MINISTRY OF EDUCATION AND TRAINING

HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY FOR HIGH QUALITY TRAINING

BH Trading PROJECT

Decentralized NFT Marketplace Website

Lecturer: Dr. Nguyễn Thiên Bảo

Members:

1. Nguyễn Hồ Quốc Bảo - 19110070
2. Nguyễn Quốc Hoàng - 19110128

HO CHI MINH CITY 12/2022

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
www.fhq.hcmute.edu.vn

SOCIALIST REPUBLIC OF VIETNAM

Independence - Freedom – Happiness

****

SOFTWARE ENGINEERING PROJECT TASK

Student name: Nguyễn Hồ Quốc Bảo Id: 19110070 Class: 19110CLA5

Student name: Nguyễn Quốc Hoàng Id: 19110128 Class: 19110CLA1

Major: Information Technology

Lecturer: Dr. Nguyễn Thiên Bảo Phone: ……..

The date of receiving the topic: 14/08/2022 Thesis submission date: 25/12/2022

1. Project name: Build a decentralized NFT marketplace website

2. Original data and documents: None

3. Content implementation of the topic:

- Learn about NextJs, Web3js, and Solidity

- Build a decentralized NFT marketplace website

LECTURER

Nguyễn Thiên Bảo

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
www.fhq.hcmute.edu.vn

SOCIALIST REPUBLIC OF VIETNAM

Independence - Freedom – Happiness

****

SUPERVISOR'S COMMENTARY

Student name: Nguyễn Hồ Quốc Bảo Id: 19110070 Class: 19110CLA5

Student name: Nguyễn Quốc Hoàng Id: 19110128 Class: 19110CLA1

Major: Information Technology

Project name: Build a decentralized NFT marketplace website.

Lecturer: Dr. Nguyễn Thiên Bảo

COMMENT

1. The topic content and workload:
   - Theory:
     - Research Solidity language, Web3Js, NextJs, Binance Smart Chain Test Net, IPFS, etherJs
     - Make a survey of popular websites about NFT Marketplace, find out the strengths and weaknesses of those systems: opensea.io, BinanceNFT.
     - Research how to build smart contracts using truffle and oppenzeppelin.
     - Research how to build user's front-end using NextJs, Vercel.
     - Research how to connect front-end with smart contract using web3js and etherJs.
   - Experiment:
     - Built Front-end using NextJs, smart contract using Solidity.
     - Connect between front-end and smart contract using web3Js.
     - Connect between front-end and database using NextJs
     - Deploy smart contracts to Binance Smart Chain Test Net.
     - Deploy user's application to vercel.
     - Successfully build an NFT marketplace website with full basic and advanced functions.
   - Link:
     - Github: https://github.com/qhoangf/My-dapp
     - Front-end: https://bh-trading.vercel.app/
     - NFT Smart contract: https://testnet.bscscan.com/address/0x72bE3b77d298c42954611D624064917e8EA96B17

- Marketplace smart contract: https://testnet.bscscan.com/address/0x61058E43a2e213B308A21F6f53cCB259ca719aab

2. Advantages:
   - Project:
     - Including basic and advanced functionalities.
     - User friendly interface and easy to use.
     - Convenient trading and minting NFT.
     - High security smart contracts
   - Group:
     - Effective communication and teamwork.
     - Dividing the work equally for each member
     - Good self-study ability
3. Disadvantages:
   - Project:
     - Responsive web is not really optimized.
     - Technology is too new to be accessible to all users.
   - Group:
     - No flexibility in meeting the instructor
     - Time for completing project of each person in group is not adequate
4. Recommend for defending or not?

…….....
.....
.....

5. Grade: .....

6. Mark: ……

Ho Chi Minh, 25/12/2022

Reviewer

(Sign, write full name)

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
www.fhq.hcmute.edu.vn

SOCIALIST REPUBLIC OF VIETNAM

Independence - Freedom – Happiness

****

REVIEWER'S COMMENTARY

Student name: Nguyễn Hồ Quốc Bảo Id: 19110070 Class: 19110CLA5

Student name: Nguyễn Quốc Hoàng Id: 19110128 Class: 19110CLA1

Major: Information Technology

Project name: Build a decentralized NFT marketplace website.

Reviewer teacher: Assoc. Prof. Dr. Hoàng Văn Dũng

COMMENT

1. Regarding the content of the topic and the volume of implementation:

.....
.....
.....
.....

2. Strength:

.....
.....

3. Drawback:

.....

4. Recommend for defending or not?

.....
.....
.....

5. Grade: .....

6. Mark: ……..

Ho Chi Minh, 02/08/2022

Reviewer

(Sign, write full name)

ACKNOWLEDGES

First of all, allow the group to express their sincere thanks to Mr. Nguyen Thien Bao, who was in charge of guiding our group in implementing and completing the report in the best and most effective way. The instructions, comments, and suggestions from the teacher have made a significant contribution to helping us complete the thesis in a complete way.

Thank you to the Faculty of Information Technology teachers for always being enthusiastic, dedicated, and dedicated to answering our questions. Besides, we would like to thank our classmates and seniors who did not hesitate to share useful information and experiences to help us improve our thesis.

The report was made in a limited time, along with the limitations of knowledge, so errors are inevitable in the implementation process. We look forward to receiving your comments and suggestions from the teachers so that we can gain more experience and complete the topics better in the future.

We sincerely thank you!

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
www.fhq.hcmute.edu.vn

SOCIALIST REPUBLIC OF VIETNAM

Independence - Freedom – Happiness

****

GRADUATE THESE SUBJECTS

Student name: Nguyễn Hồ Quốc Bảo Id: 19110070 Class: 19110CLA5

Student name: Nguyễn Quốc Hoàng Id: 19110128 Class: 18110CLA1

Major: Information Technology

Project time from 01/08/2022 đến: 25/12/2022

Project name: Build a decentralized NFT marketplace website.

Lecturer: Dr. Nguyễn Thiên Bảo

Mission of Project:

1. Learn about NextJs, blockchain, web3js
2. Research on how to write smart contracts using Solidity and Truffle.
3. Research on common smart contract attacks
4. Use the MySQL library to create a Table to save data
5. Use Web3Js to interact with smart contracts
6. Deploy and verify smart contracts to Binance Smart Chain
7. Using NextJs for front-end.
8. Testing

Table Contents

IMPLEMENTATION PROGRESS

Week	Work
1	Discuss the topic and refer to related popular applications (Bảo, Hoàng)
2	Write NFT and marketplace smart contract (Bảo) Build source FrontEnd(Bảo, Hoàng)
3	Functional design: Connect Metamask (Bảo) Building button and popup to connect wallet (Hoàng) Design Homepage (Hoàng) Deploy smart contracts to Binance Smart Chain (Bảo)
4	Design Exchange page (Hoàng) Functional desgin: get all NFTs, Transfer NFT (Bảo)
5	Design function:Mint NFT (Bảo) Design Mint Page (Bảo, Hoàng) Call API from blockchain to get status of minting (Bảo) Fix function Mint NFT from smart contract (Bảo)
6	Design function: Get all NFT (Bảo) Design Your NFT page (Hoàng)
7	Design function: get NFT information (Bảo) Design NFT detail page (Hoàng)
8	Design function: List, Buy, Change price, Cancel listing, choose type of token (Bảo) Design pupop to cancel listing, change price (Hoàng)
9	Fix function list anh buy NFT on smart contract (Bảo) Design function check if NFT is listed or not (Hoàng)
10	Design function Get all listed NFTs (Bảo) Design marketplace page (Hoàng)
11	Design Search bar (Hoàng) Design function search NFT by tokenId
12	Design function: Get address balance Design popup show balance (Hoàng, Bảo)
13	Design function: Withdraw blanace (Bảo) Design popup to withdraw balance (Hoàng)
14	Design function: Change price to Mint on smart contract (Bảo) Design 404 page (Hoàng)
15	Check and fix basic functions (Bảo, Hoàng)

16	Write test case to test the website (Bảo, Hoàng)
17	Complete and supplement reports (Bảo, Hoàng)

CHAPTER 1: OVERVIEW

1.1. Introduction

NFT is a very hot phrase for the last 2 years, it is extremely rich in all countries, and people can make money by using their own music, and photos to become NFTs and sell them to everyone in the world.

After learning and experiencing, we want to use the new knowledge after self-study to build a marketplace where people can create their own NFTs and sell them in a decentralized environment.

Besides, the smart contract used for the marketplace will make the transactions transparent and extremely safe.

1.2. Purpose of the project

Creates a decentralized NFT marketplace website where all users can make their own NFT and sell them to other users.

1.3. Object and scope of the project

The research was conducted around a group of people with knowledge of blockchain technology and NFT.

the group of technology knowledge subjects includes 4 compulsory subjects to study: NextJS, Web3JS Library, MySQL database, and Smart contract.

The scope of the research is set out at a general level, the researcher understands the general knowledge of the research content and can apply each knowledge content to the actual product, not placing heavy emphasis on theories. non-applicable.

1.4. Expected results.

Creates a decentralized NFT marketplace website that can prevent attacks. All users can make their own NFTs and sell them to other users, users also can buy other NFTs.

CHAPTER 2. STATE OF THE ART AND SYTEM REQUIREMENTS

2.1. State of the art

Currently, with the strong development of technology and NFT, there are many large blockchain technology companies in the world that have created large-scale NFT exchanges where all users can Search and buy and sell the famous NFT collections. Not only that, but users can also create their own NFTs and sell them for profit.

- **Opensea website: https://opensea.io/fr**

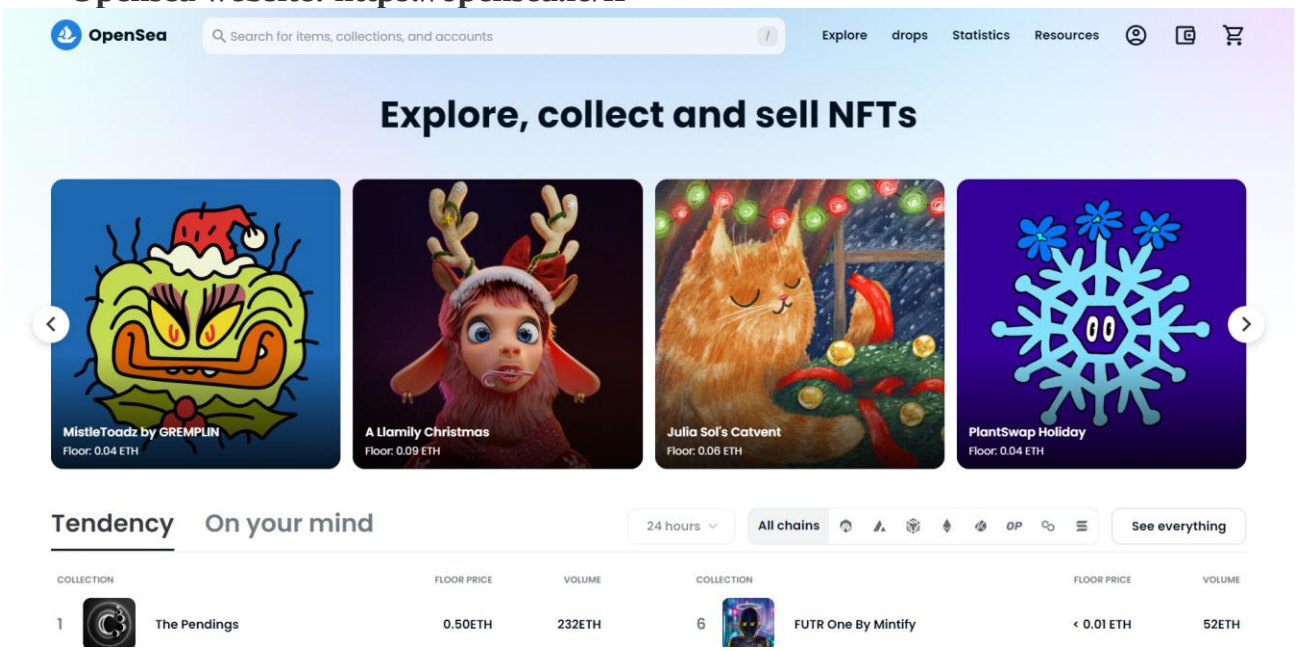

Figure 1: Opensea website

Advantages	• Very simple and clear interface to display categories and products. • There are many collections. • Faster access and easier purchases. • Easier to incentivize buyers. • Good navigation during click to view product details. • The website introduces the full information of the website (referral, policy, documents).
Disadvantages	• Still a centralized exchange • Too much information displayed • Does not support multi-language.

- **Binance NFT website: https://www.binance.com/en/nft/home**

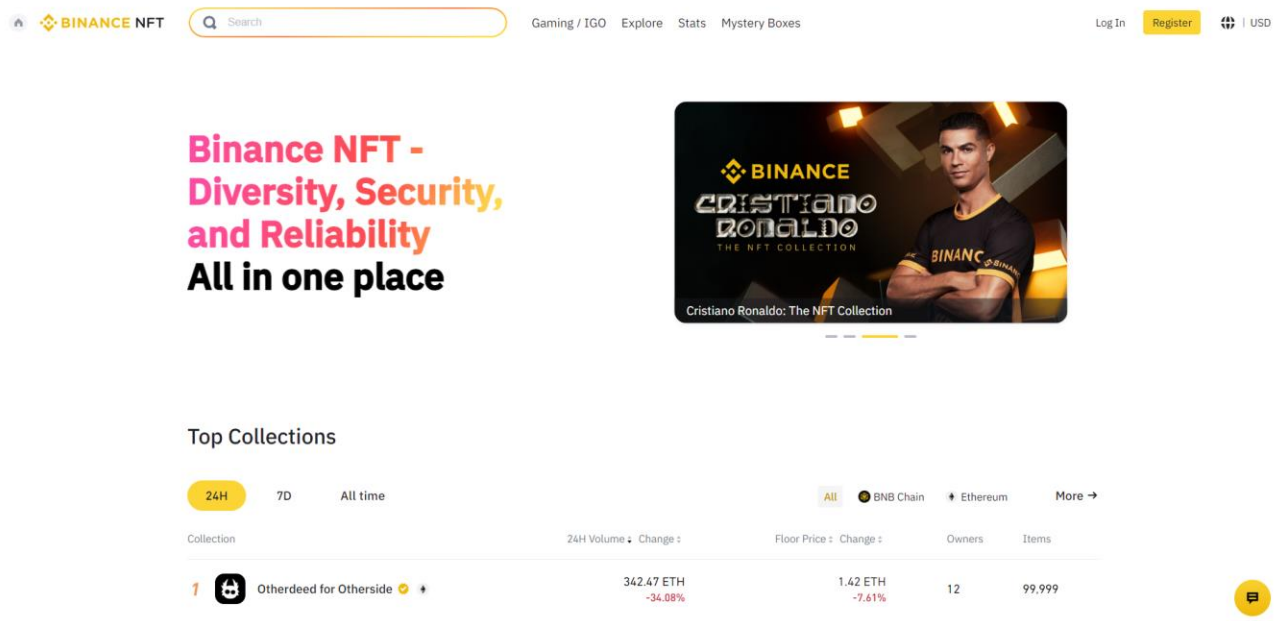

Figure 2: Binance NFT website

Advantages	• Very simple and clear interface to display categories and products. • Huge time savings for consumers. • Faster access and easier purchases. • Easier to incentivize buyers. • Good navigation during click to view product details. • The website introduces the full information of the website (referral, policy, documents). • Online chat boxes are supported.
Disadvantages	• Still a centralized exchange •The arrangement of the layout is not beautiful

2.2. Functional requirements

- Connecting Wallet to website
- System management function for admin.
- Profit management.
- Mint, buying and selling NFT.
- Change price, cancel selling NFT.

2.3. Non-functional requirements

Ordinal Number	Content	Purpose	Detail descriptions
1	User can list All NFTs from others ERC-721 smart contract	Scalability	Users don't have to list NFTs minted from our web
2	Our website is built based on web3 technology	Reliability	Because of web3 technology, all information of all transaction can be public on blockchain and everyone can view them
3	Well-built smart contracts	Security	We consulted a lot of documents and knowledge to build a smart contract that is extremely tight and resistant to attacks.

4	Everyone can use our website	Availability	Anyone can use our product if that user has knowledge about using Metamask

CHAPTER 3. SYSTEM DESIGN

3.1. Use case design

3.1.1. Main features

Actors	Use cases / Responsibility
Marketplace's admin	- Connect admin wallet - View name of the NFT's smart contract - View price to mint NFT - Change the price to mint NFT - View the profit - Withdraw the profit - View statistic
Customer	- Connect wallet - Mint NFT - View owned NFTs - Transfer owned NFT to others wallet - List NFTs to marketplace - Change price to sell NFT - Cancel listing form marketplace - Choose payment method to sell NFTs - Buy NFTs from marketplace - View NFTs from others user - View history of transactions - View balance - Withdraw balance

Table 1: Main feature

3.1.2. Use-case diagram

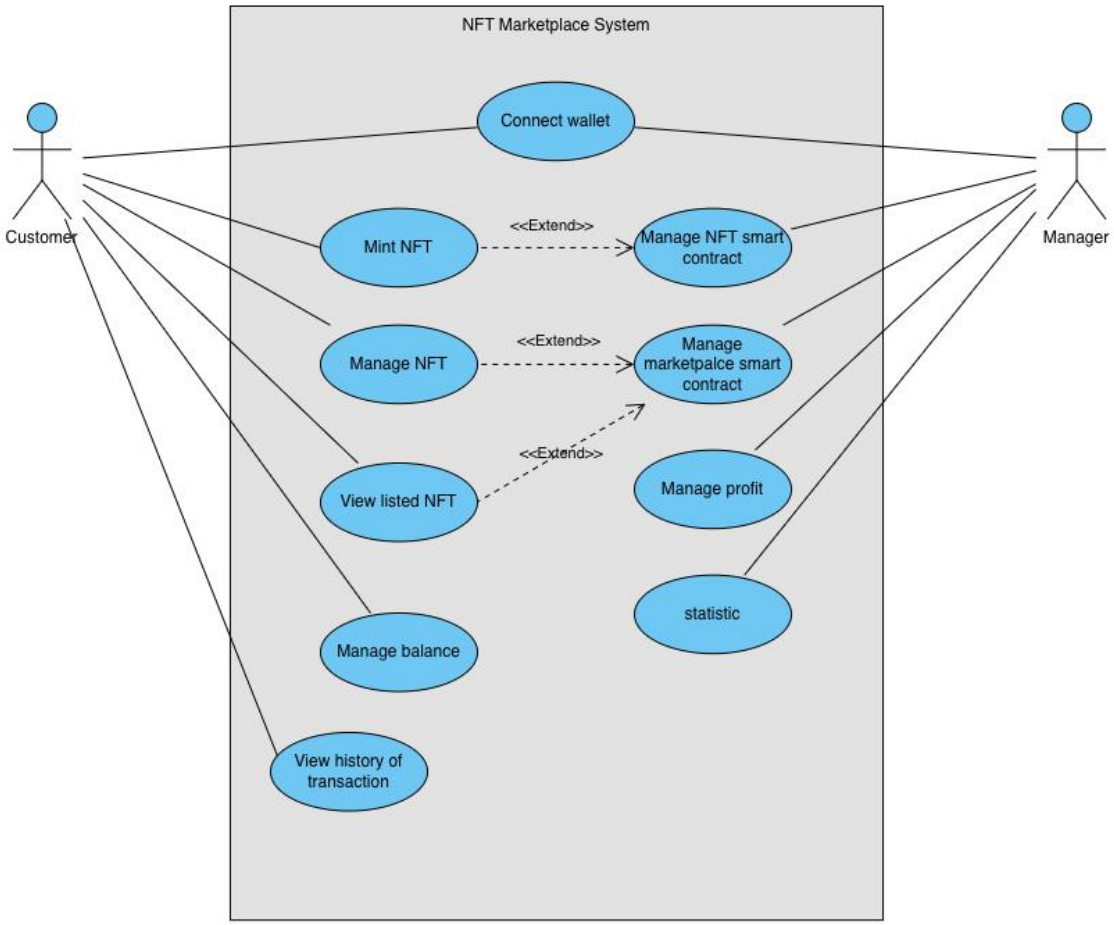

Figure 3: Use-case diagram

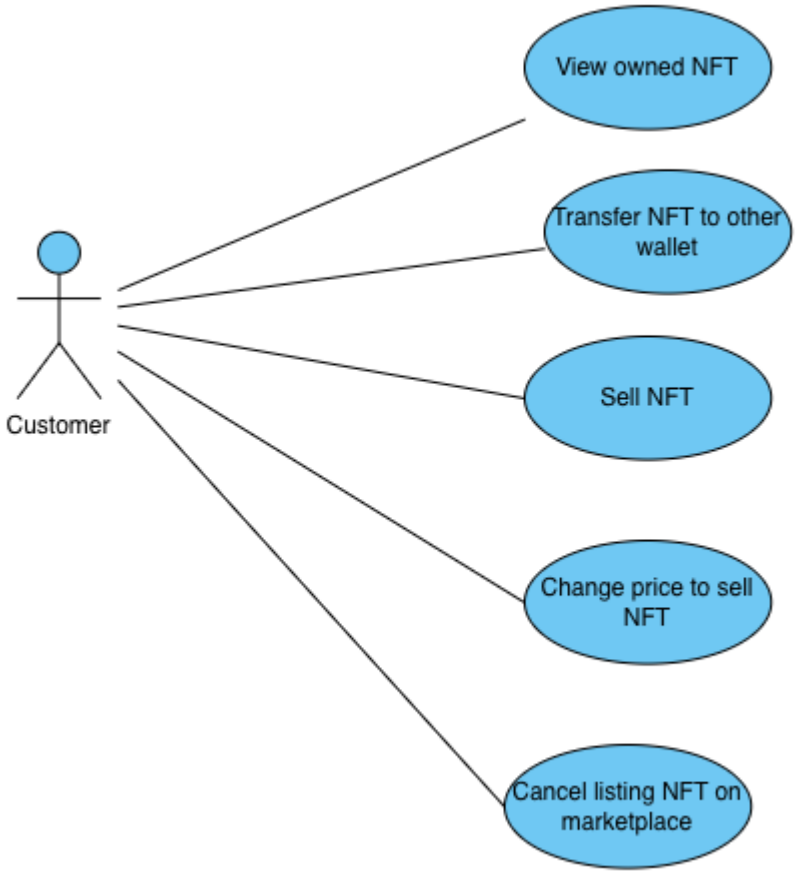

Figure 4: Manage NFTs use-case

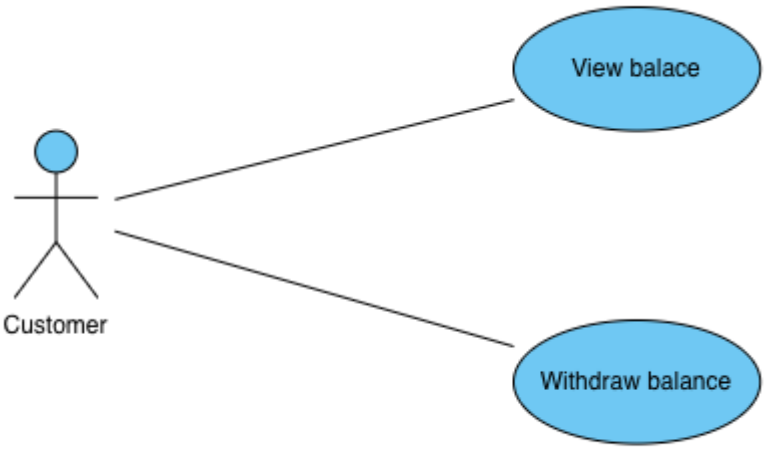

Figure 5: Manage balance use-case

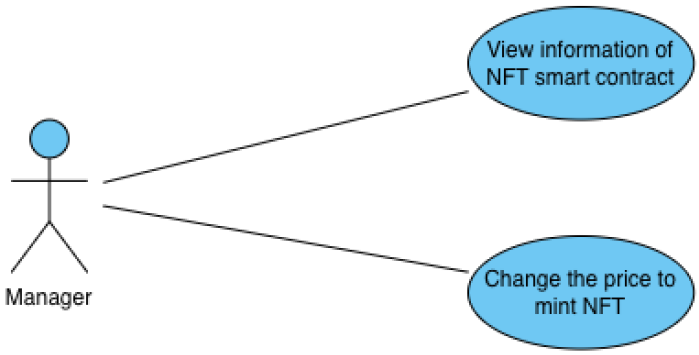

Figure 6: Manage NFT smart contract use-case

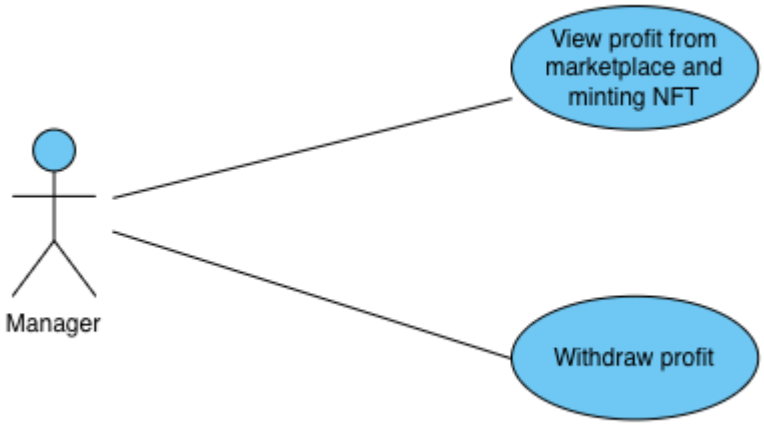

Figure 7: Manage profit use-case

3.1.3. Function Description

3.1.3.1. Connect wallet

Connect Wallet	
Description	Allow manager and user login to the system
Activation agent	Admin, Customer
Pre-conditions	(1) Admin: Has the admin wallet to connect (2) User: allow any wallet
Step to be taken	*Admin page:* (1) Open admin page (2) Press connect wallet button

	(3) Choose admin wallet and connect (4) If fail, the message "You are not the admin, please choose other wallet" will appear and request to connect other wallet (5) If successful, Admin page will appear *Customer page:* (1) Open customer page (2) Press connect button on navigation bar (3) If login fail, message "User denied to connect" will appear (4) If successful, wallet address will appear on navigation bar

Table 4: Connect wallet use-case

3.1.3.2. Mint NFT

Mint NFT	
Description	Allow user to mint NFT
Activation agent	Customer
Pre-conditions	Each wallet has only 10 times to mint NFT
Step to be taken	(1) User open system (2) Connect wallet (3) Select "Mint NFT" button on navigation bar (4) Input information of NFT (5) Click "Submit" button (6) Pay the fee to mint NFT (7) If failed, message of error will appear

	(8) If successful, message of information of transaction will appear

Table 5: Mint NFT use-case

3.1.3.3. List NFTs

List NFTs	
Description	Allow user to list their NFTs to marketplace
Activation agent	Customer
Pre-conditions	Allow list any NFT from users
Step to be taken	(1) Open system (2) Connect Wallet (3) Click "Your NFT" button on navigation bar (4) Choose NFT want to list to marketplace (5) Click "Sell" button (6) Choose coin want to trade (7) Enter the price (8) Click "Submit" button (9) A message will appear for use to sign for listing NFT (10) If successful, a message "Listing NFT successfully" will appear (11) If failed, a message of error will appear and request to try again

Table 6: List NFT use-case

3.1.3.4. Change Price to Sell

Change Price to Sell	
Description	Allow user to change the price of which NFT is listed
Activation agent	Customer
Pre-conditions	Only change the price of NFT which is owned and listed
Step to be taken	(1) Open system (2) Connect Wallet (3) Click "Your NFT" button on navigation bar (4) Choose NFT want to change the price (5) Click "Change Price" button (6) Choose coin want to trade (7) Enter the new price (8) Click "Submit" button (9) A message will appear for use to sign for change price to sell NFT (10) If successful, a message "Change Price for selling NFT successfully" will appear (11) If failed, a message of error will appear and request to try again

Table 7: Change price use-case

3.1.3.5. Listing NFT

Cancel Listing NFT	
Description	Allow user to cancel listing NFT from marketplace

Activation agent	Customer
Pre-conditions	Only cancel NFT which is owned and listed
Step to be taken	(1) Open system (2) Connect Wallet (3) Click "Your NFT" button on navigation bar (4) Choose NFT want to cancel listing (5) Click "Confirm" button (6) A request will appear to cancel listing (7) Pay the gas fee to cancel listing (8) If successful, a message "Cancel listing successfully" will appear (9) If failed, a message of error will appear

Table 8: Cancel listing use-case

3.1.3.6. Withdraw Balance

Withdraw Balance	
Description	Allow user to withdraw their balance
Activation agent	Customer
Pre-conditions	Only withdraw balance owned by their wallet

Step to be taken	(1) Open system (2) Connect Wallet (3) Click “Wallet” button (4) Click “Withdraw” button (5) Choose coin want to withdraw (6) Click “Submit” button (7) A request will appear (8) User pay gas fee to withdraw balance (9) If successful, a message “Withdraw successfully” will appear (10) If failed, a message of error will appear

Table 9: Withdraw balance use-case

3.1.3.7. Buy NFT

Buy NFT	
Description	Allow user to buy NFT from marketplace
Activation agent	Customer
Pre-conditions	Only buy NFTs which listed on marketplace
Step to be taken	(1) Open system (2) Connect Wallet (3) Click “Marketplace” button on navigation bar (4) Choose NFT want to buy (5) Click “Buy” button” (6) Click “Submit” (7) A request will appear

	(8) User pay a gas fee any price to Buy (9) If successful, a message "Buy NFT successfully" will appear (10) If failed, a message of error will appear

Table 10: Buy NFT use-case

3.1.3.8. Transfer NFT

Transfer NFT	
Description	Allow user to transfer their NFT to other wallet
Activation agent	Customer
Pre-conditions	Only transfer NFTs which owned
Step to be taken	(1) Open system (2) Connect Wallet (3) Click "Exchange" button on navigation bar (4) Enter the wallet which will receive the NFT (5) Choose which NFT want to transfer (6) A request will appear (7) User pay a gas fee to transfer NFT (8) If successful, a message "Transfer NFT successfully" will appear (9) If failed, a message of error will appear

Table 11: Buy NFT use-case

3.1.3.9. Withdraw Profit

Withdraw Profit	
Description	Allow admin to withdraw all profit
Activation agent	Admin
Pre-conditions	Only admin can withdraw profit

Step to be taken	(1) Open admin system (2) Connect Wallet (3) Click “Balance” button (4) Click “Withdraw” button (5) A request will appear (6) Admin pay a gas fee to withdraw (7) If successful, a message “Withdraw successfully” will appear (8) If failed, a message of error will appear

Table 12: Withdraw profit use-case

3.1.4. Sequence diagram

3.1.4.1. Sequence Transfer NFT

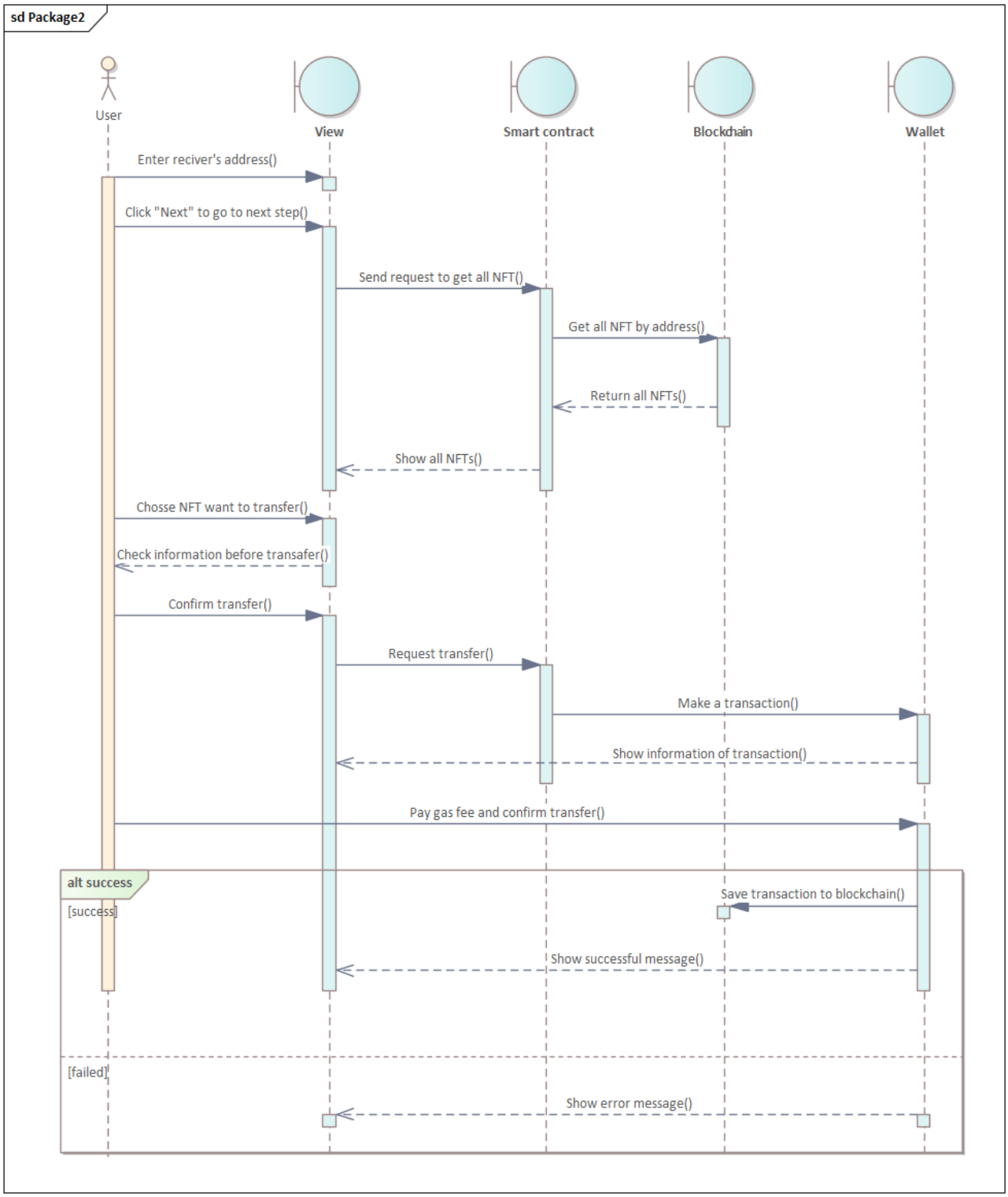

Figure 8: Sequence transfer NFT

3.1.4.2. Sequence Mint NFT

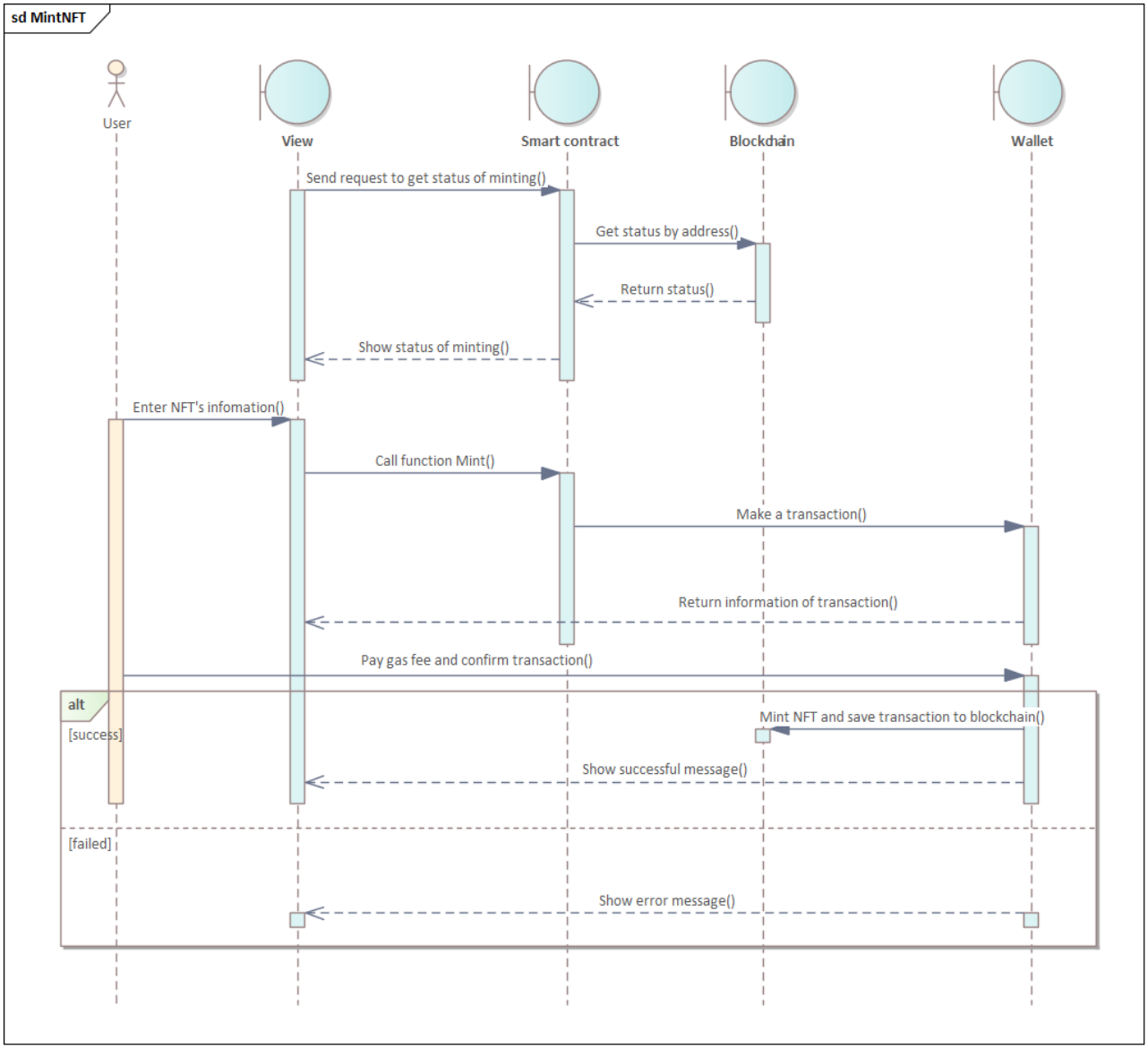

Figure 9: Sequence mint NFT

3.1.4.3. Sequence View all NFT

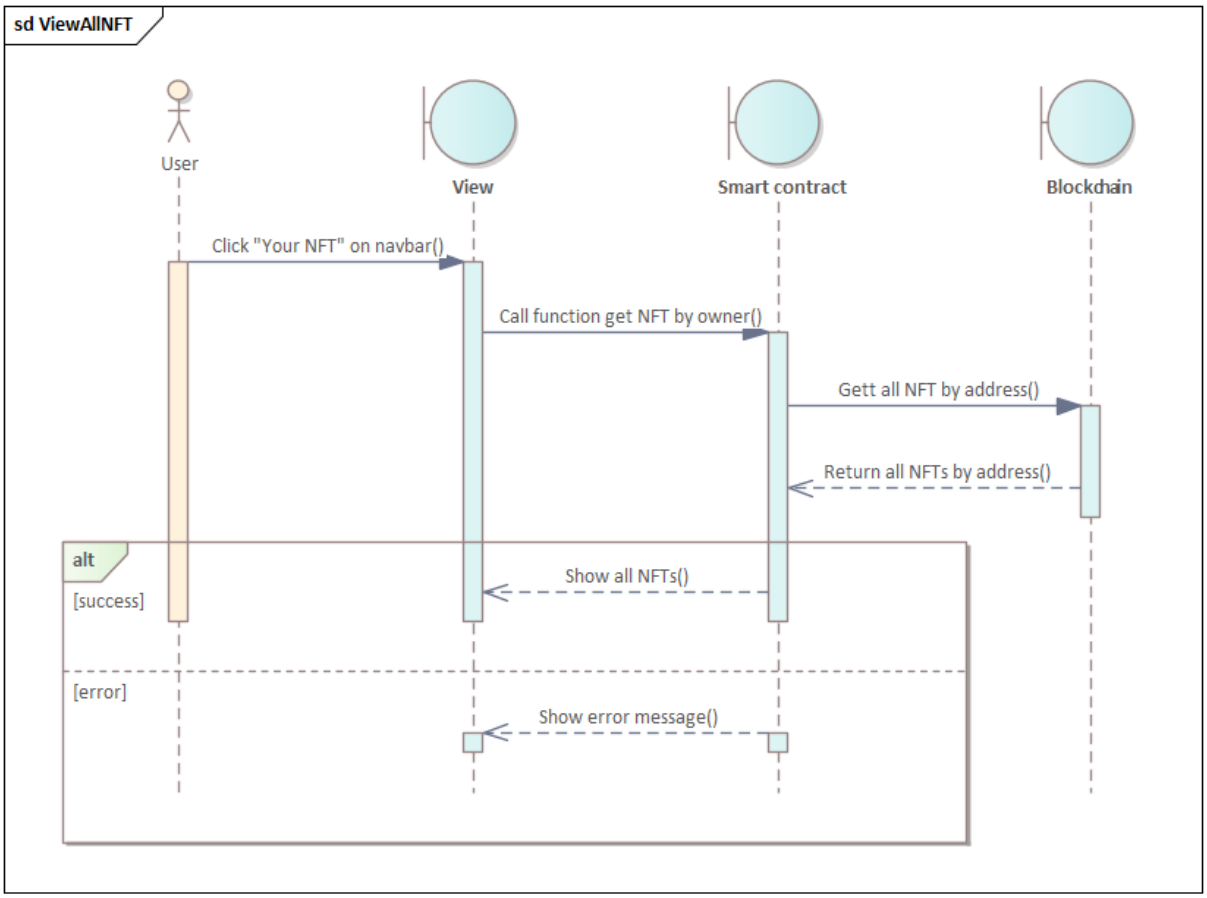

Figure 10: Sequence View all NFTs

3.1.4.4. Sequence Search NFT

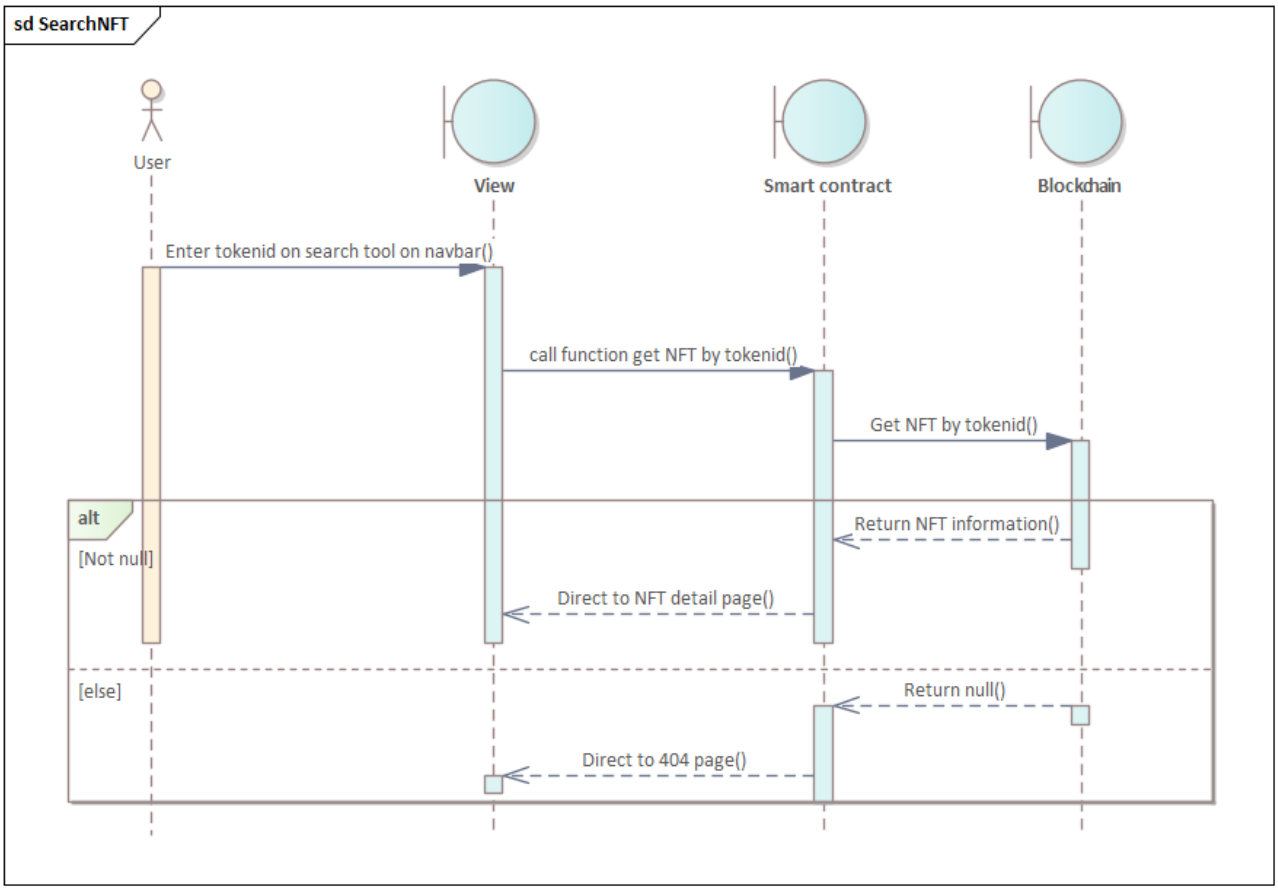

Figure 10: Sequence search NFT

3.1.4.5. Sequence List NFT

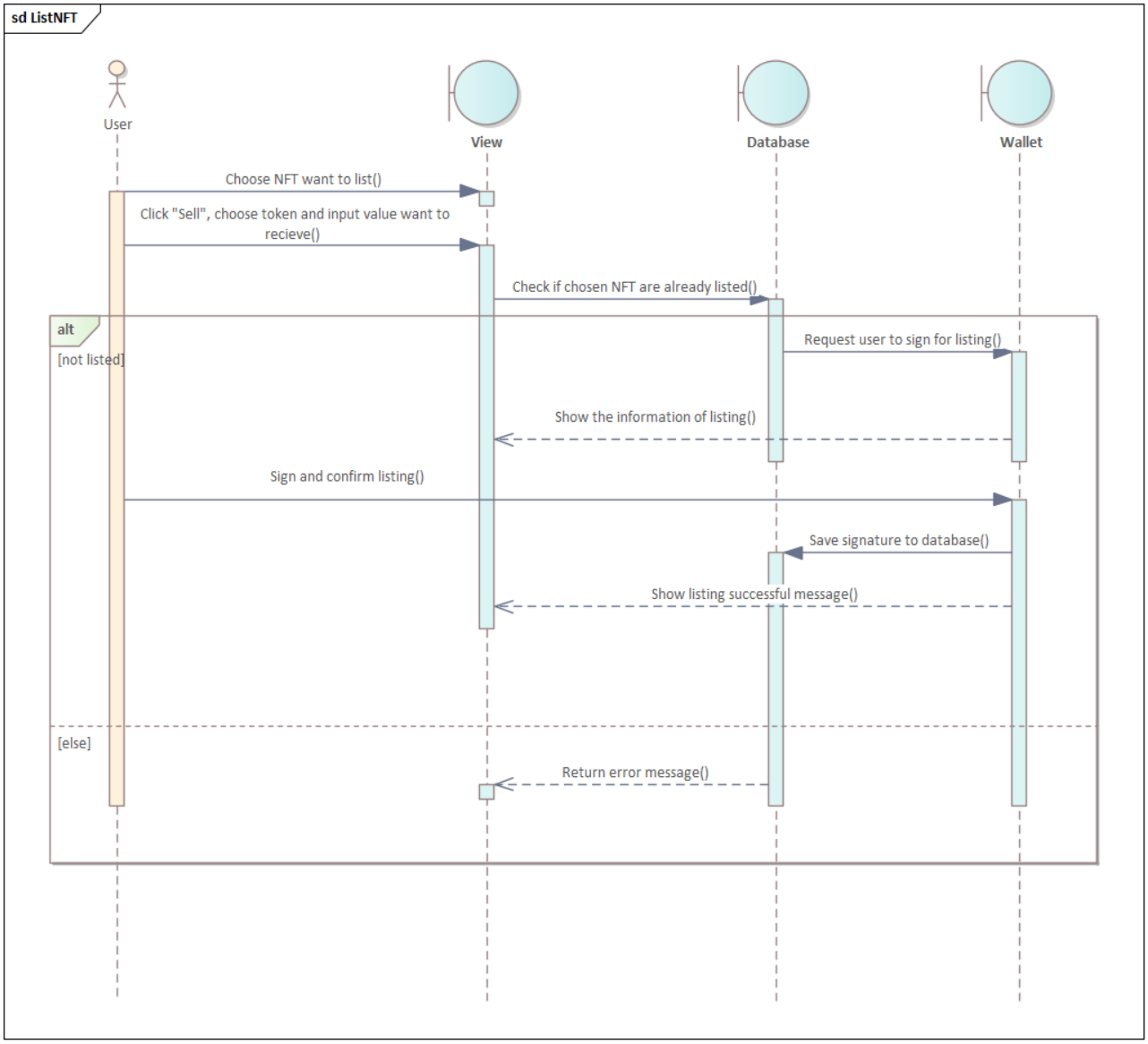

Figure 11: Sequence list NFT

3.1.4.6. Sequence Buy NFT

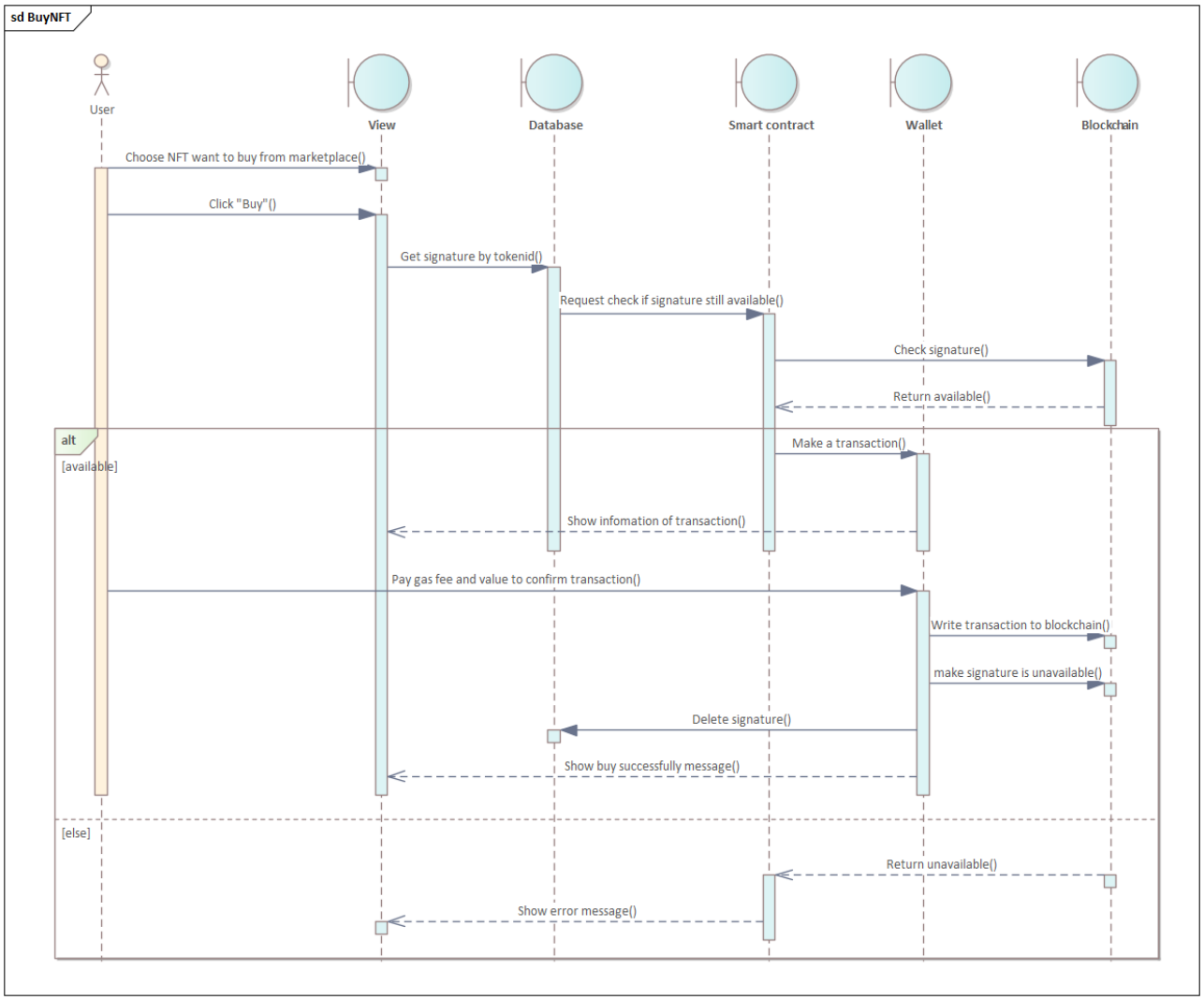

Figure 11: Sequence buy NFT

3.1.4.7. Sequence Change price

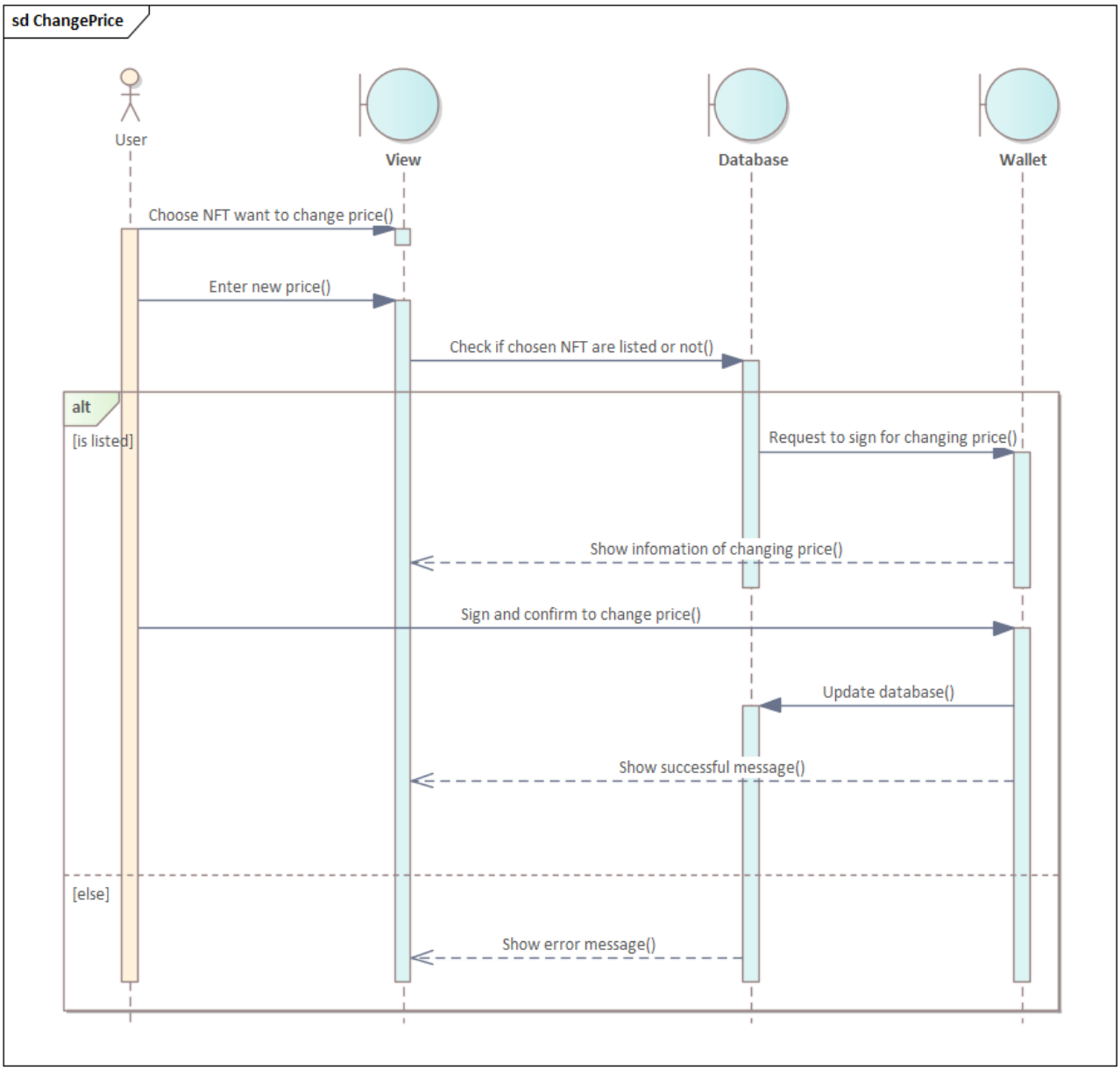

Figure 12: Sequence change price

3.1.4.8. Sequence Cancel listing

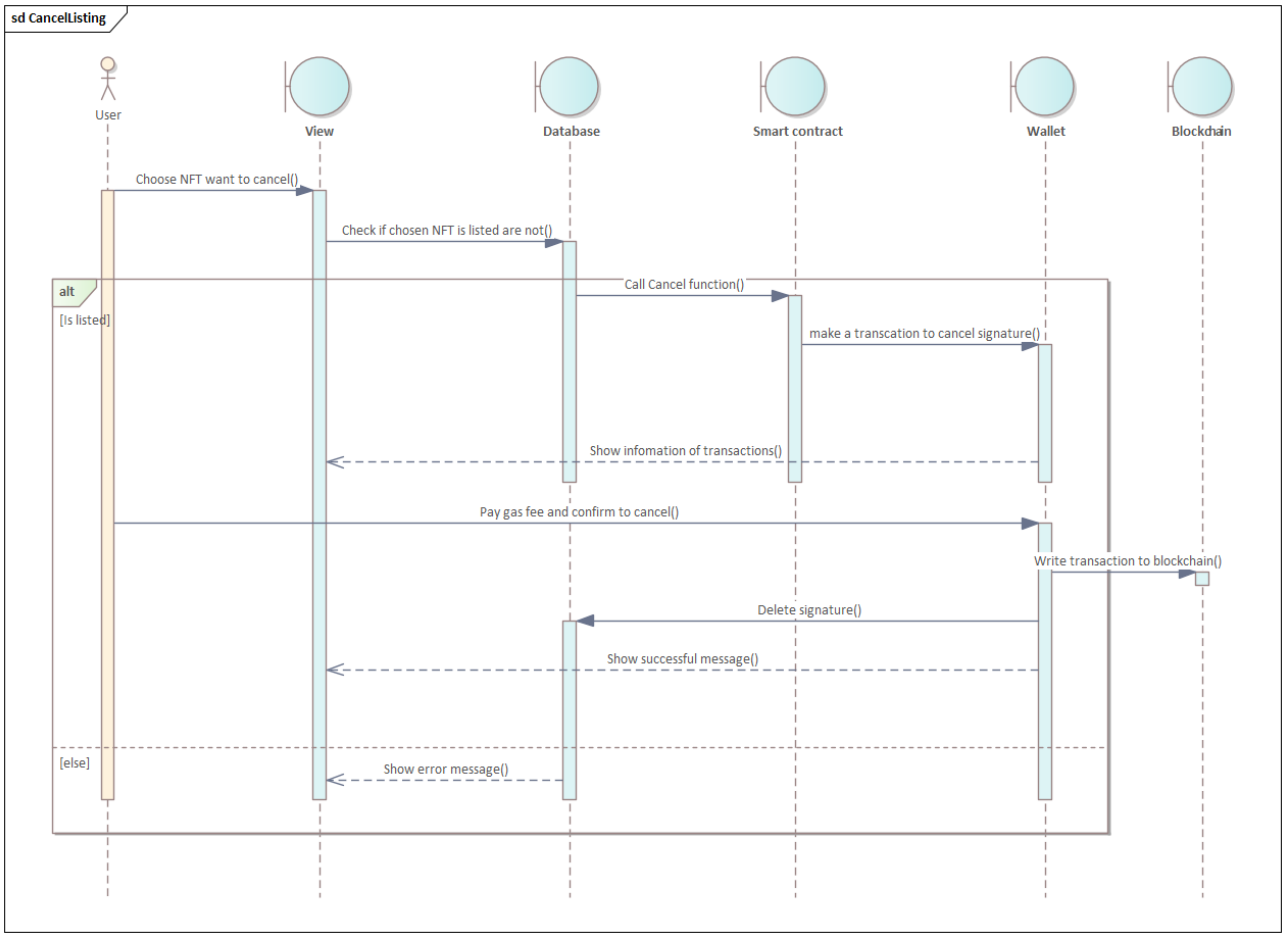

Figure 13: Sequence cancel listing

3.1.4.9. Sequence Withdraw balance

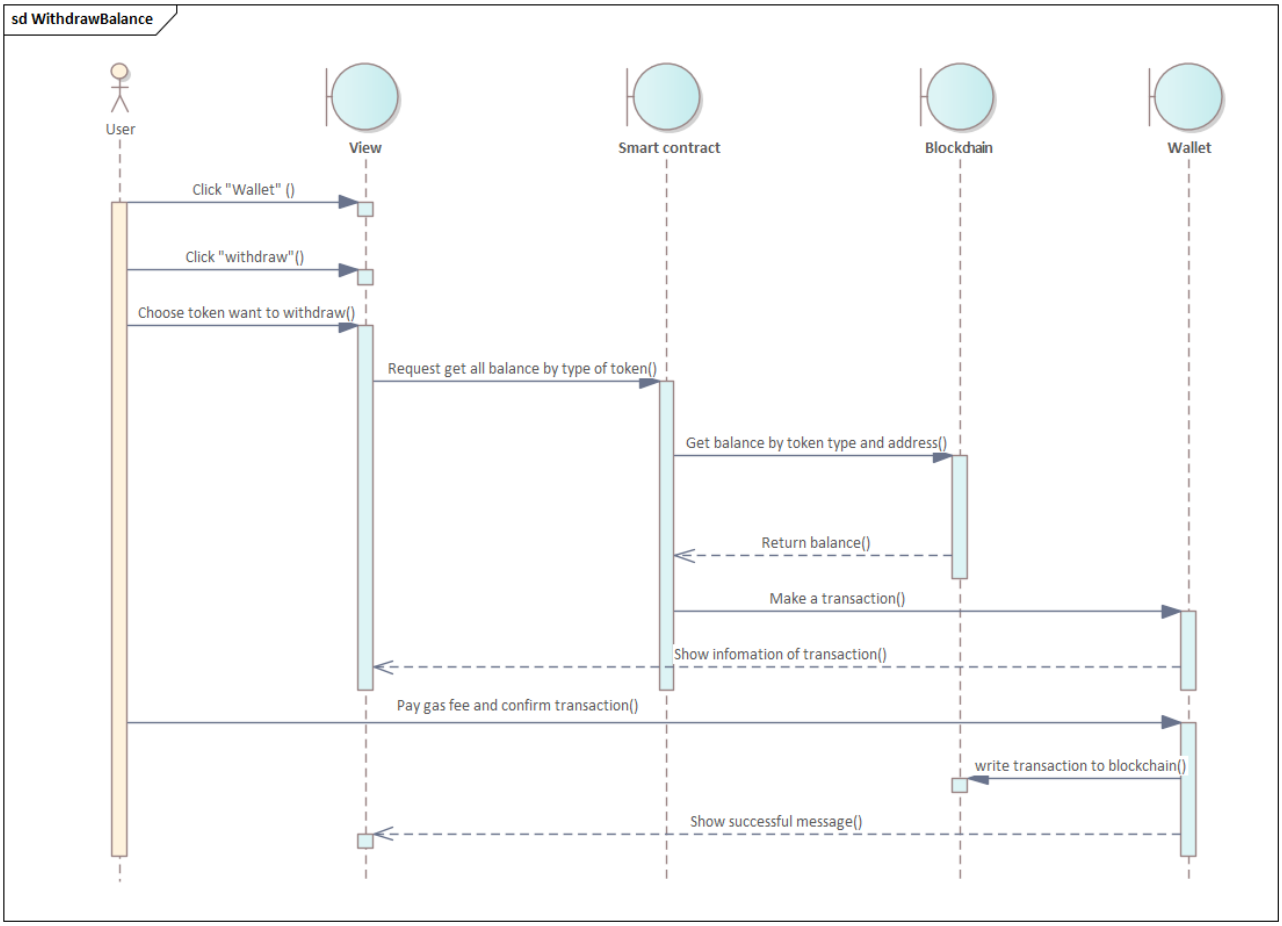

Figure 14: Sequence withdraw balance

3.2. Database design

Marketplace:

ID	Attribute	Type	Range	Meaning
1	tokenId	INT		Token Id
2	owner	VARCHAR	50	Owner of NFT

3	name	VARCHAR	255	Name of NFT
4	image	VARCHAR	255	Link to NFT's image
5	price	VARCHAR	255	Price for selling NFT
6	Contract_address	VARCHAR	255	NFT contract address
7	coupon	VARCHAR	255	Signature of seller
8	typecoin	VARCHAR	50	Type of coin for selling

Table 13: Database marketplace

3.3. Smart contract design

3.3.1. ERC-20 smart contract

ID	Function	Type	Input	Meaning
1	admin	Read		Return admin of smart contract
2	allowance	Read	owner, spender	Returns the result of allowance of the owner's token to the spender
3	balanceOf	Read	account	Return balance of given account

4	decimals	Read		Return the decimal of the token's unit
5	name	Read		Return the name of the token
6	symbol	Read		Return the symbol of the token
7	totalSupply	Read		Return total supply of the token
8	approve	Write	spender, amount	Authorize spender the given number of tokens
9	burn	Write	amount	Burn the given number of tokens
10	decreaseAllowance	Write	spender, subtractedValue	Decrease the Authorization to spender with the given number of tokens
11	increaseAllowance	Write	spender, addedValue	Increase the Authorization to spender with the given number of tokens
12	mint	Write	to, amount	Mint the given number of tokens to given address (only owner can mint)
13	transfer	Write	to, amount	Transfer given number of tokens to given address
14	transferFrom	Write	from, to, amount	Transfer given number of tokens from given address to other address

Table 14: ERC-20 smart contract

3.3.2. NFT smart contract

ID	Function	Type	Input	Meaning
1	balanceOf	Read	account	Return balance of given account
2	getApproved	Read	tokenID	Return address which approved by owner
3	getBalanceContract	Read		Return balance on smart contract
4	getMaxMint	Read	owner	Returns the number of minted turns
5	isApprovedForAll	Read	owner, operator	Return if operator is approved by owner
6	mintPrice	Read		Return price for minting NFT
7	name	Read		Return the name of the NFT
8	owner	Read		Return owner of smart contract
9	ownerOf	Read	tokenId	Return owner of the given token id
10	supportInterface	Read	interfaceId	Returns true if this contract implements the interface defined by interfaceId
11	symbol	Read		Return the symbol of the NFT

12	tokenByIndex	Read	index	Return tokenId by given index
13	tokenOwnerOfByIndex	Read	owner, index	Return token id by given index and account
14	tokenURI	Read	tokenId	Return metadata of given token id
15	totalSupply	Read		Return total supply of NFT
16	approve	Write	to, tokenId	Approve the given token Id to the given address
17	mintNft	Write	uri	Mint NFT with the given metadata
18	renounceOwnership	Write		Leaves the contract without owner (only owner)
19	safeMintNft	Write	uri	Mint NFT with the given metadata
20	safeTransferFrom	Write	from, to, tokenId	Transfer the given token id from the given address to other address
21	setApprovalForAll	Write	operator, approved	Approve or remove operator as an operator for the caller. Operators can call transferFrom or safeTransferFrom for any token owned by the caller
22	setMintPrice	Write	mintPrice	Update price to mint NFT (only owner)
23	transferFrom	Write	from, to, tokenId	Transfer the given token id from the given address to other address

24	transferOwnership	Write	newOwner	Transfer smart contract to new owner (only owner)
25	withdraw	Write	amount	Withdraw the given amount of balance from contract (only owner)

Table 15: NFT smart contract

3.3.3. Marketplace smart contract

ID	Function	Type	Input	Meaning
1	getProceeds	Read	seller	Return balance of BNB by the given address
2	getProceedsErc20	Read	seller	Return balance of ERC-20 tokens by the given address
3	buyItem	Write	nftAddress, tokenId, hashedMessage, _v, _r, _s, admin, price	Buy NFT using BNB payment method
4	buyItemWithErc20	Write	nftAddress, tokenId, hashedMessage, _v, _r, _s, admin, price	Buy NFT using ERC-20 token payment method
5	cancelListing	Write	nftAddress, tokenId, hashedMessage, _v, _r, _s	Cancel listing NFT
6	withdrawProceeds	Write		Withdraw all BNB balance of caller

7	withdrawProceedsErc20	Write		Withdraw all ERC-20 tokens balance of caller

Table 16: marketplace smart contract

3.4. User Interface

3.4.1. Home page (Index page)

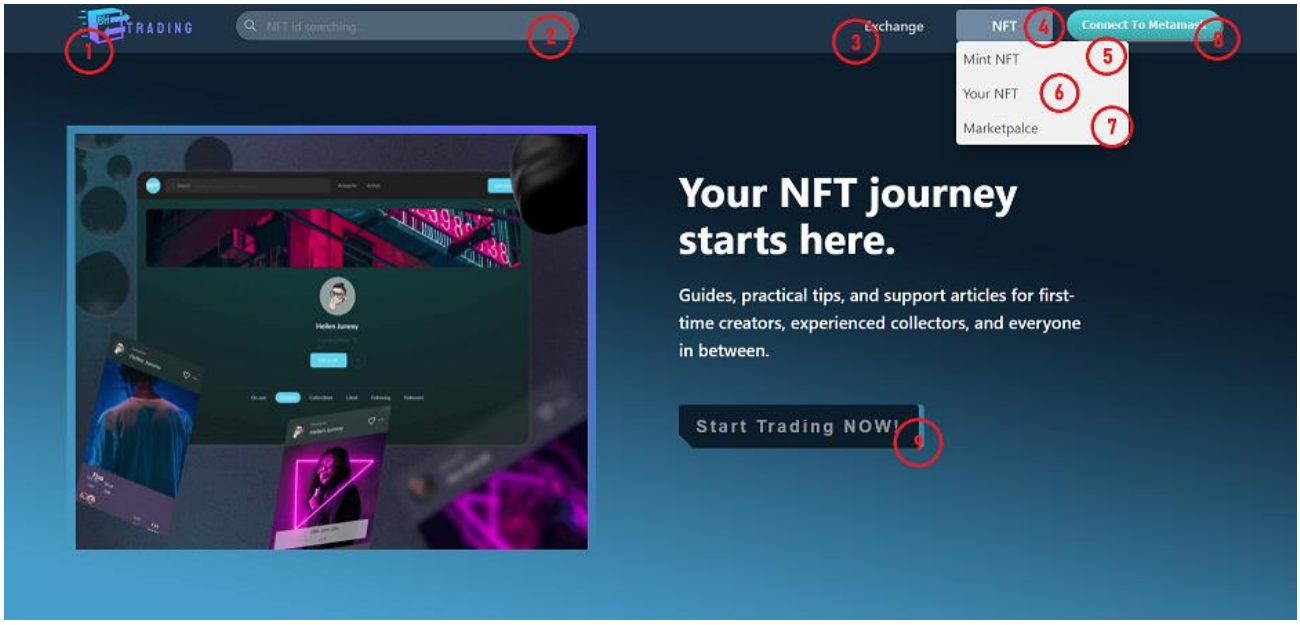

Figure 15: Homepage Section 1

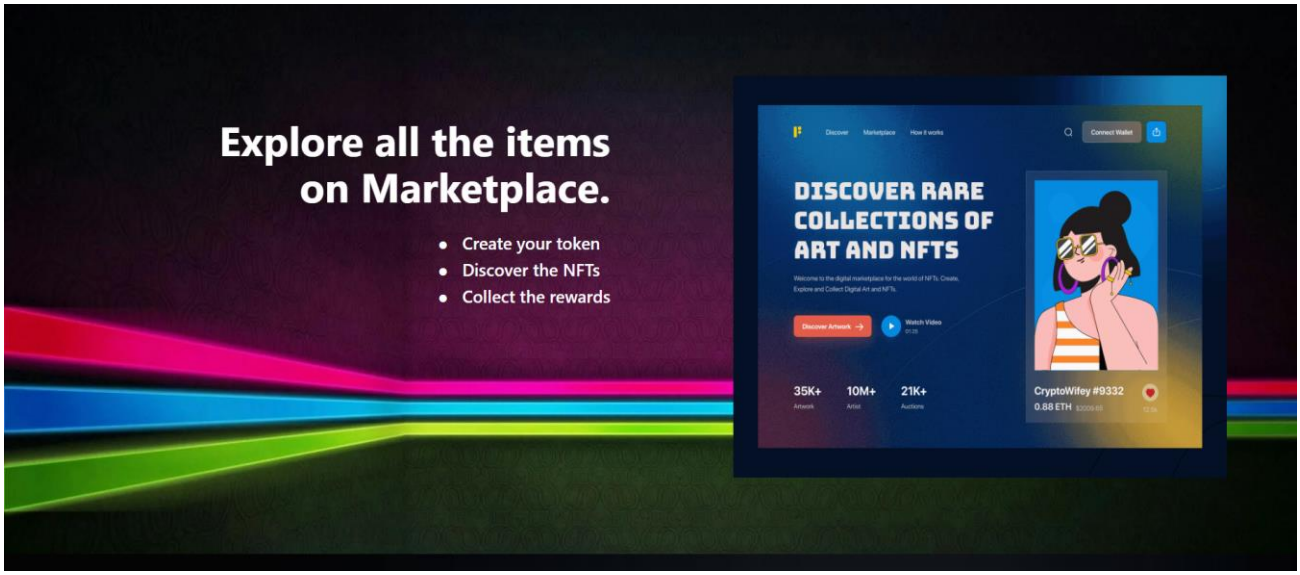

Figure 16: Homepage Section 2

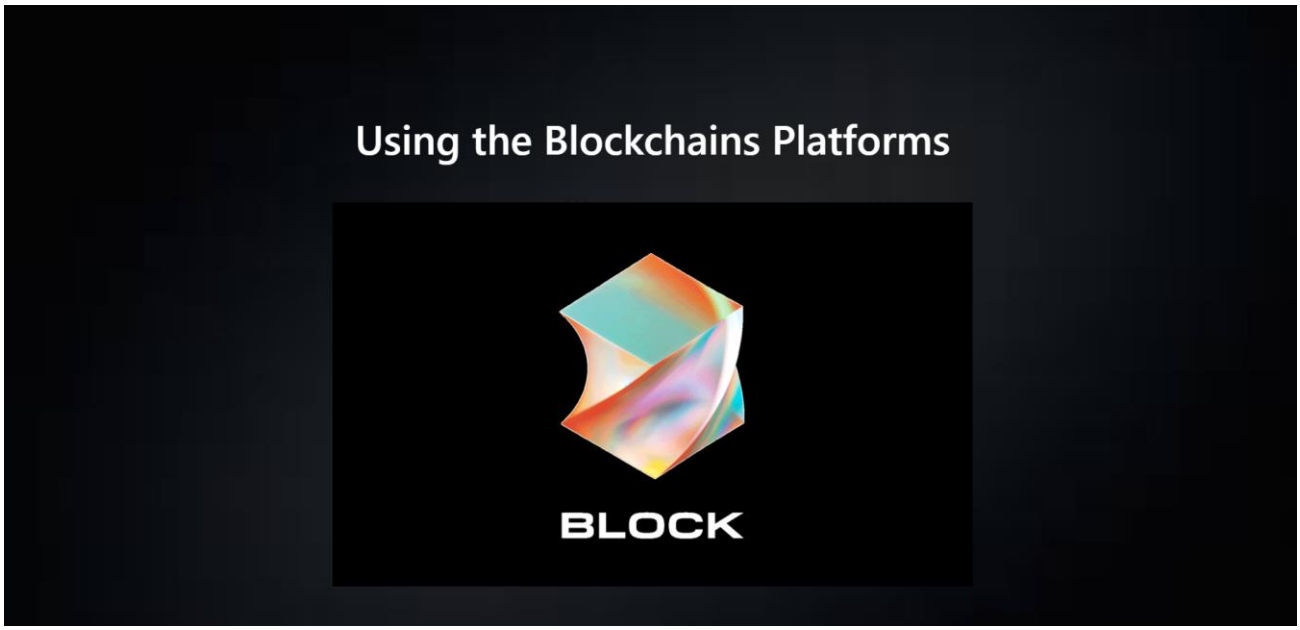

Figure 17: Homepage Section 3

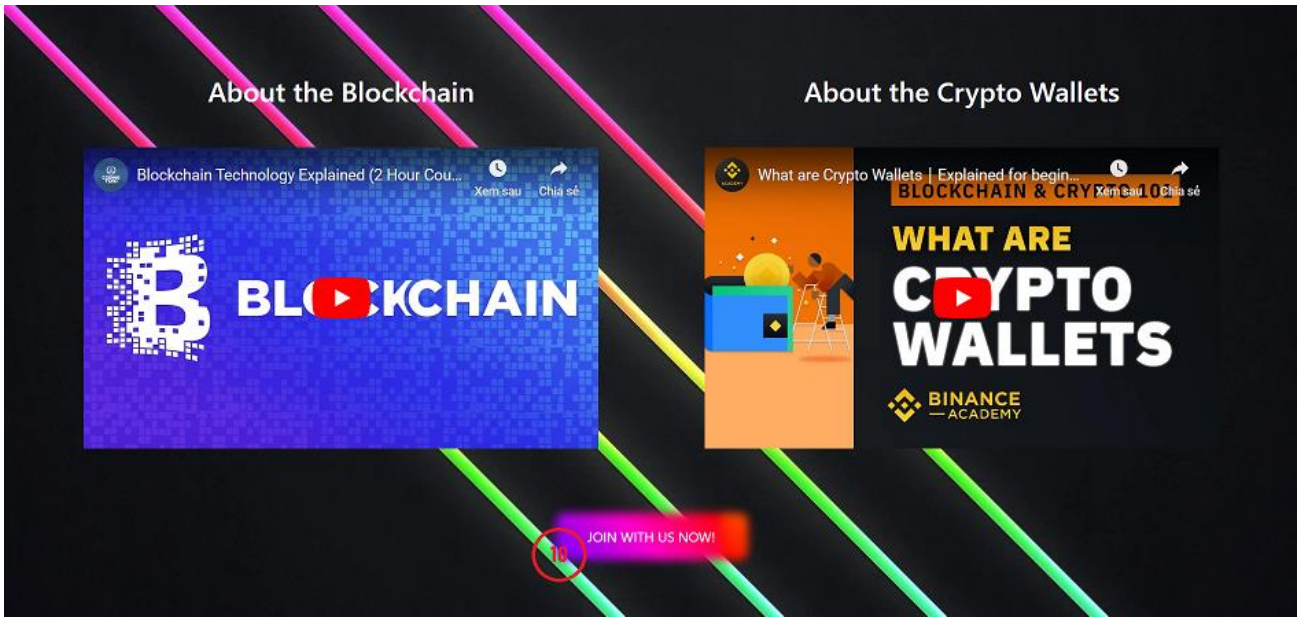

Figure 18: Homepage Section 4

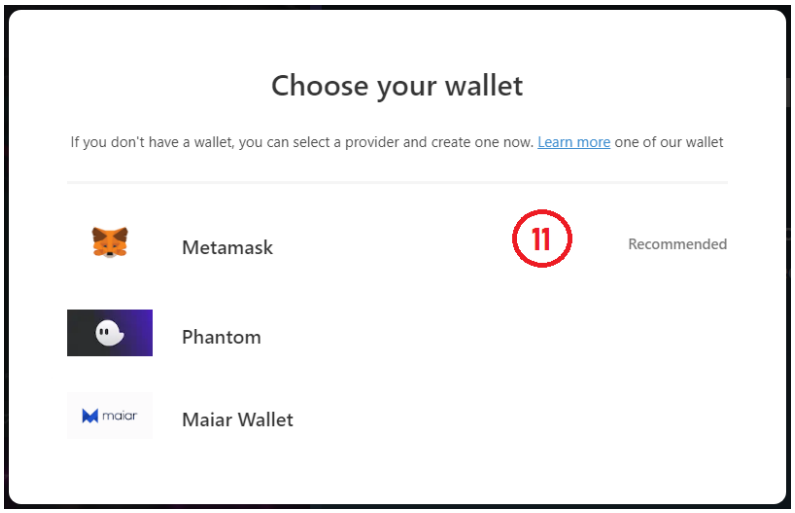

Figure 19: Homepage Wallet List

Order	Type	Meaning
1	Button (Home)	Move to home page
2	Input (Search)	Searching NFT by using Tokenid
3	Button (Exchange)	Move to exchange page
4	Button (NFT)	Show list as dropdown of 3 pages (Mint NFT, Your NFT, Marketplace)
5	Button (Mint NFT)	Move to Mint NFT page
6	Button (Your NFT)	Move to Your NFT page
7	Button (Marketplace)	Move to Marketplace page
8	Button (Connect Wallet Metamask)	Click to appear Metamask Popup to connect Metamask Wallet
9	Button (Showing Wallet List using Popup)	Click to show Wallet List Popup
10	Button (Showing Wallet List using Popup)	Click to show Wallet List Popup
11	Button on Wallet List (Connect Wallet Metamask)	Click to appear Metamask Popup to connect Metamask Wallet

Table 17. Homepage Entities

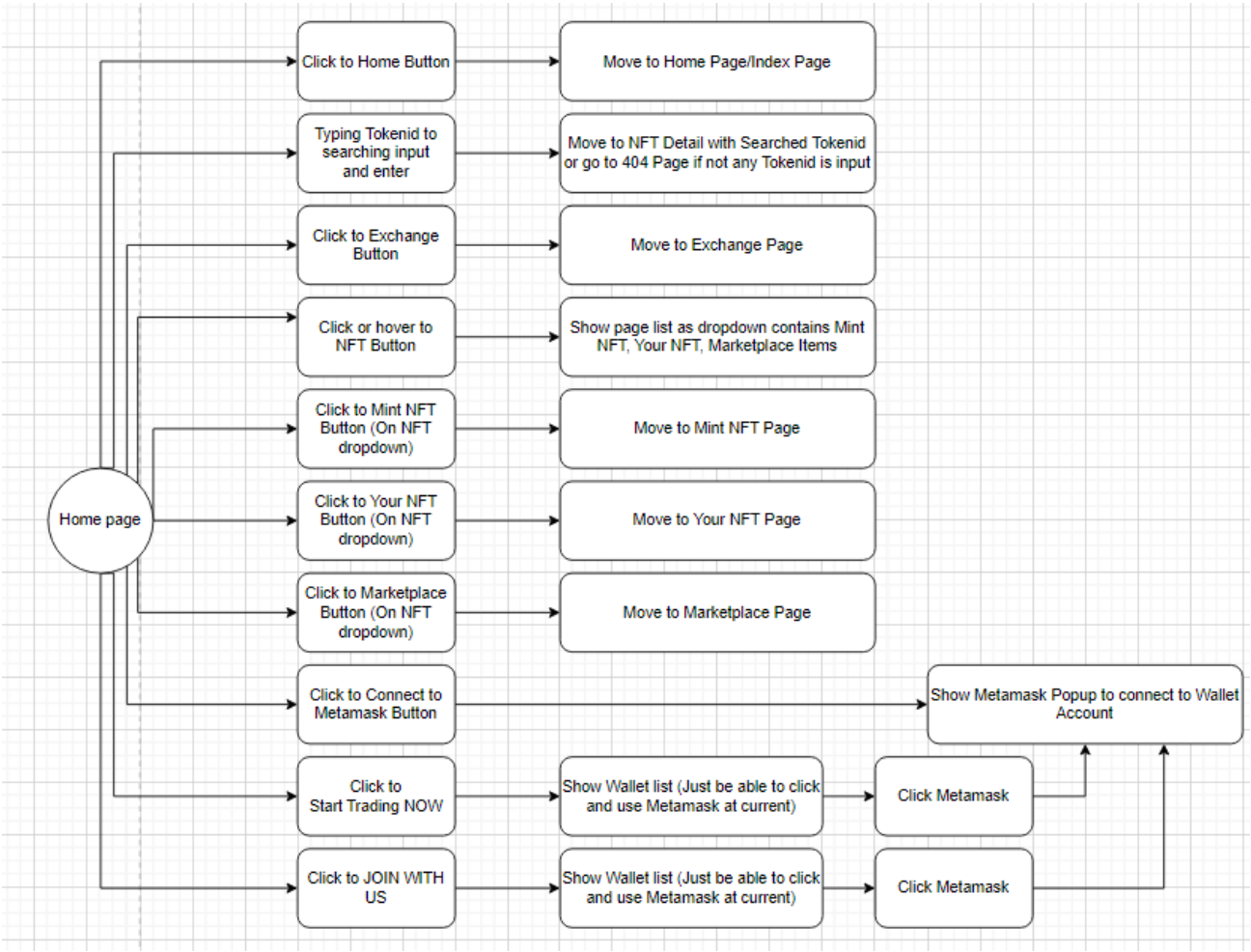

Figure 20: Event handling homepage diagram

3.4.2. Mint NFT Page

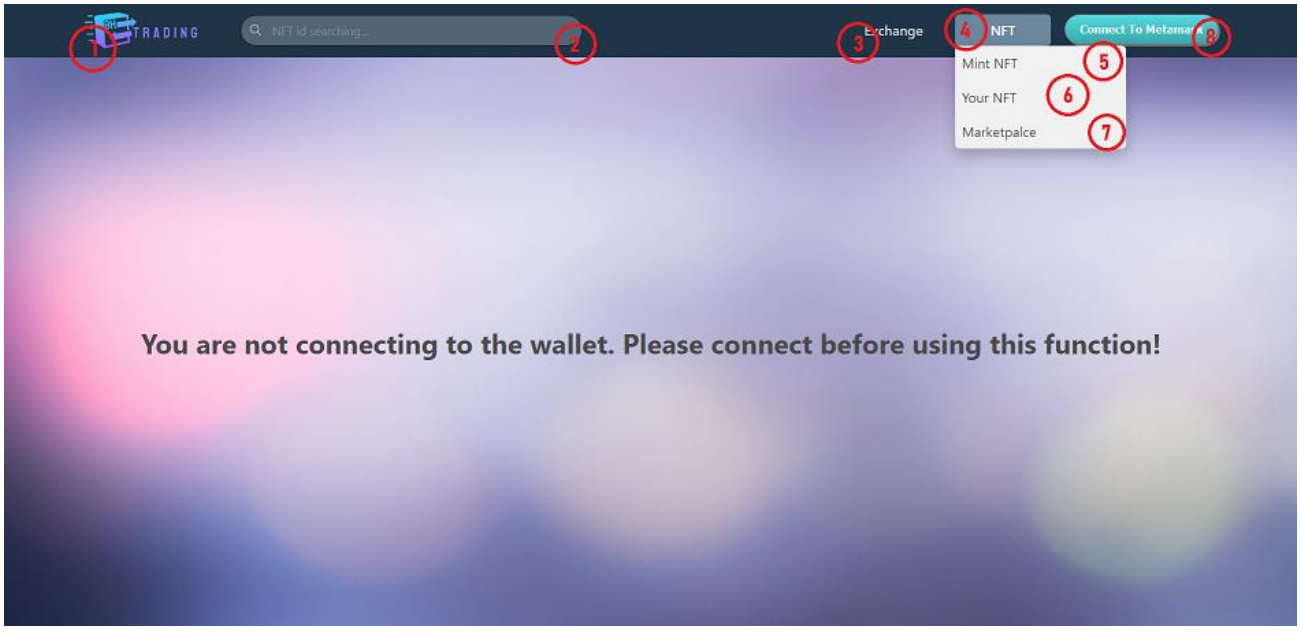

Figure 21: Mint NFT Page (Not login)

Order	Type	Meaning
1	Button (Home)	Move to home page
2	Input (Search)	Searching NFT by using Tokenid
3	Button (Exchange)	Move to exchange page
4	Button (NFT)	Show list as dropdown of 3 pages (Mint NFT, Your NFT, Marketplace)
5	Button (Mint NFT)	Move to Mint NFT page
6	Button (Your NFT)	Move to Your NFT page
7	Button (Marketplace)	Move to Marketplace page
8	Button (Connect Wallet Metamask)	Click to appear Metamask Popup to connect Metamask Wallet

Table 18. Entities in Mint NFT (Not login) page

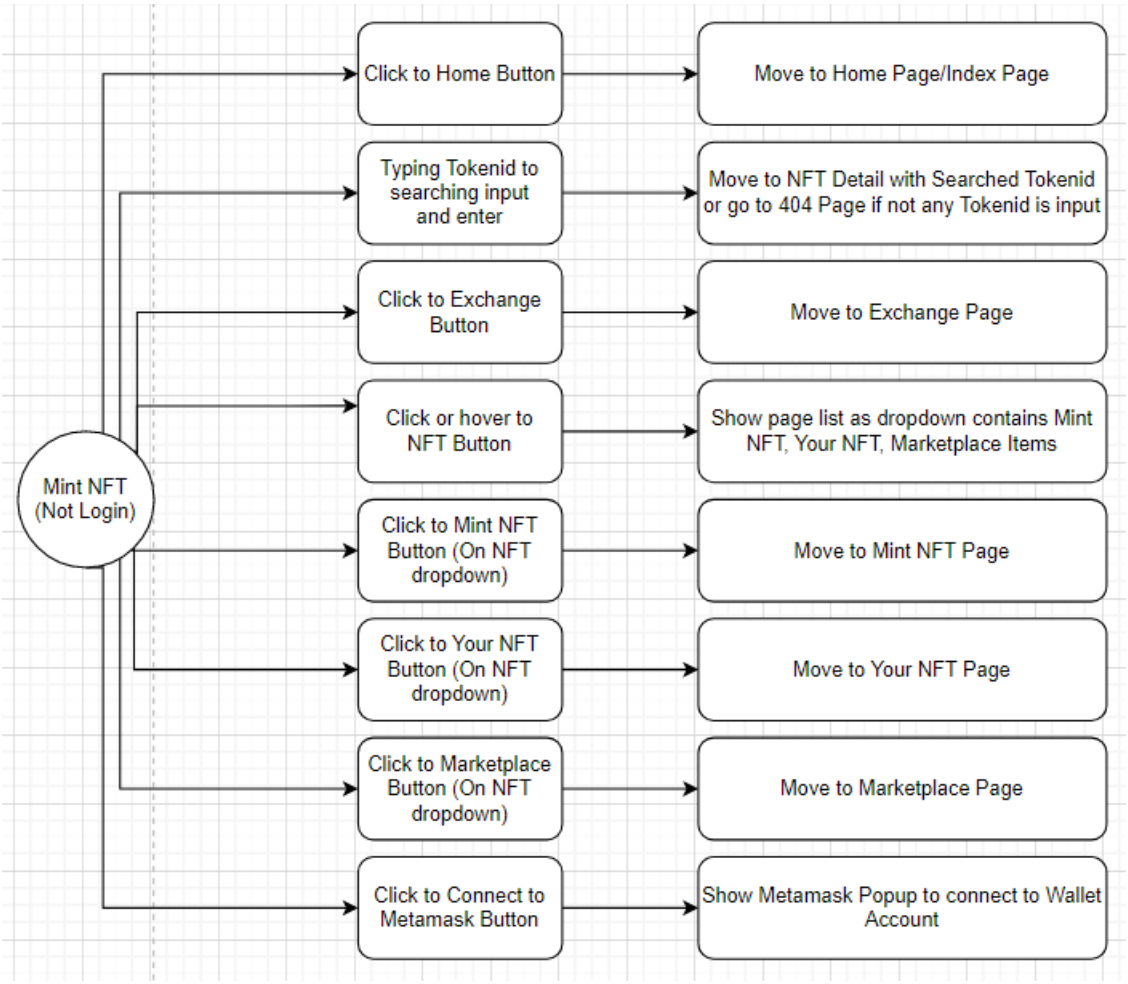

Figure 22: Event handling Mint NFT (Not login) diagram

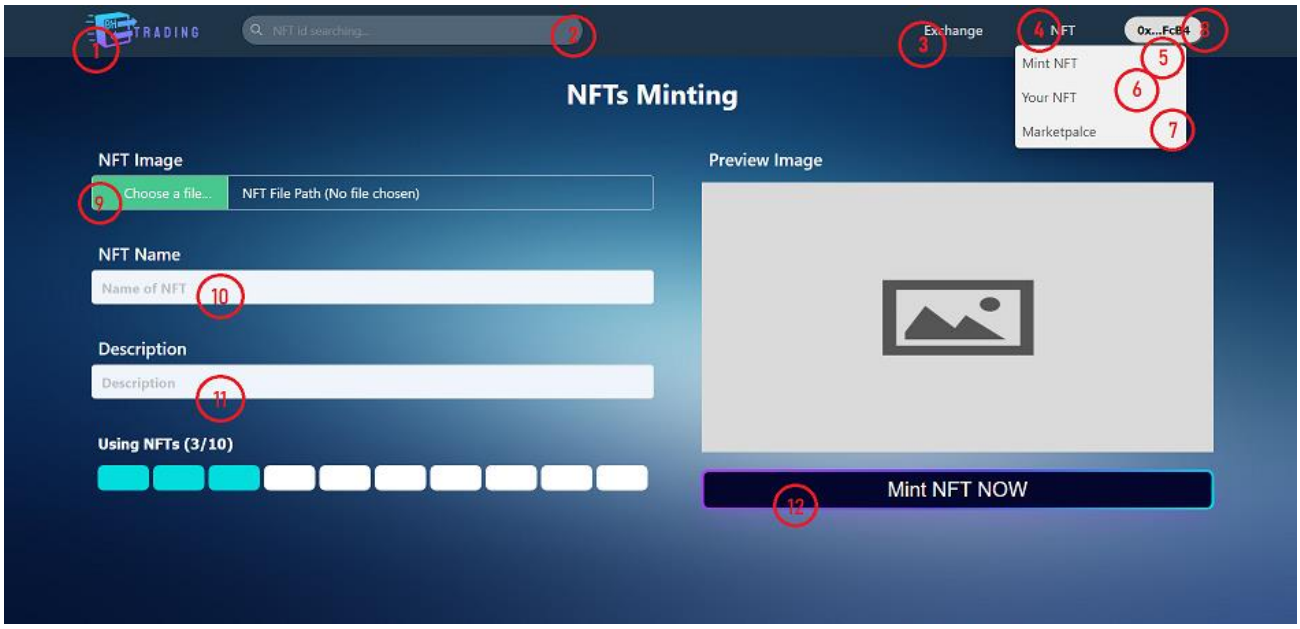

Figure 23: Mint NFT Page (Logged In)

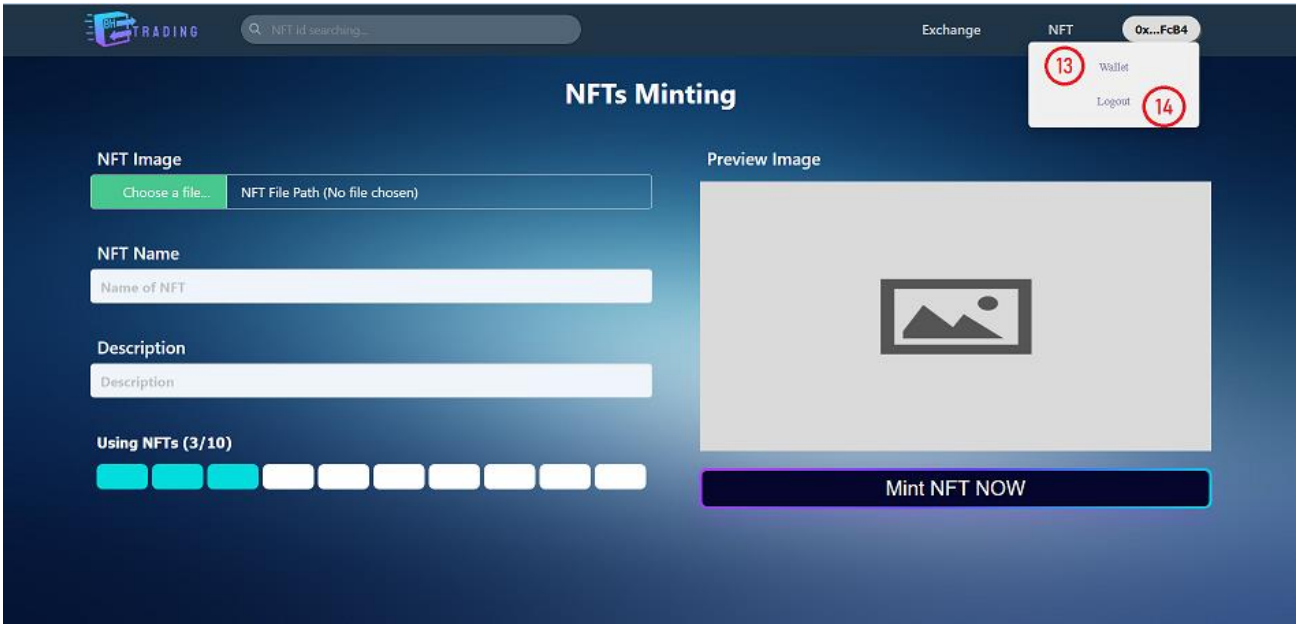

Figure 24: Mint NFT Page (Logged In) – Account Dropdown

Order	Type	Meaning
1	Button (Home)	Move to home page
2	Input (Search)	Searching NFT by using Tokenid
3	Button (Exchange)	Move to exchange page
4	Button (NFT)	Show list as dropdown of 3 pages (Mint NFT, Your NFT, Marketplace)
5	Button (Mint NFT)	Move to Mint NFT page

6	Button (Your NFT)	Move to Your NFT page
7	Button (Marketplace)	Move to Marketplace page
8	Button (Metamask Account)	Click to interact to current account
9	Button (Choose Image)	Click to choosing image on your desktop
10	Input (NFT Name)	Input NFT name
11	Input (NFT Description)	Input NFT description
12	Button (Mint NFT NOW)	Click to Mint current NFT
13	Button (Wallet)	Click to see Token Wallet
14	Button (Log out)	Click to log out current Account

Table 19. Entities in Mint NFT (Logged in) page

3.4.3. Your NFT page

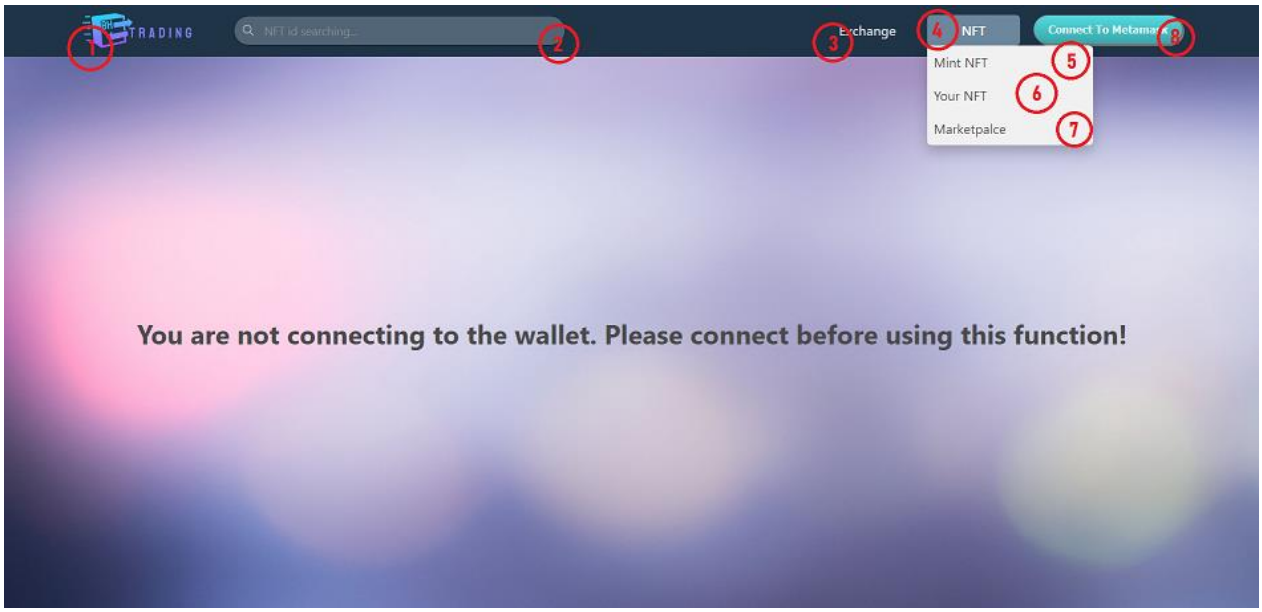

Figure 25: Your NFT Page (Not login)

Order	Type	Meaning
1	Button (Home)	Move to home page
2	Input (Search)	Searching NFT by using Tokenid
3	Button (Exchange)	Move to exchange page

4	Button (NFT)	Show list as dropdown of 3 pages (Mint NFT, Your NFT, Marketplace)
5	Button (Mint NFT)	Move to Mint NFT page
6	Button (Your NFT)	Move to Your NFT page
7	Button (Marketplace)	Move to Marketplace page
8	Button (Connect Wallet Metamask)	Click to appear Metamask Popup to connect Metamask Wallet

Table 20. Entities in Your NFT (Not login) page

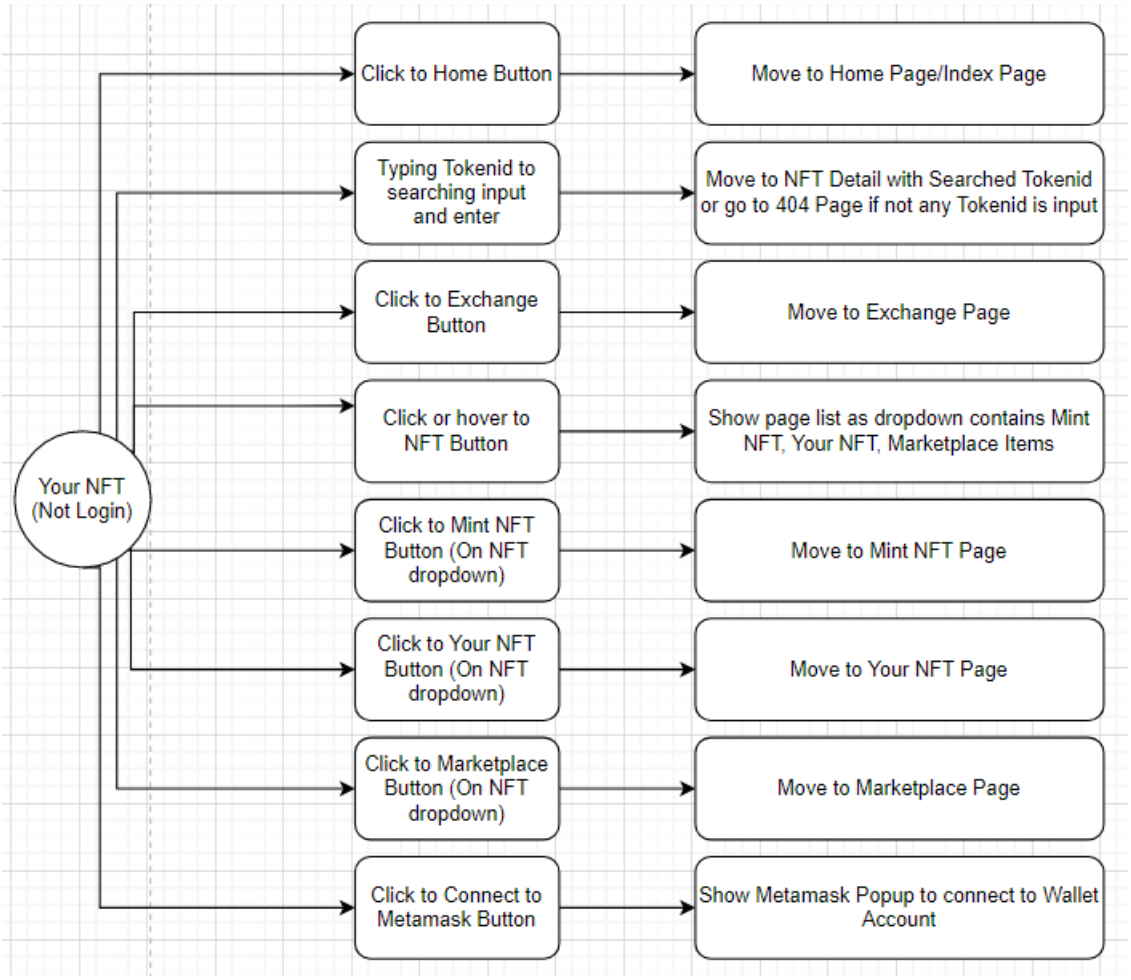

Figure 26: Event handling Your NFT (Not login) diagram

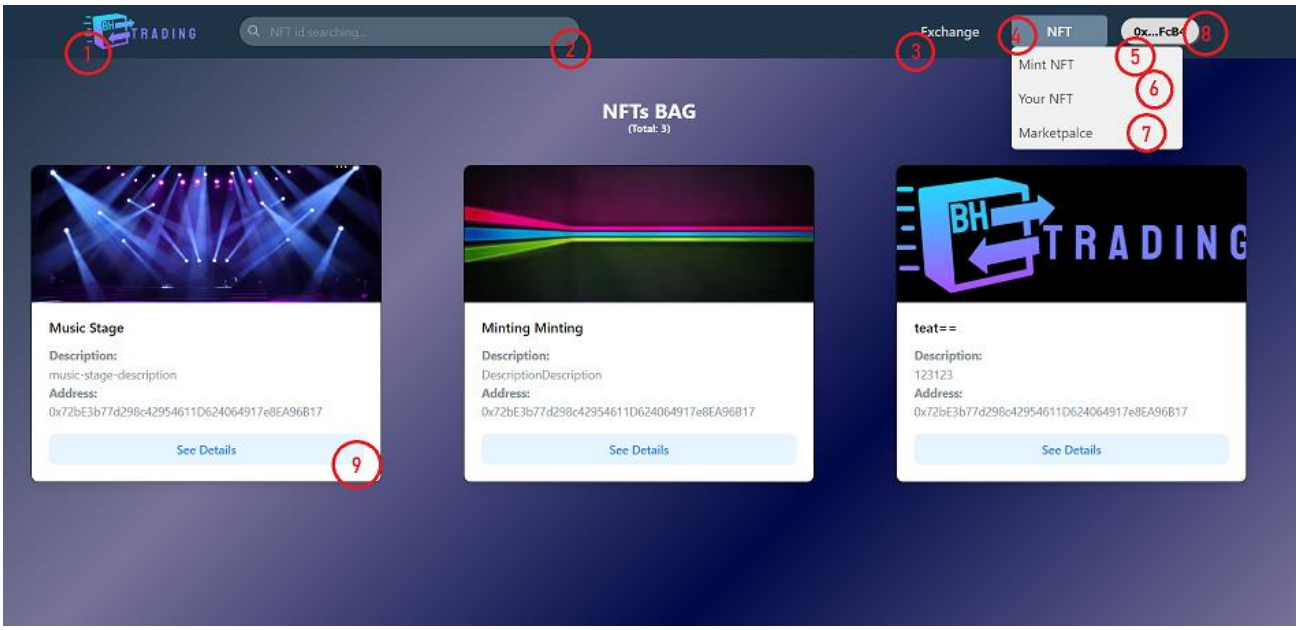

Figure 27: Your NFT Page (Logged In)

Order	Type	Meaning
1	Button (Home)	Move to home page
2	Input (Search)	Searching NFT by using Tokenid
3	Button (Exchange)	Move to exchange page
4	Button (NFT)	Show list as dropdown of 3 pages (Mint NFT, Your NFT, Marketplace)
5	Button (Mint NFT)	Move to Mint NFT page
6	Button (Your NFT)	Move to Your NFT page
7	Button (Marketplace)	Move to Marketplace page
8	Button (Connect Wallet Metamask)	Click to interact to current account
9	Card NFT or Button (See Details)	Click to See details, or Current Card to move to NFT Detail

Table 21. Entities in Your NFT (Logged In)

3.4.4. Market place page

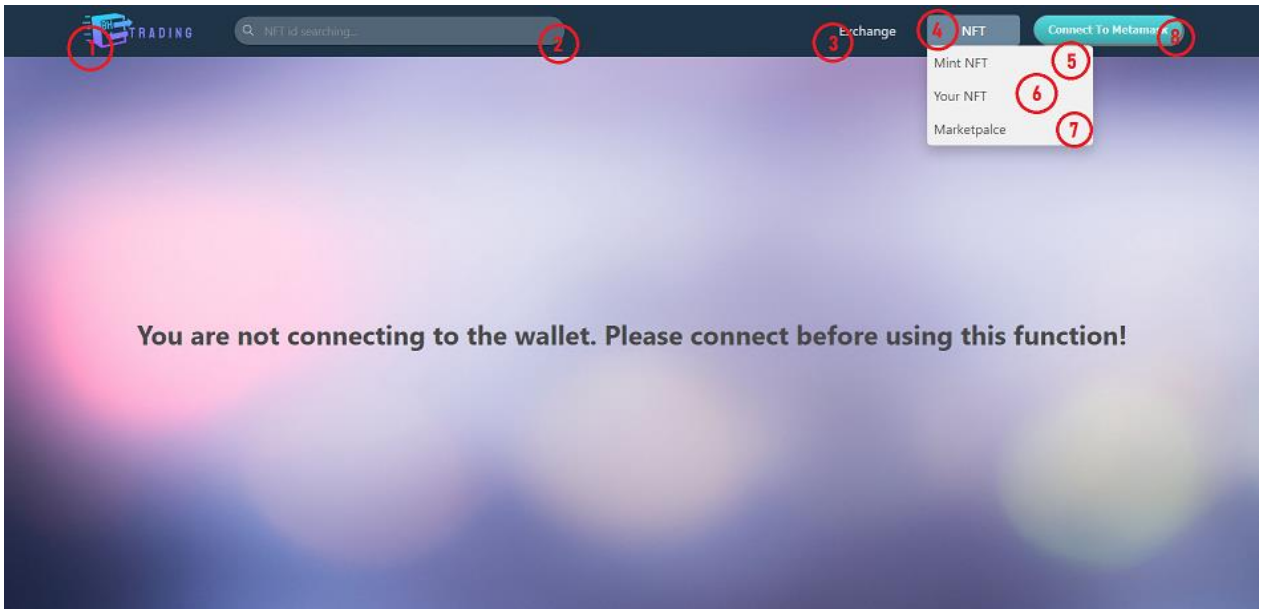

Figure 28: Marketplace Page (Not login)

Order	Type	Meaning
1	Button (Home)	Move to home page
2	Input (Search)	Searching NFT by using Tokenid
3	Button (Exchange)	Move to exchange page
4	Button (NFT)	Show list as dropdown of 3 pages (Mint NFT, Your NFT, Marketplace)
5	Button (Mint NFT)	Move to Mint NFT page
6	Button (Your NFT)	Move to Your NFT page
7	Button (Marketplace)	Move to Marketplace page
8	Button (Connect Wallet Metamask)	Click to appear Metamask Popup to connect Metamask Wallet

Table 22. Entities in Marketplace (Not login)

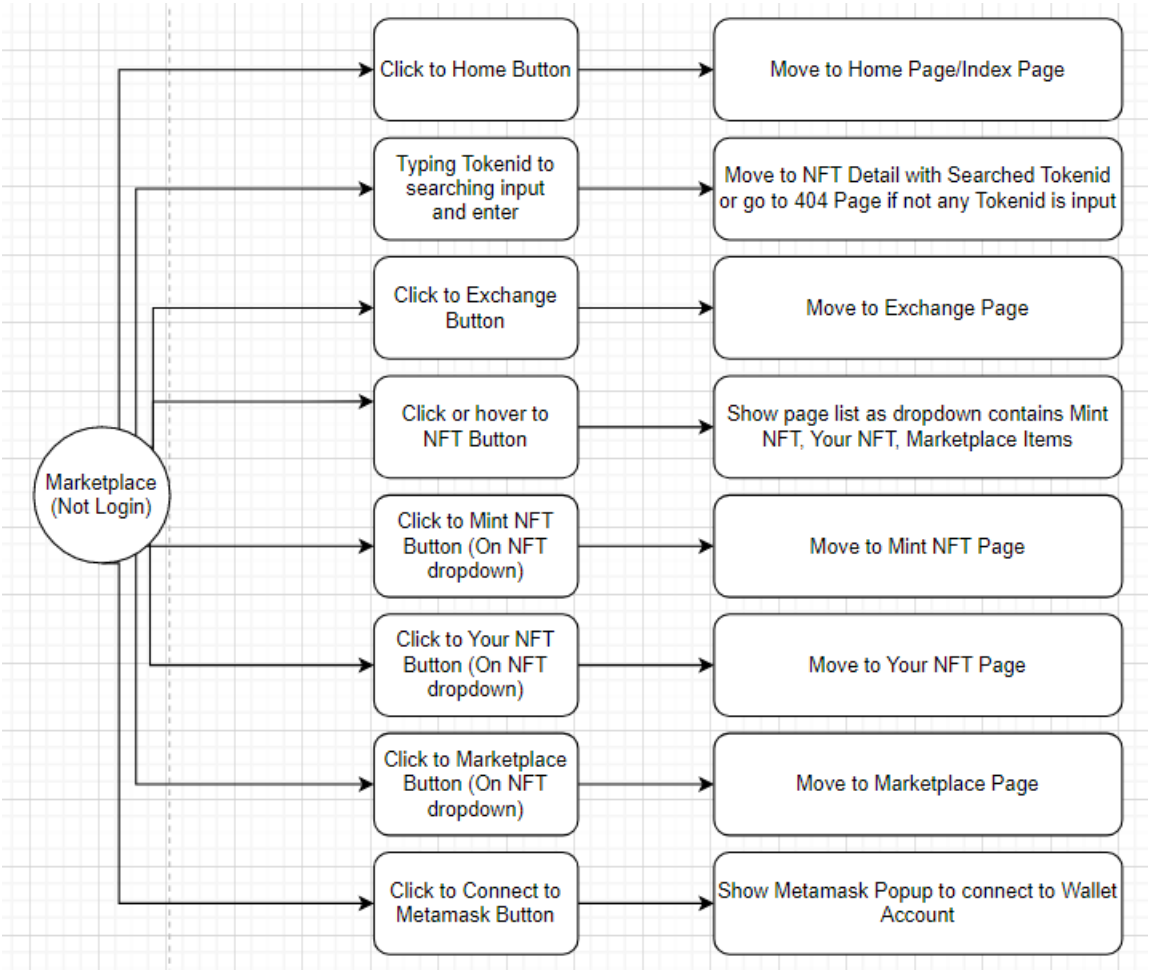

Figure 29: Event handling Marketplace (Not login) diagram

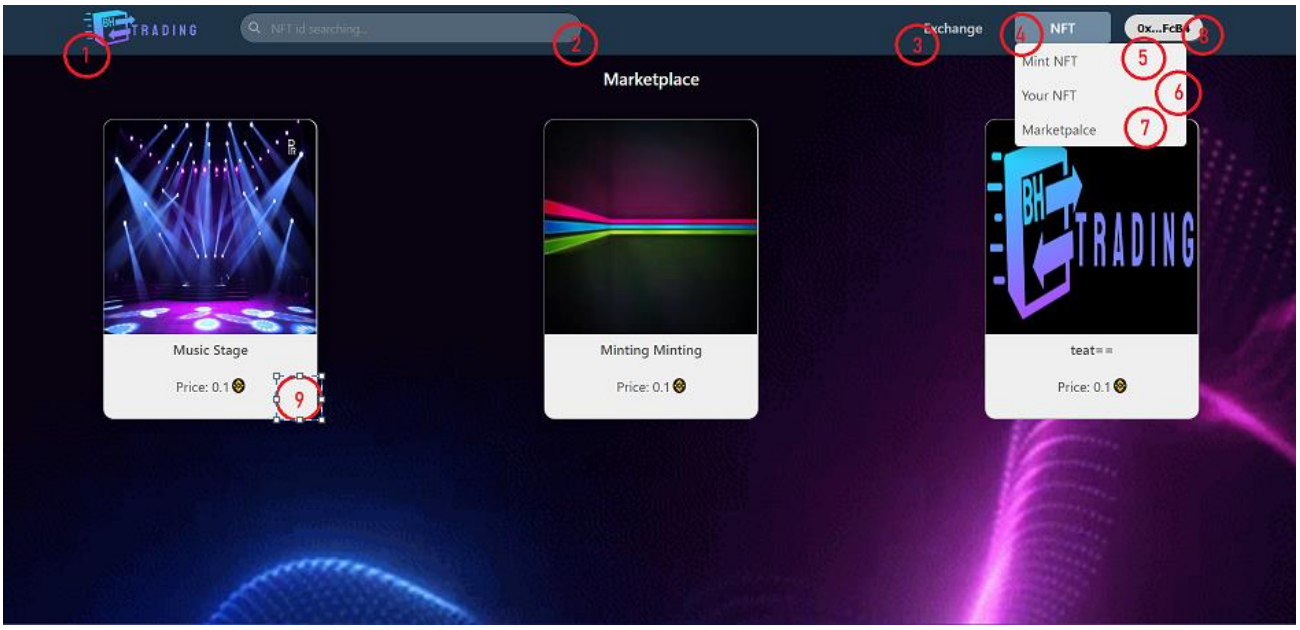

Figure 30: Marketplace page (Logged In)

Order	Type	Meaning
1	Button (Home)	Move to home page
2	Input (Search)	Searching NFT by using Tokenid
3	Button (Exchange)	Move to exchange page
4	Button (NFT)	Show list as dropdown of 3 pages (Mint NFT, Your NFT, Marketplace)
5	Button (Mint NFT)	Move to Mint NFT page
6	Button (Your NFT)	Move to Your NFT page
7	Button (Marketplace)	Move to Marketplace page
8	Button (Connect Wallet Metamask)	Click to interact to current account
9	Card (NFT Detail)	Click to move to NFT Detail Page

Table 23. Entities in Marketplace (Logged In)

3.4.5. Detail NFT Page

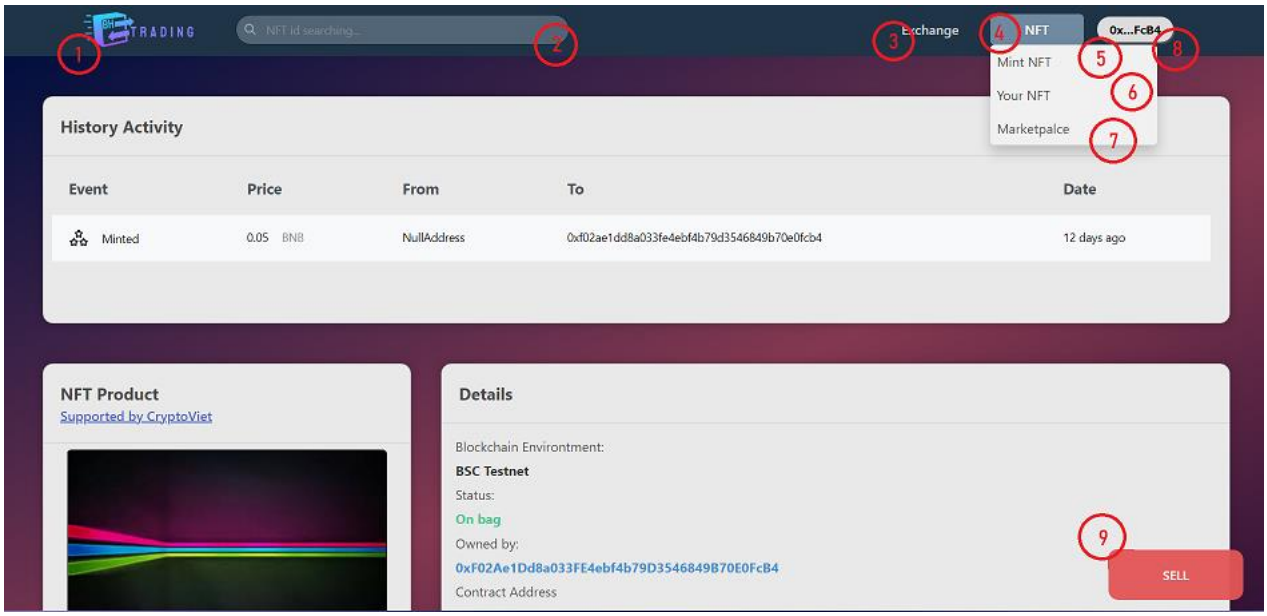

Figure 31: Detail NFT Page (Status: On Bag)

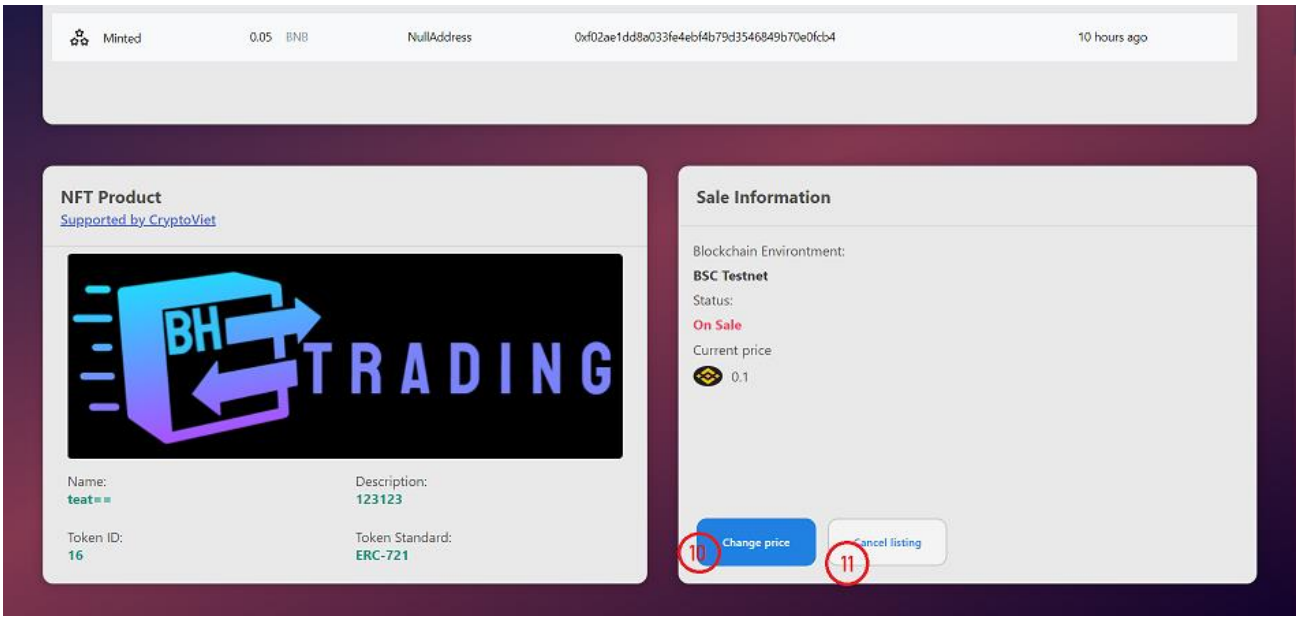

Figure 32: Detail NFT Page (Status: On Sale – Owner Account)

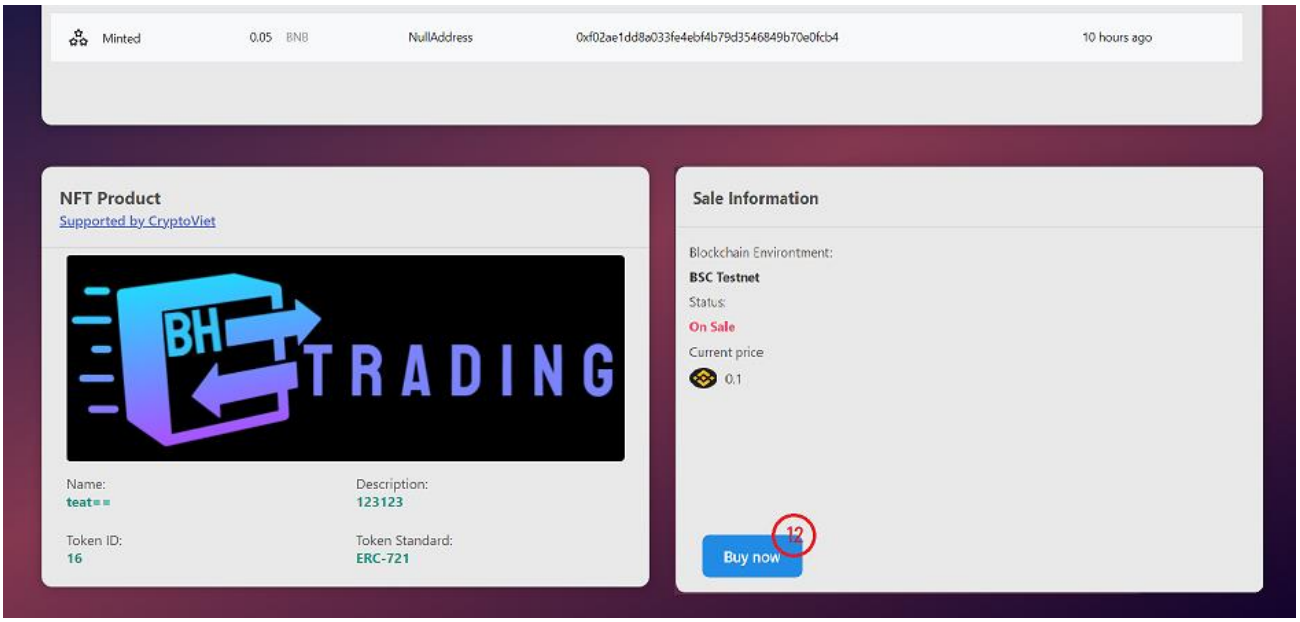

Figure 33: Detail NFT Page (Status: On Sale – Another Account)

Order	Type	Meaning
1	Button (Home)	Move to home page
2	Input (Search)	Searching NFT by using Tokenid
3	Button (Exchange)	Move to exchange page
4	Button (NFT)	Show list as dropdown of 3 pages (Mint NFT, Your NFT, Marketplace)
5	Button (Mint NFT)	Move to Mint NFT page

6	Button (Your NFT)	Move to Your NFT page
7	Button (Marketplace)	Move to Marketplace page
8	Button (Connect Wallet Metamask)	Click to interact to current account
9	Button (Sell) – When NFT is on Bag	Click to Sell current NFT Token and confirm to Sell
10	Button (Change Price) – When NFT is on Bag	Click to show popup change price
11	Button (Cancel Listing) – When NFT is on sale (owner)	Click to cancel list current NFT on marketplace (Status On Sale to On Bag)
12	Button (Buy now) – When NFT is on sale (customer)	Click to Buy NFT on marketplace

Table 24. Entities in Detail NFT Page (Status: On Bag)

3.4.6. Exchange NFT Page

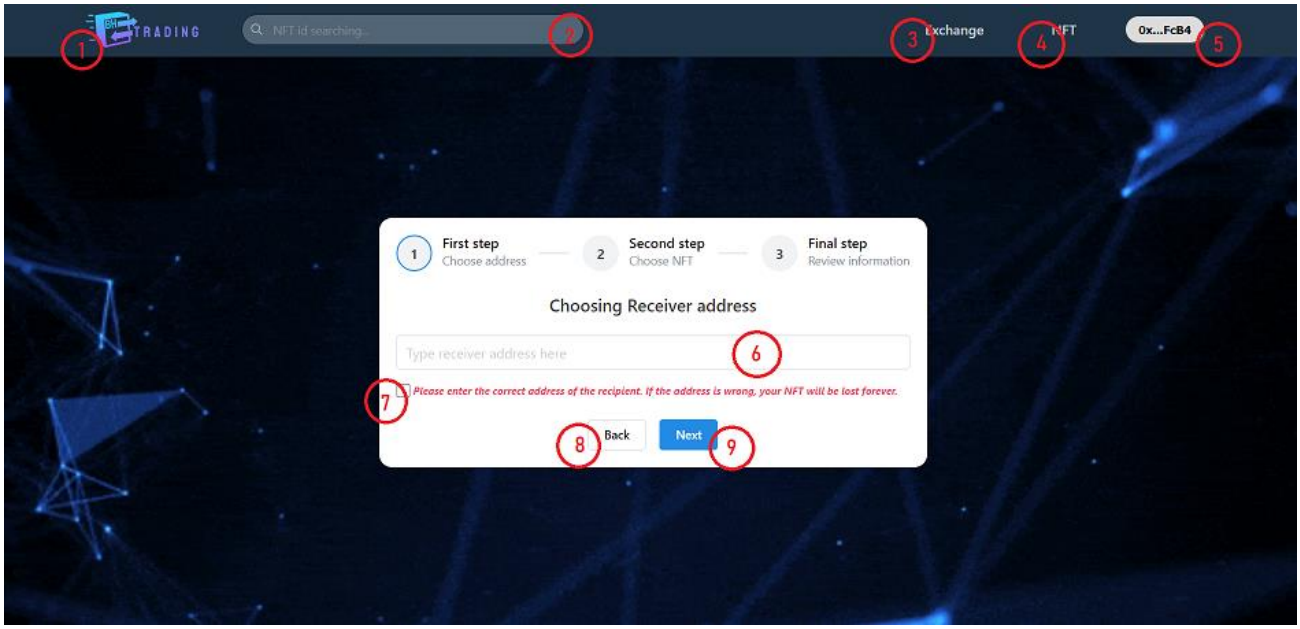

Figure 34: Exchange Page Step 1 (Logged In)

Order	Type	Meaning
1	Button (Home)	Move to home page
2	Input (Search)	Searching NFT by using Tokenid
3	Button (Exchange)	Move to exchange page

4	Button (NFT)	Show list as dropdown of 3 pages (Mint NFT, Your NFT, Marketplace)
5	Button (Connect Wallet Metamask)	Click to interact to current account
6	Input (Address)	Input the address to transfer
7	Checkbox (Confirm)	Click to confirm the transferred address
8	Button (Back) – Not allow if Exchange on step 1	Click to back previous step
9	Button (Next) – Not allow if Exchange on step 3	Click to reach next step

Table 25. Entities in Exchange Page

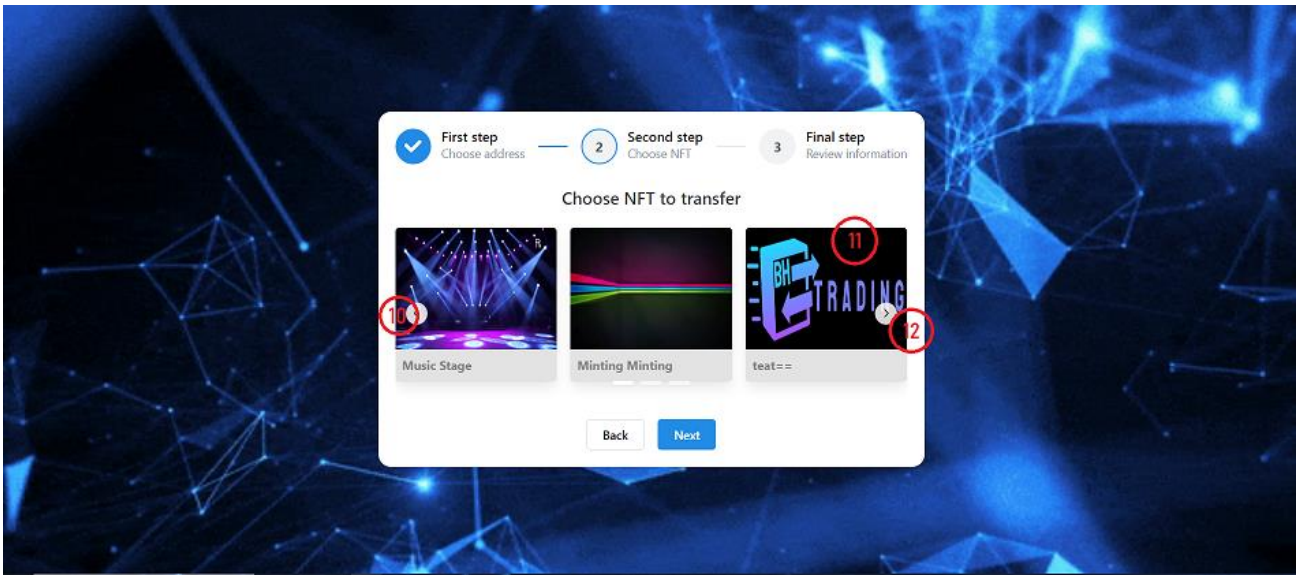

Figure 35: Exchange Page Step 2 (Logged In)

Order	Type	Meaning
10	Button (Left)	View the left NFT
11	Card	Choose the NFT
12	Button (Right)	View the right NFT

Table 26. Entities in Exchange Page

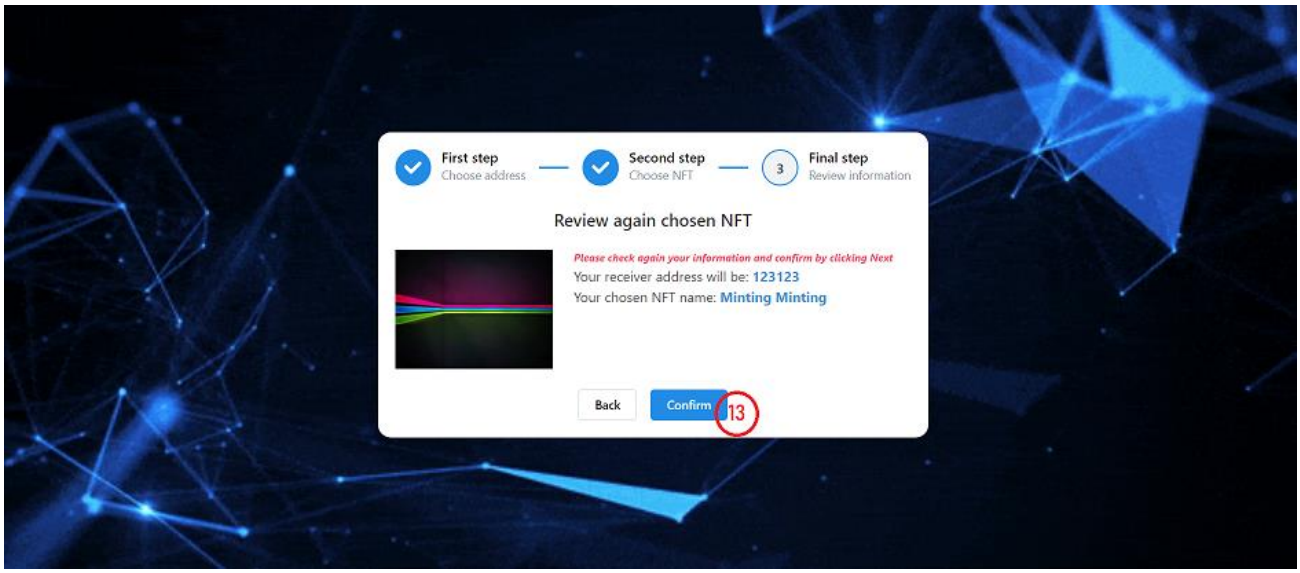

Figure 36: Exchange Page Step 3 (Logged In)

Order	Type	Meaning
13	Button (Confirm)	Click to confirm exchange NFT to the chosen address

Table 26. Entities in Exchange Page

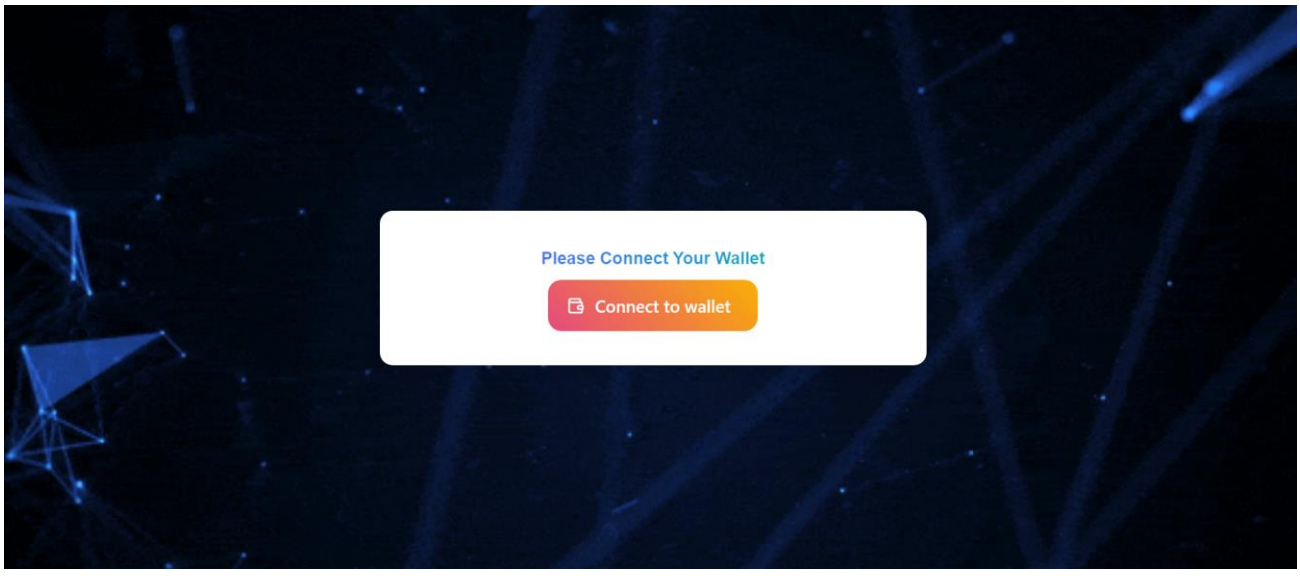

Figure 37: Exchange Page (Not login)

Order	Type	Meaning
13	Button (Connect to Wallet)	Click to appear Metamask Popup to connect Metamask Wallet

CHAPTER 4. SYSTEM ARCHITECTURE AND TECHNOLOGY

4.1. Overall architecture of the system

4.1.1. System architecture

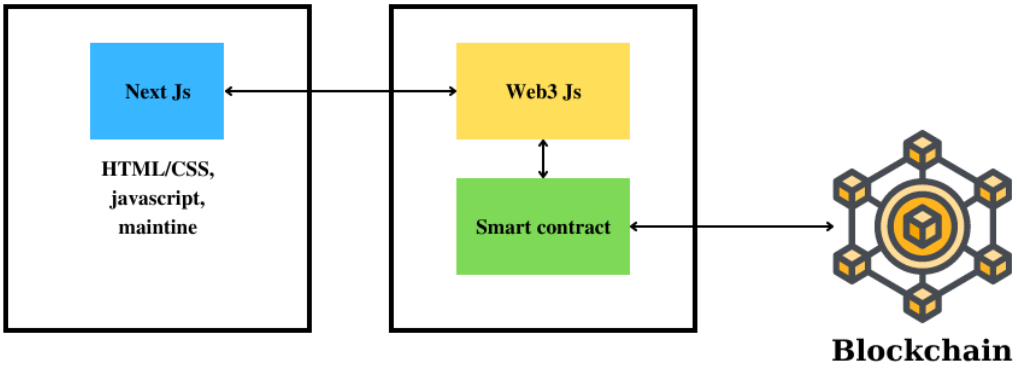

Figure 47: System architecture

4.1.2. File structure

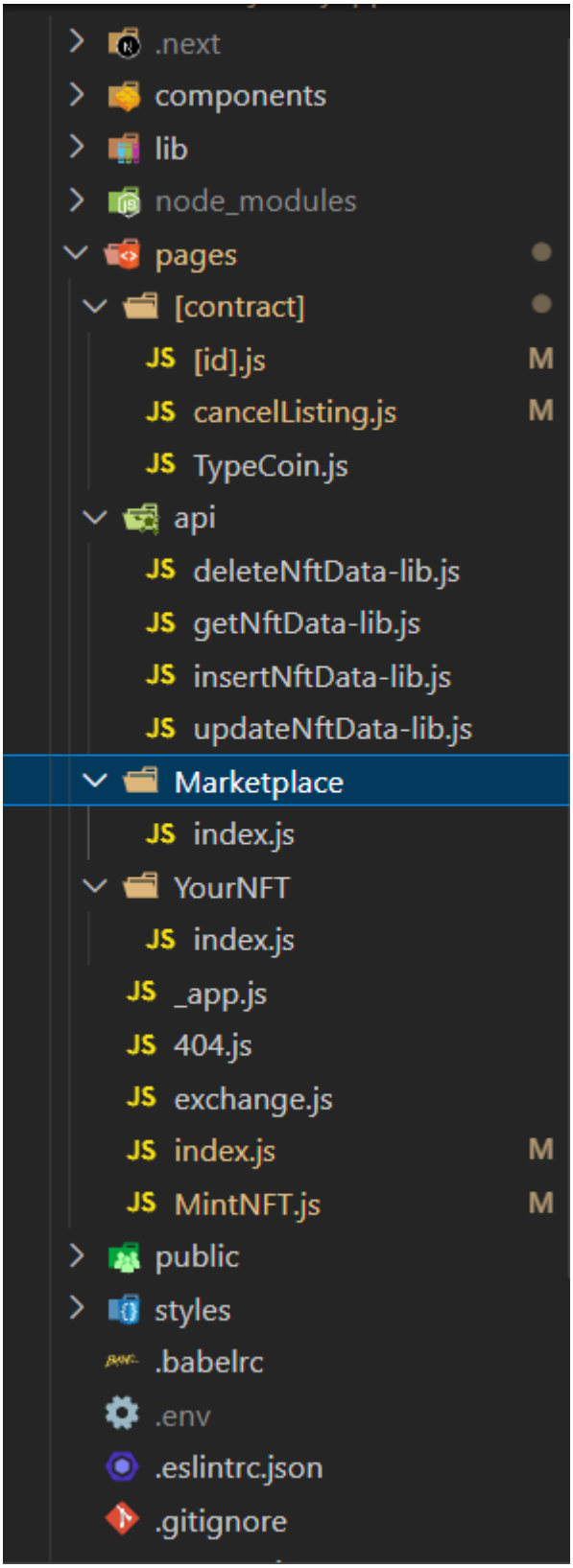

Figure 61: Front end folder organization

4.1.3. Package

4.1.3.1. Front end

Package	Version	Uses
Mantine	5.9.4	Support UI
Axios	0.27.2	Request api to get data from database
Bulma	0.9.4	Support UI
Dotenv	16.0.1	Save environment variable
Moment	2.24.0	interact with date
Cors	2.8.5	collaboration with cors
ethers	5.6.9	Ether library to help for interacting with smart contract
Ipfs-http-client	57.0.3	Help to save metadata into ipfs
Mysql2	4.1.3	Connect to database Mysql
Web3	1.7.4	Interact with smart contract
Web3auth	1.1.1	Help to connect metamask

Table 16: Front end library

4.1.3.2. Smart contract

Package	Version	Uses
openzeppelin	4.7.2	Use ERC-20, ERC-721 standard
truffle		Framework to make smart contract
truffle/hdwallet-provider	2.0.13	Get provider to deploy smart contract
Dotenv	16.0.1	Save environment variable
truffle-plugin-verify	0.5.27	Verify smart contract

Table 17: Smart contract library

4.2. NextJS

4.2.1. What is ReactJs?

- ReactJS is a front-end JavaScript library for creating user interfaces that is open-source. Facebook and a community of individuals and businesses maintain ReactJS. It's commonly used as a foundation for single-page websites and mobile apps. It's simple to use and allows users to develop reusable user interface components.
- By library, we mean React provides helpful functions to build UI, but leaves it up to the developer where to use those functions in their application.
- Part of React's success is that it is relatively unopinionated about the other aspects of building applications. This has resulted in a flourishing ecosystem of third-party tools and solutions.
- It also means, however, that building a complete React application from the ground up requires some effort. Developers need to spend time configuring tools and reinventing solutions for common application requirements

4.2.2. What is NextJs?

- Next.js is a React framework that gives you building blocks to create web applications.
- By framework, we mean Next.js handles the tooling and configuration needed for React, and provides additional structure, features, and optimizations for your application.
- You can use React to build your UI, then incrementally adopt Next.js features to solve common application requirements such as routing, data fetching, integrations - all while improving the developer and end-user experience.
- Whether you're an individual developer or part of a larger team, you can leverage React and Next.js to build fully interactive, highly dynamic, and performant web applications.

4.3. Smart contract

4.3.1. What is blockchain?

4.3.1.1. Blockchain overview

- Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network. An asset can be tangible (a house, car, cash, land) or intangible (intellectual property, patents, copyrights, branding). Virtually anything of value can be tracked and traded on a blockchain network, reducing risk and cutting costs for all involved.
- Business runs on information. The faster it's received and the more accurate it is, the better. Blockchain is ideal for delivering that information because it provides immediate, shared and completely transparent information stored on an immutable

ledger that can be accessed only by permissioned network members. A blockchain network can track orders, payments, accounts, production and much more. And because members share a single view of the truth, you can see all details of a transaction end to end, giving you greater confidence, as well as new efficiencies and opportunities.

4.3.1.2. How blockchain work

- **As each transaction occurs, it is recorded as a "block" of data:** Those transactions show the movement of an asset that can be tangible (a product) or intangible (intellectual). The data block can record the information of your choice: who, what, when, where, how much and even the condition — such as the temperature of a food shipment.
- **Each block is connected to the ones before and after it:** These blocks form a chain of data as an asset moves from place to place or ownership changes hands. The blocks confirm the exact time and sequence of transactions, and the blocks link securely together to prevent any block from being altered or a block being inserted between two existing blocks.
- **Transactions are blocked together in an irreversible chain:** Each additional block strengthens the verification of the previous block and hence the entire blockchain. This renders the blockchain tamper-evident, delivering the key strength of immutability. This removes the possibility of tampering by a malicious actor — and builds a ledger of transactions you and other network members can trust.

4.3.1.3. Benefit of blockchain

- Operations often waste effort on duplicate record keeping and third-party validations. Record-keeping systems can be vulnerable to fraud and cyberattacks. Limited transparency can slow data verification. And with the arrival of IoT, transaction volumes have exploded. All of this slows business, drains the bottom line — and means we need a better way. Enter blockchain:
    - **Greater trust**: With blockchain, as a member of a members-only network, you can rest assured that you are receiving accurate and timely data, and that your confidential blockchain records will be shared only with network members to whom you have specifically granted access.
    - **Greater security:** Consensus on data accuracy is required from all network members, and all validated transactions are immutable because they are recorded permanently. No one, not even a system administrator, can delete a transaction.
    - **More efficiencies**: With a distributed ledger that is shared among members of a network, time-wasting record reconciliations are eliminated. And to speed transactions, a set of rules — called a smart contract — can be stored on the blockchain and executed automatically.

4.3.2. What is smart contract?

4.3.2.1. Smart contract overview

- Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met.

4.3.2.2. How smart contract work

- Smart contracts work by following simple "if/when…then…" statements that are written into code on a blockchain. A network of computers executes the actions when predetermined conditions have been met and verified. These actions could include releasing funds to the appropriate parties, registering a vehicle, sending notifications, or issuing a ticket. The blockchain is then updated when the transaction is completed. That means the transaction cannot be changed, and only parties who have been granted permission can see the results.
- Within a smart contract, there can be as many stipulations as needed to satisfy the participants that the task will be completed satisfactorily. To establish the terms, participants must determine how transactions and their data are represented on the blockchain, agree on the "if/when...then…" rules that govern those transactions, explore all possible exceptions, and define a framework for resolving disputes.
- Then the smart contract can be programmed by a developer – although increasingly, organizations that use blockchain for business provide templates, web interfaces, and other online tools to simplify structuring smart contracts.

4.3.2.3. Benefit of smart contract

- **Speed, efficiency and accuracy**: Once a condition is met, the contract is executed immediately. Because smart contracts are digital and automated, there's no paperwork to process and no time spent reconciling errors that often result from manually filling in documents.
- **Trust and transparency**: Because there's no third party involved, and because encrypted records of transactions are shared across participants, there's no need to question whether information has been altered for personal benefit.
- **Security**: Blockchain transaction records are encrypted, which makes them very hard to hack. Moreover, because each record is connected to the previous and subsequent records on a distributed ledger, hackers would have to alter the entire chain to change a single record.
- **Savings**: Smart contracts remove the need for intermediaries to handle transactions and, by extension, their associated time delays and fees.

4.3.3. What is Solidity?

- Solidity is an object-oriented, high-level language for implementing smart contracts. Solidity is a curly-bracket language designed to target the Ethereum Virtual Machine (EVM). It is influenced by C++, Python and JavaScript.
- Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features. With Solidity you can create contracts for uses such as voting, crowdfunding, blind auctions, and multi-signature wallets.

4.3.4. What is NFT?

4.3.4.1. NFT overview

- NFTs are tokens that we can use to represent ownership of unique items. They let us tokenize things like art, collectibles, even real estate. Ownership of an asset is secured by the Ethereum blockchain – no one can modify the record of ownership or copy/paste a new NFT into existence.
- NFT stands for non-fungible token. Non-fungible is an economic term that you could use to describe things like your furniture, a song file, or your computer. These things are not interchangeable for other items because they have unique properties.
- Fungible items, on the other hand, can be exchanged because their value defines them rather than their unique properties. For example, ETH or dollars are fungible because 1 ETH / $1 USD is exchangeable for another 1 ETH / $1 USD.
- NFTs and Ethereum solve some of the problems that exist in the internet today. As everything becomes more digital, there's a need to replicate the properties of physical items like scarcity, uniqueness, and proof of ownership. Not to mention that digital items often only work in the context of their product. For example you can't re-sell an iTunes mp3 you've purchased, or you can't exchange one company's loyalty points for another platform's credit even if there's a market for it.

4.3.4.2. How NFT work

- NFTs are created through a process called minting in which the information of the NFT is published on a blockchain. At a high-level, the minting process entails a new block being created, the information of the NFT being validated by a validator, and the information being recorded. This minting process often entails incorporating smart contracts that assign ownership and manage the transferability of the NFT.
- As tokens are minted, they are assigned a unique identifier directly linked to one blockchain address. Each token has an owner, and the ownership information (i.e. the address in which the minted token resides) is publicly available. Even if 5,000 NFTs of the same exact item are minted (i.e. general admission tickets to a music festival), each of the tickets has a unique identifier and can be distinguished from one another.

4.4. IPFS

4.4.1. IPFS overview

- The InterPlanetary File System (IPFS) is a decentralized distributed file system that enables content-addressed storage. IPFS defines itself as "a peer-to-peer hypermedia protocol designed to preserve and grow humanity's knowledge by making the web upgradeable, resilient, and more open."
- The IPFS protocol stands in direct contrast to today's widely used HTTP system. HTTP is a centralized, location-addressed protocol. In HTTP, users retrieve resources (such as HTML documents) based solely on their location on centralized servers. This system features several disadvantages.
- When resources live on a single device, that device is vulnerable to many threats. A content-addressed distributed file system like IPFS has several advantages over HTTP.
- It's possible to achieve these advantages in your IT environment, but it's crucial to build the infrastructure to support IPFS first. Here are five major advantages of IPFS.

4.4.2. How IPFS work

- IPFS is a peer-to-peer (p2p) storage network. Content is accessible through peers located anywhere in the world, that might relay information, store it, or do both. IPFS knows how to find what you ask for using its content address rather than its location.
- There are three fundamental principles to understanding IPFS:
    - Unique identification via content addressing
    - Content linking via directed acyclic graphs (DAGs)
    - Content discovery via distributed hash tables (DHTs)

4.4.3. Benefit of IPFS

- IPFS Uses a Decentralized Protocol:
    - The current location-based HTTP system is subject to several disadvantages. All resources in HTTP are housed on a centralized single server. That server is always vulnerable to cybersecurity threats, such as DDoS attacks. It is also subject to failure or poor performance. And that server can be inactivated or censored based on what entity manages it, thereby eliminating the only source of the data housed on it.
    - IPFS is a decentralized system that uses content-addressed storage. In this protocol, each piece of data is assigned a unique content identifier (CID). All content-addressed data in IPFS can be found and retrieved based on this unique CID.
    - IPFS content is housed in several locations in a shared, peer-to-peer network using a distributed hash table (DHT). This decentralized protocol

means there is no one point of failure in a single server. And no one entity can censor or eliminate the data.

- IPFS Offers Enhanced Security:
    - Not only is each resource in IPFS assigned a unique CID, but these CID files are immutable and cannot be altered by any third-party entity. IPFS also uses transport-encryption to keep data secure when being sent from one IPFS node to another.
    - This creates huge security advantages over HTTP. The immutable nature of resources in IPFS greatly reduces many cybersecurity threats.
- IPFS Enables High Performance:
    - Storing and distributing data with IPFS saves bandwidth by retrieving data from multiple peers at once. A user requests content based on its unique CID, IPFS retrieves that data based on the CID from multiple nodes at once, and it is then delivered to the user in the quickest, most efficient way possible.
    - Content is uncoupled from distant servers and stored closer to the users. This content retrieval method has been shown to save up to 60% bandwidth for video. IPFS can efficiently distribute high volumes of data quickly and without duplication.
- IPFS Enables Easier Deduplication and Archiving:
    - The CID created with IPFS provides a digital fingerprint that can ensure authenticity and uniqueness. This makes deduplication simpler by creating a single instance of data with an immutable CID. And because there is only one copy of each resource, authentication is also much simpler.
    - Since there's no duplication, IPFS minimizes the storage space consumed by data backups and archives. This creates a huge advantage for any organization archiving their data.
- IPFS Allows Improved Content Control:
    - IPFS provides much more control to content creators. Creators can distribute their work themselves without being dependent on a content distributor entity.
    - Creators also don't have to spend money on servers to control their content. With IPFS, anyone can make their content available in the network, and anyone in the world can receive that content securely.
- Because of these benefits, most of the information of the NFTs is stored on IPFS to reduce the cost of minting an NFT, and can store NFT information data in a safe and easily accessible way

4.5. Truffle

4.5.1. Truffle overview

- Truffle is a world-class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM), aiming to make life as a developer easier.
- Truffle is widely considered the most popular tool for blockchain application development with over 1.5 million lifetime downloads. Truffle supports developers across the full lifecycle of their projects, whether they are looking to build on Ethereum, Hyperledger, Quorum, or one of an ever-growing list of other supported platforms. Paired with Ganache, a personal blockchain, and Drizzle, a front-end dApp development kit, the full Truffle suite of tools promises to be an end-to-end dApp development platform.
    - Built-in smart contract compilation, linking, deployment and binary management.
    - Automated contract testing for rapid development.
    - Scriptable, extensible deployment & migrations framework.
    - Network management for deploying to any number of public & private networks.
    - Package management with EthPM & NPM, using the ERC190 standard.
    - Interactive console for direct contract communication.
    - Configurable build pipeline with support for tight integration.
    - External script runner that executes scripts within a Truffle environment.

4.5.2. How Truffle work

- Truffle is the most popular development tooling for Ethereum programmers. Easily deploy smart contracts and communicate with their underlying state without heavy client side programming. An especially useful library for the testing and iteration of Ethereum smart contracts.

4.6. Web3Js

- Web3.js is a collection of libraries that allow developers to interact with a remote or local Ethereum node using HTTP, IPC, or WebSocket. Using this library, you can develop websites or clients that interact with the blockchain. This can be actions like sending Ether from one user to another, checking data from smart contracts, creating smart contracts, among other things.
- Ethereum nodes provide interfaces to users in order to complete transactions: of which, nodes receive this information through a JSON RPC interface. This is an encoding format that allows running processes to receive new and verify existing data. Web3.js helps to make the process of running and selecting nodes participating in the Ethereum network simpler and easier to grasp.

- Web3.js itself represents a JavaScript language binding for the aforementioned JSON RPC interface. This allows for the library to be inherently usable and flexible (since Javascript is natively supported in most popular web browsers), but also allows use on the server side in Node.js applications or Electron-based ones. The most common use is the one frequently stated in this piece, however: as accessing the Ethereum blockchain seems to be the chief reason that someone seeks out Web3.js. Users utilize Ethereum nodes via HTTP, which can either be locally hosted, by a Dapp (decentralized application) provider, or through public gateways like Moralis or Infura.

4.7. MySQL

4.7.1. MySQL overview

- MySQL is a relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL).
- A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or a place to hold the vast amounts of information in a corporate network. In particular, a relational database is a digital store collecting data and organizing it according to the relational model. In this model, tables consist of rows and columns, and relationships between data elements all follow a strict logical structure. An RDBMS is simply the set of software tools used to actually implement, manage, and query such a database.
- MySQL is integral to many of the most popular software stacks for building and maintaining everything from customer-facing web applications to powerful, data-driven B2B services. Its open-source nature, stability, and rich feature set, paired with ongoing development and support from Oracle, have meant that internet-critical organizations such as Facebook, Flickr, Twitter, Wikipedia, and YouTube all employ MySQL backends.

4.7.2. Benefit of MySQL

- MySQL is widely compatible:
    - Though often associated with internet applications or web services, MySQL was designed to be extensively compatible with other technologies and architectures. The RDBMS runs on all major computing platforms, including Unix-based operating systems, such as the myriad Linux distributions or Mac OS, and Windows.
    - MySQL’s client-server architecture means it can support a variety of backends, as well as different programming interfaces. Data can be directly migrated from MySQL to its forks (e.g. MariaDB), as well as most other RDBMSs thanks to architectural and language similarities.
    - Established Oracle and third-party migration tools further allow MySQL to move data to and from a vast set of general storage systems, whether these are designed to be on-premises or cloud-based. MySQL can be

deployed in virtualized environments, distributed or centralized, and even exists as portable standalone libraries for learning purposes, testing, or small applications.
    - MySQL's wide compatibility with all these other systems and software makes it a particularly practical choice of RDBMS in most situations.
- MySQL databases are relational:
    - The primary factor differentiating relational databases from other digital storage lies in how data is organized at a high level. Databases like MySQL contain records in multiple, separate, and highly codified tables, as opposed to a single all-encompassing repository, or collections of semi- or unstructured documents.
    - This allows RDBMSs to better optimize actions like data retrieval, updating information, or more complex actions like aggregations. A logical model is defined over all of the contents of the database, describing for example the values allowed in individual columns, characteristics of tables and views, or how indices from two tables are related.
    - Relational models have remained popular for several reasons. They empower users with intuitive, declarative programming languages — essentially telling the database what result is wanted in language akin to, or at least comprehensible as, written english, instead of meticulously coding up each step of the procedure leading to that result. This moves a lot of the work into the RDBMS and SQL engines, better enforcing logical rules and saving valuable resources and manpower.
- MySQL is open-source:
    - Any individual or enterprise may freely use, modify, publish, and expand on Oracle's open-source MySQL code base. The software is released under the GNU General Public License (GPL).
    - For MySQL code needing to be integrated or included in a commercial application (or if open-source software is not a priority), enterprises can purchase a commercially licensed version from Oracle.
    - Again, these options provide organizations with additional flexibility if deciding to work with MySQL. The public and community-based nature of open-source releases enriches MySQL's documentation and online support culture, while also ensuring that sustained or newly-developed capabilities never stray too far from current user needs.
- MySQL is easy to use:
    - Though MySQL's relational nature and the ensuing rigid storage structures might seem restrictive, the tabular paradigm is perhaps the most intuitive, and ultimately allows for greater usability.

- In fact, MySQL makes many concessions to supporting the widest possible variety of data structures, from the standard but rich logical, numeric, alphanumeric, date, and time types, to more advanced JSON or geospatial data. Beyond mere data types and an expansive built-in feature set, the MySQL ecosystem also includes a variety of tools, easing everything from server management to reporting and data analysis.
- Regardless of the RDBMS's overarching architecture, users can invariably find a MySQL feature allowing them to model and codify data how they wish. MySQL remains one of the most straightforward database technologies to learn and use.

4.8. Implementation

4.8.1. Tools

- Visual Studio Code
- Postman

4.8.2. Technology

- NextJs
- Solidity
- Web3Js
- IPFS
- Truffle
- MySQL

4.8.3. Hardware

- Laptop ASUS TUF Gaming F15 FX506HCB HN1138W
-

4.8.4. Version control

- Github: https://github.com/qhoangf/My-dapp

CHAPTER 5. SOFTWARE TESTING

The test object is a learning website that fully meets the basic needs of users. This test plan is applicable for testing the typical functionalities specified in the document. Here is all test functions of our website:

ID	Description	Test Steps	Test Data	Actual Result	Status
Market place_ 01	Test if wallet popup will appear	• Click "Start Trading NOW"		As Expected	Pass
Market place_ 02	Test if wallet Metamask appear when click wallet item on popup	• Click "Start Trading NOW" • Click "Metamask"		As Expected	Pass
Market place_ 03	Test if wallet Metamask when connecting will display Loading	• Click "Start Trading NOW" • Click "Metamask" • Popup switch to Loading Popup		As Expected	Pass
Market place_ 04	Check if the loading popup when connected will switch to the chosen wallet popup	• Click "Start Trading NOW" • Click "Metamask" • Popup switch to Loading Popup • Connected to the Metamask	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Market place_ 05	Check if the loading popup when connected will switch to the chosen wallet popup	• Click "Start Trading NOW" • Click "Metamask" • Popup switch to Loading Popup • Close Metamask popup	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Market place_ 06	Check if click JOIN WITH US NOW	• Clicking on JOIN WITH US		As Expected	Pass
Market place_ 07	Check if the Hover NFT display dropdown	• Mouse hover on NFT		As Expected	Pass
Market place_ 08	Check if click on home icon will switch to index page	• Click BH Trading logo		As Expected	Pass

Marketplace_09	Check warning if go to page MINT NFT but not login	• Hover NFT • Click MINT NFT		As Expected	Pass
Marketplace_10	Check warning if go to page YOUR NFT but not login	• Hover NFT • Click YOUR NFT		As Expected	Pass
Marketplace_11	Check if popup Metamask display	• Click "Connect to Metamask"		As Expected	Pass
Marketplace_12	Check warning if go to page MARKETPLACE but not login	• Hover NFT • Click MARKETPLACE		As Expected	Pass
Marketplace_13	Check if user can go to page MARKETPLACE when login	• Hover NFT • Click MARKETPLACE		As Expected	Pass
Marketplace_14	Check if user can go to page MINT NFT when login	• Hover NFT • Click MINT NFT		As Expected	Pass
Marketplace_15	Check if user can go to page YOUR NFT when login	• Hover NFT • Click YOUR NFT		As Expected	Pass
Marketplace_16	Check if page exchange can reachable when login	• Click "Exchange"		As Expected	Pass
Marketplace_17	Check if page exchange cannot reachable when not login yet	• Click "Exchange"		As Expected	Pass
Marketplace_18	Cannot go next step when not input anything	• Click "Exchange" • Click "Next"	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_19	Cannot go next step when not input address	• Click "Exchange" • Click checkbox • Click "Next"	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_20	Cannot go next step when not tick checkbox	• Click "Exchange" • Input address • Click "Next"	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass

Market place_ 21	Click step 1 if anything happen	• Click "exchange" • Click step 1	Wallet address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd	As expected	Pass
Market place_ 22	Click step 2 if anything happen	• Click "exchange" • Click step 2	Wallet address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd	As expected	Pass
Market place_ 23	Click step 3 if anything happen	• Click "exchange" • Click step 3	Wallet address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd	As expected	Pass
Market place_ 24	Switch step 2 when input all required things	• Click "Exchange" • Input address • Tick checkbox • Click "Next"	Wallet address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd		
Market place_ 25	Click back when step 1	• Click "exchange" • Input address • Tick checkbox • Click "back"	Wallet address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd	As expected	Pass
Market place_ 26	Click back when step 2	• Click "Back"	Wallet address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd	As Expected	Pass
Market place_ 27	Click Next when choose nothing on step 2	• Click "Next"	Wallet address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd	As Expected	Pass
Market place_ 28	Choose NFT card on step 2	• Click any NFT card	Wallet address: 0x11d4e0365f2d827d0eb 957edfdec3cf109fa90dd	As Expected	Pass

			Address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd		
Marketplace_29	Choose 2 NFT card on step 2	• Click any NFT card • Click another NFT card	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_30	Choose NFT card on step 2 and click Next	• Click any NFT card • Click "next"	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_31	Show correct chosen information on step 3 when completed step 2	• Click any NFT card • Click "next" • Show chosen information	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_32	Go to page MINT NFT and MINT some NFT	• Choose image and input all required things • Click MINT NFT NOW • Show successful minting	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_33	Go to page MINT NFT and MINT some NFT when not choose image	• Choose image and input all required things • Click MINT NFT NOW • Show unsuccessful minting	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd Address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd Description: test description	As Expected	Pass
Marketplace_34	Check if the preview image will active	• Choose any image • Preview picture will appear the chosen image	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd Image: test.jpg	As Expected	Pass
Marketplace_35	Showing number of data on bag when being on your NFT page	• See the number of having NFT Card	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass

Marketplace_36	Click on see detail NFT Item	• Click "See detail"	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_37	Click on card NFT Item	• Click on card NFT	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_38	See the price when being on MARKETPLACE page	• Get data from backend • See the price of each item	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_39	Go on page detail if the NFT is on bag	• Show information about NFT • Show NFT details	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass
Marketplace_40	Go on page detail if the NFT is on sale	• Show information about NFT • Show sale information details	Wallet address: 0x11d4e0365f2d827d0eb957edfdec3cf109fa90dd	As Expected	Pass

CHAPTER 6. CONCLUSION

Over the period of completing the graduation thesis, the group has successfully completed our objectives. In the process of doing the thesis, the group also learned and absorbed a lot of new knowledge through building the "Decentralized NFT Marketplace website" application using NextJs, in addition, there was good coordination among members while working on the project. job. Besides the results achieved, the application still has many things to overcome and improve, these are also the future development directions that the team is aiming for.

6.1. Achievement

- Know the theory of NextJs, Blockchain, Solidity, Web3Js, MySQL. Proficient use of add-on libraries. Learn how to build front-end websites, build smart contracts and build databases and connect them.
- Gain effective teamwork skills. Logical thinking is also significantly improved. Know how to start a project effectively and divide the work evenly among team members.
- Project: A Decentralized NFT Marketplace website with all functions that work well. Beautiful interface and easy to use.

6.2. Strengths and drawbacks

- The website is designed with an easy-to-see and user-friendly interface. Simple operation and functions are easy to use with all objects. Can mint, transfer, sell and buy the NFT easily.
- On the other hand, Because of blockchain is still a very new technology, this project will be difficult for those who are new or inexperienced to experience the blockchain market to access. Not only that, the project also requires our team to make efforts to learn about these new technologies on their own.

6.3. Reflection gained

- Learn how to manage source control, very good support in doing multi-person projects.
- Learn new technology

6.4. Future development direction

- Build a NFT collection.
- Build some stats
- Improve website performance.

- The team will work to improve the application and add more features in the future.
- Enhance the user interface and experience.

REFERENCES

[1]. Blockchain: https://www.ibm.com/topics/what-is-blockchain
[2]. NextJs https://nextjs.org/
[3]. Web3Js https://web3js.readthedocs.io/en/v1.8.1/
[4]. Truffle https://trufflesuite.com/
[5]. Online learning Solidity https://cryptozombies.io/
[6]. IPFS https://docs.ipfs.tech/
[7]. Package management https://npmjs.com/
[8]. Udemy - course online https:/udemy.com/
[9]. MySQL https://www.mysql.com/
[10]. Openzeppelin https://www.openzeppelin.com/
[11]. Stackoverflow - IT developers forum https://stackoverflow.com/
[12]. Binance Smart Chain Testnet https://testnet.bscscan.com/