**คู่มือสำหรับนักพัฒนา**

เทคโนโลยีที่โดยรวมของระบบ NFT-Marketplace จะมีการพัฒนาอยู่ 3 ส่วนดังนี้ frontend, backend, smart contract

1. **Frontend**

frontend โดยระบบของ NFT-Marketplace ใช้ library React , Bootstrap สำหรับการทำ UI และทำการใช้ Redux เพื่อจัดการ state ในระบบ NFT-Marketplace โดยจะเห็นได้ว่าในระบบจะมีโครงสร้างทั้งหมดดังนี้

A screenshot of a computer

Description automatically generated with medium confidence

**รูปที่ 1 frontend directory**

ในภาพไฟล์ package สำหรับ project, ไฟล์ config ของ firebase สำหรับการเรียกใช้การสำหรับการ deploy ,ไฟล์ config สำหรับ tailwind css และจะมี folder อยู่ทั้งหมด 3 folder ดังนี้

* 1. github folder

**A black background with white text

Description automatically generated with low confidence**

**รูปที่ 2 workflows สำหรับ deploy**

ซึ่งทั้ง 2 ไฟล์นี้เป็นไฟล์สำหรับการทำ workflow สำหรับการทำ auto deploy สำหรับตัว project ใน github

* 1. public folder

A screen shot of a computer

Description automatically generated with medium confidence

**รูปที่ 3 public folder สำหรับแสดง interface**

ใน folder public จะไฟล์ index.html ที่ใช้สำหรับการเรียกใช้ไฟล์ root ของ react เพื่อแสดง UI และมีไฟล์ favicon.png เป็น logo ใน header ของเว็บ และไฟล์ 404.html แสดงผลการไม่เจอหน้า UI และใน folder images คือ folder ที่เอาไว้เก็บรูปภาพที่เกี่ยวข้องกับตัวหน้าเว็บ

* 1. src folder

**A screenshot of a computer

Description automatically generated**

**รูปที่ 4 src directory**

จากภาพเป็นโครงสร้างใน folder ของ src โดยที่มีการทำงานจากการ render ใน file index.ts ในไฟล์ นี้จะเป็นการ render root ที่มี component App ภายใน provider ของ redux store และมีการ import bootstrap css framework ที่ใช้สำหรับการ styling UI

**1.1 Store ของระบบ**

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 5 store ของระบบ**

ในไฟล์ store/index.tsx นี้จะมีการ import slice ที่ใช้ในระบบดังนี้ userAccount, transactionRes, jwt โดยมีการกำหนดชื่อของ state สำหรับการนำข้อมูลไปเก็บไว้ใน localstorage และทำการสร้าง configureStore ในชื่อ store โดยนำ slice ที่ import มาสร้าง reducer และกำหนด middleware โดยนำ state ที่เราสร้างมา save ข้อมูลเก็บไว้ใน localstorage

Reducer ทั้ง 3 ที่ได้ import มีดังนี้

1. UserAction
2. TransactionAction
3. JWTAction

A screenshot of a computer

Description automatically generated with medium confidence

**รูปที่ 6 โครงสร้างของ ruducer**

ใน UserAction มีไฟล์อยู่ 2 ไฟล์ คือ index.tsx, hook.tsxโดยที่ไฟล์ hook มีไว้สำหรับการเรียกใช้งานข้อมูลหรือ function สำหรับการเขียนข้อมูลใน state สำหรับ Object ส่วนไฟล์ index.tsx จะเป็นการจัดการข้อมูลใน state ของ store และ folder store มีไฟล์ type.ts เพื่อใช้ export interface ดังนี้

A screen shot of a computer screen

Description automatically generated with low confidence

**รูปที่ 7 interface for store**

**1.2 Component App**

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 8 App.tsx สำหรับการจัดวาง frontend**

สำหรับ app.tsx ในโฟลเดอร์ src นั้นจะเป็นส่วนเริ่มต้นของ react โดยมีการสร้าง tag Header เพื่อใช้ในทุก route ซึ่งจะทำหน้าที่ path ไปยัง component อื่นซึ่งจะประกอบด้วย component ดังต่อไปนี้

A screenshot of a computer

Description automatically generated with medium confidence

**รูปที่ 9 component structure**

โดยในโครงสร้างของ component structure แต่ละ component จะมีหน้าที่ดังต่อไปนี้

1. BlankCard มีหน้าที่ในการสร้าง BlankCard เพื่อรองรับการ fetch ข้อมูล
2. boxComponent มีหน้าที่สำหรับการสร้าง Card สำหรับแสดงข้อมูล NFT และ Collection
3. CreateNFT มีหน้าที่ในการ render หน้า createNFT สำหรับการสร้าง NFT
4. Explore เป็น Component เพื่อใช้ในการปรับแต่งของหน้า ExploreCollection และ ExploreNFT
5. ExploreCollection มีหน้าที่ในการ render เพื่อแสดงข้อมูล Collection
6. ExploreNFT มีหน้าที่ในการ render เพื่อแสดงข้อมูล NFT
7. Footer มีหน้าที่ในการ render ส่วนของ Footer ให้มีพื้นที่ว่างที่เหมาะสม
8. Header มีหน้าที่ในการ render Navbar สำหรับการ Search และการ path
9. Home มีหน้าที่ในการ render หน้าหลักของ Application
10. MyCollection มีหน้าที่ในการ render หน้าของผู้ใช้งานสำหรับ MyCollection
11. Profile มีหน้าที่ในการ render ส่วนของหน้าผู้ใช้งาน
12. Search มีหน้าที่ในการ render และจัดการการรับ parameters ในส่วนของการ Search
13. SellNFT มีหน้าที่ในการ render หน้าสำหรับการตั้งราคาขาย
14. Setting มีหน้าที่ในการ render หน้าสำหรับการตั้งค่าเปลี่ยนโปรไฟล์ของ User
15. ViewNFT มีหน้าที่ในการ render หน้าสำหรับการแสดง NFT
16. WaitTransaction มีหน้าที่ในการ render ส่วนของ Modal Popup สำหรับรอการยืนยัน Transaction

โดยในแต่ละ component นั้นจะมีโครงสร้างหลัก ๆ ประกอบด้วย index.tsx ทำหน้าที่จัดการในส่วนของ HTML และ TypeScript โดยที่ frontend จะมี hook ในการทำ action หรือเรียกใช้งาน services ต่าง ๆ นอกเหนือจากนี้ในบาง component จะมี style เป็นของตัวเองจึงได้มี .css เพื่อใช้เก็บ style สำหรับ component นั้นๆ

**1.3 Hook and services**

A screenshot of a computer

Description automatically generated with medium confidence

**รูปที่ 10 hook structure**

สำหรับ Hook and Service นั้นจะเป็นส่วนที่ทำหน้าที่ในการทำ logic ต่าง ๆ ที่เกี่ยวข้องกับการติดต่อไปยัง backend, IPFS และส่วนของ smart contract ซึ่งจะแบ่งประเภทออกมาดังต่อไปนี้

1. useAddress.ts มีหน้าที่จัดการ service และ state ที่เกี่ยวข้อง address ของ wallet
2. useAuth.ts มีหน้าที่จัดการ service และ state ที่เกี่ยวข้อง การ Login, Logout
3. useBackend.ts มีหน้าที่จัดการ service และ state ที่เกี่ยวข้องกับการติดต่อกับส่วนของ backend ผ่านการใช้ axios ซึ่งเป็น library ที่ถูก Import มาใช้งานในการใช้ HTTP request
4. useCollection.ts มีหน้าที่จัดการ service และ state ที่เกี่ยวข้องกับ Collection ผ่านการ fetch data จาก backend
5. useContracts.ts มีหน้าที่จัดการ service และ state ที่เกี่ยวข้องกับ Smart Contract โดยมีการเรียกใช้งาน library จาก Web3 และ Ethers เพื่อเข้าถึง Blockchain
6. useIPFS.ts มีหน้าที่จัดการ service ที่เกี่ยวข้องกับการใช้ IPFS

**1.4 Config**

ในส่วนของ Config นั้นจะเป็นส่วนของการตั้งค่าใน frontend เป็นในลักษณะ global เพื่อให้ง่ายต่อการแก้ไข variables ที่ถูกเรียกใช้บ่อยสำหรับการตั้งค่าต่าง ๆ ภายในระบบ โดยจะมี 3 ส่วนคือ

1. abi.json สำหรับ interface CONTRACT\_ADDRESS เพื่อให้ง่ายต่อการเรียกใช้ function ใน smart contract ส่วนของ NFT
2. abi2.json สำหรับ interface Market\_ADDRESS เพื่อให้ง่ายต่อการเรียกใช้ function ใน smart contract ส่วนของ Marketplace
3. index.js สำหรับการ exports variables ที่นำไปใช้ในการตั้งค่า application

**A picture containing text, screenshot, font

Description automatically generated**

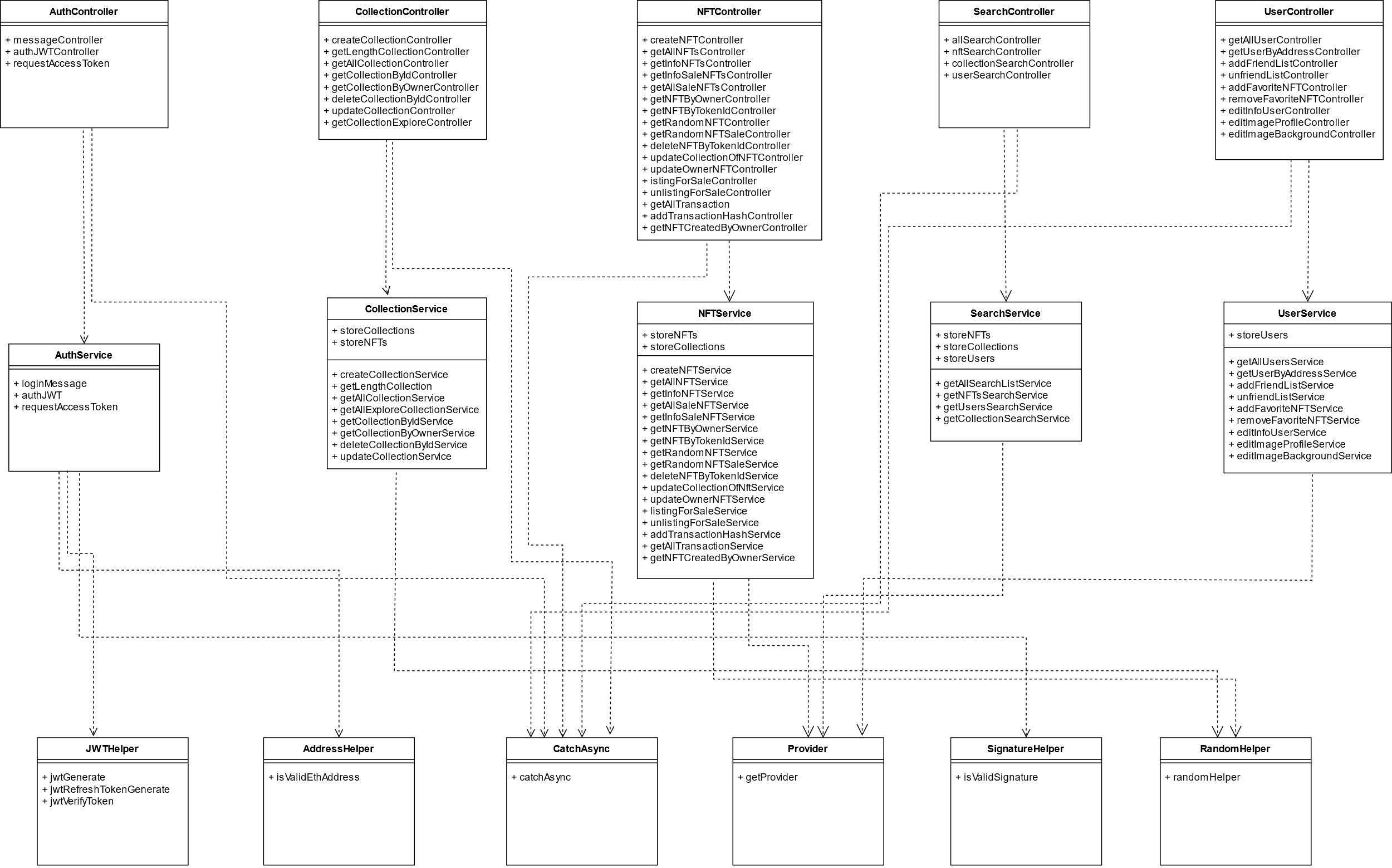
**รูปที่ 11 Config สำหรับ frontend**

**1.5 Package version**

ในส่วนของ version ที่ใช้ในการพัฒนาและ support นั้นสำหรับ dependencies สามารถดูได้จาก package.json โดยจะมี package หลักที่สำคัญประกอบด้วย package ดังต่อไปนี้

1. react/react-dom version 18.1.0
2. npm version 5.1.0
3. axios version 0.27.2
4. ethers version 5.7.0
5. web3 version 1.7.5
6. bootstrap version 5.2.0
7. @reduxjs/toolkit 1.8.3
8. **Backend**

backend โดยระบบของ NFT-Marketplace ใช้ node Express และใช้ Firebase เป็น database โดยในโครงสร้างของระบบ backend ที่ใช้ express นั้นจะมีโครงสร้างดังนี้



**รูปที่ 12 โครงสร้าง backend**

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 13 folder function**

ซึ่งจะเห็นได้ว่ามี folder functions, ไฟล์ .firebaserc และไฟล์ firebase.json โดยระบบเราทำการ deploy ไว้ในระบบของ firebase functions ดังนั้นจึงต้องจัดการ structure ของระบบให้เป็นในรูปแบบนี้

เมื่อทำการเปิด folder functions ดังภาพนี้

A screenshot of a computer

Description automatically generated with medium confidence

**รูปที่ 14 โครงสร้าง functions**

ก็จะเห็นว่าใน folder ของ functions นั้นจะมี folder src ที่จะเป็น code สำหรับการทำงานทั้งหมดของระบบ ไฟล์ .env ที่จะมีตัวแปร PRIVATE\_KEY ที่ใช้สำหรับการอ่านค่าที่อยู่ในระบบของ blockchain และไฟล์ index.js

A screenshot of a computer program

Description automatically generated with medium confidence

**รูปที่ 15 index.js**

ที่จะมีการกำหนด routes สำหรับการเรียกใช้งาน path แต่ละ path ที่ต้องการข้อมูลแตกต่างกันซึ่งจากภาพในบรรทัดที่ 20 มีการกำหนด path /v1/ และตามด้วย routes ซึ่งตัวของ routes เรียกใช้ routes จากไฟล์ path ./src/routes เมื่อเราเปิดเข้าไปใน folder src เพื่อจะไปที่ folder routes ที่อยู่ใน src จะเห็นโครงสร้างข้างใน src ดังนี้

A black screen with white text

Description automatically generated with low confidence

**รูปที่ 16 โครงสร้าง src**

ซึ่งใน folder src ก็จะการแบ่งโครงสร้างดังนี้

1. config
2. controllers
3. middlewares
4. routes
5. services
6. utils
7. validations

โดย folder ที่ถูกเรียกใช้เป็นอันดับแรกในไฟล์ index คือ routes เมื่อทำการเปิด folder routes จะเห็นไฟล์ทั้งหมดใน folder routes ดังนี้

A screenshot of a computer

Description automatically generated with medium confidence

**รูปที่ 17 โครงสร้าง routes**

จากภาพจะเห็นไฟล์ทั้งหมดใน routes ซึ่งการทำหลักจะอยู่ในไฟล์ index โดยหากเปิดไฟล์ index ขึ้นมาแล้วจะเห็น code ดังภาพ



**รูปที่ 18 การเพิ่ม path**

ตามภาพจะเห็นว่ามีการ เพิ่ม path สำหรับการ nft , auth, collection, user, search โดย path ที่สร้างขึ้นนี้ก็จะไปเรียกใช้ข้อมูลของ route ที่สร้างขึ้นมา nftRoute, authRoute, collectionRoute, userRoute, searchRoute ซึ่งหากผู้พัฒนาต้องการที่ติดต่อกับข้อมูลส่วนไหน ผู้พัฒนาก็ใส่ path ที่ต้องการใน API โดยแต่ route จะมีดังนี้

nftRoute แบ่งเป็น 4 ส่วน คือ Get, Post, Patch, Delete

1. API สำหรับการ Get ข้อมูล

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 19 โครงสร้าง router.get**

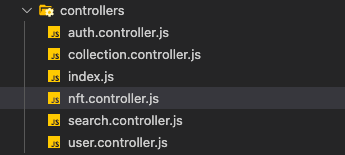
โดยจะผู้พัฒนาจะเห็น router.get ซึ่งจะ get ข้อมูลตาม path ที่ทางผู้พัฒนาเลือกซึ่งแบ่งออกเป็น path ได้ดังนี้

1. Path /getAllTransaction โดย API นี้จะทำการเรียกใช้ module getAllTransaction ใน controller ของ NFT
2. Path / โดย API นี้จะทำการเรียกใช้ module getAllNFTsController ใน controller ของ NFT
3. Path /sale โดย API นี้จะทำการเรียกใช้ module getAllSaleNFTsController ใน controller ของ NFT
4. Path /random โดย API นี้จะทำการเรียกใช้ module getRandomNFTController ใน controller ของ NFT
5. Path /randomNFTSale โดย API จะทำการเรียกใช้ module getRandomNFTSaleController ใน controller ของ NFT
6. Path /info โดย API นี้จะทำการเรียกใช้ module getInfoNFTsController ใน controller ของ NFT
7. Path /infoSale โดย API นี้จะทำการเรียกใช้ module getInfoSaleNFTsController ใน controller ของ NFT
8. Path /getNFTByOwner โดย API นี้จะทำการเรียกใช้ module getNFTByOwnerController ใน controller ของ NFT และจะต้องมีการ validate ข้อมูลโดยการเรียกใช้ validate ของ NFT และใช้ module getNFTByOwnerValidate
9. Path /getNFTCreatedByOwner โดย API นี้จะทำการเรียกใช้

module getNFTCreatedByOwnerController ใน controller ของ NFT และจะต้องมีการ validate ข้อมูลโดยการเรียกใช้ validate ข้อมูลของ NFT และใช้ module getNFTCreatedByOwnerValidate

1. Path /getNFTByTokenId โดย API นี้ทำการเรียกใช้ module getNFTByTokenIdController ใน controller ของ NFT และจะต้องมีการ validate ข้อมูลโดยการเรียกใช้ validate ของ NFT และใช้ module getNFTByTokenIdValidate

โดย controller ของ NFT ที่เรียกใช้นั้นจะอยู่ใน folder ของ controllers ดังภาพ



**รูปที่ 20 controllers**

และจะมี module สำหรับการ get ข้อมูลทั้งหมดดังภาพ

A screen shot of a computer program

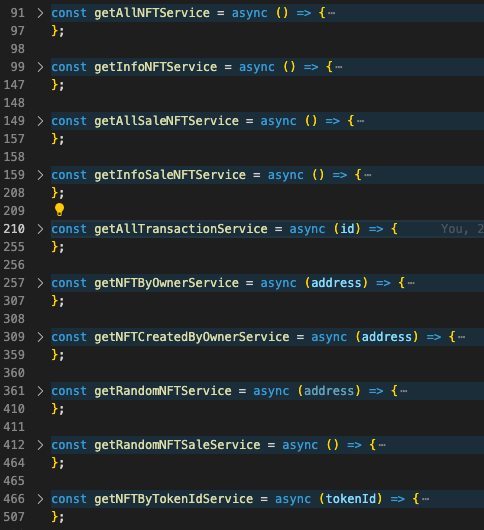
Description automatically generated with low confidence

**รูปที่ 21 module สำหรับ get ข้อมูล**

โดยแต่ละ module จะมีการทำงานดังนี้

1. getAllNFTsController โดย controller ตัวนี้จะมีการเรียกใช้งาน module getAllNFTService ที่เป็น service ของ NFT และทำการส่ง response กลับไป
2. getInfoNFTsController โดย controller ตัวนี้จะมีการเรียกใช้งาน module getInfoService ที่เป็น service ของ NFT และทำการส่ง response กลับไป
3. getInfoSaleNFTsController โดย controller ตัวนี้จะมีการเรียกใช้งาน module getInfoSaleNFTService ที่เป็น service ของ NFT และทำการส่ง response กลับไป
4. getAllSaleNFTsController โดย controller ตัวนี้จะมีการเรียกใช้งาน module getAllSaleNFTService ที่เป็น service ของ NFT และทำการส่ง response กลับไป
5. getRandomNFTController โดย controller ตัวนี้จะมีการเรียกใช้งาน module getRandomNFTService ที่เป็น service ของ NFT และทำการส่ง response กลับไป
6. getRandomNFTSaleController โดย controller ตัวนี้จะมีการเรียกใช้งาน module getRandomNFTSaleService ที่เป็น service ของ NFT และทำการส่ง response กลับไป
7. getNFTByOwnerController โดย controller ตัวนี้จะมีการสร้างตัวแปร address สำหรับเก็บข้อมูลที่อยู่ใน query ของ request และนำไปใส่ใน module getNFTByOwnerService ที่เป็น service ของ NFT และทำการส่ง response กลับไป
8. getNFTCreatedByOwnerController โดย controller ตัวนี้จะมีการสร้างตัวแปร address สำหรับเก็บข้อมูลที่อยู่ใน query ของ request และนำไปใส่ใน module getNFTCreatedByOwnerService ที่เป็น service ของ NFT และทำการส่ง response กลับไป
9. getNFTByTokenIdController โดย controller ตัวนี้จะมีการสร้างตัวแปร tokenId สำหรับเก็บข้อมูลที่อยู่ใน query ของ request และนำไปใส่ใน module getNFTByTokenIdService ที่เป็น service ของ NFT และทำการส่ง response กลับไป
10. getAllTransaction โดย controller ตัวนี้จะมีการสร้างตัวแปร id สำหรับเก็บข้อมูลที่อยู่ใน query ของ request และนำไปใส่ใน module getAllTransactionService ที่เป็น service ของ NFT และทำการส่ง response กลับไป

โดย controller เหล่านี้มีการเรียกใช้งาน service ที่มีการทำงานค้นหาข้อมูลใน database เพื่อที่จะทำการ return ข้อมูลกลับไปให้ controller ที่เรียกใช้ module ใน service เพื่อส่ง response กลับไปซึ่ง service ที่เป็น service สำหรับการ get ข้อมูลทั้งหมดดังนี้



**รูปที่ 22 service สำหรับการ get**

1. getAllNFTService เป็นการค้นหาข้อมูล NFT ทั้งหมดใน database และทำการ return ข้อมูลของ NFT ทั้งหมดไปให้ controller ที่เรียกใช้
2. getInfoNFTService เป็นการค้นหาข้อมูล NFT ทั้งหมดใน database แต่จะมีการคัดกรองข้อมูลที่สำคัญเช่น tokenId, nameNFT, tokenURI, collectionName, category, statusSale, price และทำการ return กลับไปให้ controller ที่เรียกใช้
3. getAllSaleNFTService เป็นการค้นหาข้อมูล NFT ทั้งหมดที่ทำการลงขายอยู่ในระบบและทำการ return กลับไปให้ controller ที่เรียกใช้
4. getInfoSaleNFTService เป็นการค้นหาข้อมูลที่ NFT ทั้งที่ทำการลงขายอยู่ในระบบพร้อมกับข้อมูลที่มีการคัดกรองข้อมูลที่สำคัญเช่น tokenId, nameNFT, tokenURI, collectionName, category, statusSale, price และทำการ return กลับไปให้ controller ที่เรียกใช้
5. getAllTransactionService เป็นการค้นหาข้อมูล transaction ของ NFT ที่ต้องการหาข้อมูลโดยจะมีการรับ id ของ token และจะทำการอ่านข้อมูล transactionHash และทำการอ่าน event และ return list ข้อมูล event, eventData, date, time กลับไปให้กลับ controller ที่เรียกใช้
6. getNFTByOwnerService เป็นการค้นหาข้อมูล NFT ทั้งหมดของ address ที่รับเข้ามาและทำการ return list ที่มีข้อมูล tokenId, nameNFT, tokenURI, collectionName, category, statusSale กลับไปให้ controller ที่เรียกใช้
7. getNFTCreatedByOwnerService เป็นการค้นหาข้อมูล NFT ทั้งหมดที่สร้างโดย address ที่รับเข้ามาและทำการ return list ที่มีข้อมูล tokenId, nameNFT, tokenURI, collectionName, category, statusSale กลับไปให้ controller ที่เรียกใช้
8. getRandomNFTService เป็นการค้นหาข้อมูล NFT แบบสุ่มและทำการ return NFT จำนวน 4 token ที่มีข้อมูลของ tokenId, nameNFT, tokenURI, collectionName, category, statusSale กลับไปให้ controller ที่เรียกใช้
9. getRandomNFTSaleService เป็นการค้นหาข้อมูล NFT ที่ทำการลงขายแบบสุ่มและทำการ return NFT จำนวน 4 token ที่มีข้อมูลของ tokenId, nameNFT, tokenURI, collectionName, category, statusSale กลับไปให้ controller ที่เรียกใช้
10. getNFTByTokenIdService เป็นการค้นหาข้อมูลของ NFT ที่ค้นหาจาก TokenId ที่รับเข้ามาและทำการ return ข้อมูลทั้งหมดจาก database กับไปและมีการเพิ่ม id ,tokenURI และ price ของ token
11. API สำหรับการ POST

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 23 router.post**

โดยจะผู้พัฒนาจะเห็น router.post ซึ่งจะ post ข้อมูลตาม path ที่ทางผู้พัฒนาเลือกซึ่งแบ่งออกเป็น path ได้ดังนี้

1. Path / โดย API นี้จะมีการเรียกใช้ module createNFTController ใน controller ของ NFT และ มีการเรียกใช้ middleware เพื่อทำการ validate jwt สำหรับการ post ข้อมูล และก็เรียกใช้ module validate body โดยใช้ createNFTValidate
2. Path /addTransactionHash โดย API นี้จะมีการเรียกใช้ module addTransactionHashController ใน controller ของ NFT และมีการเรียกใช้ middleware เพื่อทำการ validate jwt สำหรับการ post ข้อมูล

โดยใน Route นี้จะมีการเรียกใช้งาน controller ของ NFT สำหรับการ post ดังนี้

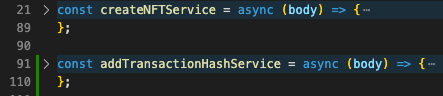
A screen shot of a computer screen

Description automatically generated with low confidence

**รูปที่ 24 createNFTController**

1. createNFTController เป็นการเรียกใช้ module createNFTService ที่เป็น service ของ NFT และมี argument ใน module คือ req.body และจะได้ข้อมูลกลับมาเพื่อส่ง response กลับไป
2. addTransactionHashController เป็นการเรียกใช้ module addTransactionHashService ที่เป็น service ของ NFT และมี argument ใน module คือ req.body และจะได้ข้อมูลกลับมาเพื่อส่ง response กลับไป

โดย controller เหล่านี้มีการเรียกใช้งาน service ที่มีการเขียนข้อมูลลงไปใน database โดยจะมี module สำหรับ post ทั้งหมดดังนี้



**รูปที่ 25 module สำหรับ post**

1. createNFTService เป็นการสร้าง NFT โดยที่จะรับ body เข้ามาและทำการสร้างตัวแปร tokenId,ownerAddress, nameNFT, description, category, collectionId, transactionHash และทำการเขียนข้อมูลลงไปใน ใน database ดังนี้

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 26 การเขียนข้อมูลลงใน database**

1. addTransactionHashService เป็นการเพิ่มข้อมูลของ transactionHash เข้าไปใน list ของ transationHash เข้าไปใน token ที่รับเข้ามา
2. API สำหรับ Patch



**รูปที่ 27 การ patch**

โดยจะผู้พัฒนาจะเห็น router.patch ซึ่งจะ update ข้อมูลตาม path ที่ทางผู้พัฒนาเลือกซึ่งแบ่งออกเป็น path ได้ดังนี้

1. Path / จะมีการเรียกใช้ module updateCollectionOfNFTController ใน controller ของ NFT และมีการเรียกใช้ middleware เพื่อทำการ validate jwt สำหรับการ update ข้อมูล และมีการเรียกใช้ module validate ของ NFT ที่ validate body
2. Path /updateOwner จะมีการเรียกใช้ module updateOwnerNFTController ใน controller ของ NFT และมีการเรียกใช้ middleware เพื่อทำการ validate jwt สำหรับการ update ข้อมูล และมีการเรียกใช้ module validate ของ NFT ที่ validate body
3. Path /listingForSale จะมีการเรียกใช้ module listingForSaleController ใน controller ของ NFT และมีการเรียกใช้ middleware เพื่อทำการ validate jwt สำหรับการ update ข้อมูล และมีการเรียกใช้ module validate ของ NFT ที่ validate body และ validate query
4. Path /unlistingForSale จะมีการเรียกใช้ module unlistingForSaleController ใน controller ของ NFT และมีการเรียกใช้ middleware เพื่อทำการ validate jwt สำหรับการ update ข้อมูล และมีการเรียกใช้ module validate ของ NFT ที่ validate body และ validate query

โดย routes สำหรับการ update เหล่านี้เรียกใช้ controller ดังภาพ

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 28 routes สำหรับการ update**

1. listingForSaleController รับ request body มาเป็น id และรับ request query มาเป็น ownerAddress และทำการเรียกใช้ module listingForSaleService ซึ่งเป็น service ของ NFT และมี argument ต้องใส่คือ id และ ownerAddress และทำการส่ง response กลับไป
2. unlistingForSaleController รับ request body มาเป็น id และรับ request query มาเป็น ownerAddress และทำการเรียกใช้ module unlistingForSaleService ซึ่งเป็น service ของ NFT และมี argument ต้องใส่คือ id และ ownerAddress และทำการส่ง response กลับไป
3. updateCollectionOfNFTController โดย controller นี้มีการเรียกใช้ module updateCollectionOfNftService และมี argument เป็น request body ที่ใส่เข้าไป และทำการส่ง response กลับไป
4. updateOwnerNFTController โดย controller นี้มีการเรียกใช้ module updateOwnerNFTService และมี argument เป็น request body ที่ใส่เข้าไป และทำการส่ง response กลับไป

โดย controller เหล่านี้มีการเรียกใช้ service ที่ใช้สำหรับการอัพเดททั้งหมดดังนี้

A picture containing text, screenshot, font, software

Description automatically generated

**รูปที่ 29 routes สำหรับการ update**

1. listingForSaleService การทำงานของ service นี้จะทำการรับ id ของ token และ ownerAddress เข้ามาเพื่อที่จะทำการเปลี่ยน status การลงขายของ NFT และ return string กลับไปยืนยันการ update
2. unlistingForSaleService การทำงานของ service นี้จะทำการรับ id ของ token และ ownerAddress เข้ามาเพื่อที่จะทำยกเลิกการลงขายของ NFT และ return string กลับไปยืนยันการ update
3. updateCollectionOfNftService การทำงานของ service นี้จะทำการรับ id, collectionId, ownerAddress โดยจะมีการอัพเดท collection จาก collectionId ที่รับเข้ามา และ return string กลับไปยืนยันการ update
4. updateOwnerNFTService การทำงานของ service นี้จะทำการรับ id จาก body โดยจะมีการอัพเดทเจ้าของ NFT ที่รับเข้ามา และ return string กลับไปยืนยันการ update
5. API สำหรับการ Delete

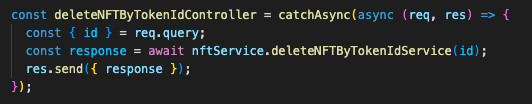
A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 30 routes สำหรับการ delete**

โดยจะผู้พัฒนาจะเห็น router.delete ซึ่งจะ delete ข้อมูลตาม path ที่ทางผู้พัฒนาเลือกซึ่งแบ่งออกเป็น path ได้ดังนี้

1. Path / จะมีการเรียกใช้ module deleteNFTByTokenIdController ใน controller ของ NFT และมีการเรียกใช้ middleware เพื่อทำการ validate jwt สำหรับการ delete ข้อมูล และมีการเรียกใช้ module validate ของ NFT ที่ validate query โดย route นี้มีการเรียกใช้ controller สำหรับการ delete ข้อมูลดังนี้



**รูปที่ 31 controller สำหรับการ delete NFTByTokenId**

1. deleteNFTByTokenIdController โดยจะมีการรับ id ที่อยู่ใน request body และทำการเรียกใช้ module delteNFTTokenService และใส่ argument เป็น id เข้าไปและจะได้ response ส่งกลับไปซึ่ง controller นี้เรียกใช้ service สำหรับการ delete ข้อมูลที่อยู่ใน database ดังนี้

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 32 controller สำหรับการ กำสำ NFTByTokenId**

1. deleteNFTByTokenIdService โดย Service นี้จะทำการลบข้อมูลใน database จาก tokenId ที่รับเข้ามาและทำการ return string เพื่อยืนยันการลบข้อมูลโดยจาก route ของ NFT มีการเรียกใช้งาน validation ที่เรียกใช้ nftValidation ในบรรทัดที่ 4 ดังภาพ



**รูปที่ 33 nftValidation**

จากบรรทัดที่ผู้พัฒนาจะเห็นได้ว่ามีการเรียกจาก folder validations ที่ไฟล์ nft.validation.js ดังภาพ

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 34 folder validations**

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 35 โครงสร้าง validation**

เมื่อผู้พัฒนาเข้าไปในไฟล์ nft.validation.js จะเห็นการสร้างตัว validate ข้อมูลดังนี้

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 36 การสร้าง validate**

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 37 listingForSaleValidate**

โดยผู้พัฒนาจะสังเกตเห็นว่าตัวแปร Object ทั้งหมดโดยจะมีการแบ่งดังนี้

1. createNFTValidate ที่มีการสร้างว่า body จะต้องเป็น object ที่มี
2. ownerAddress เป็น string
3. nameNFT เป็น string
4. description เป็น string
5. category เป็น array
6. collectionId เป็น string
7. tokenId เป็น number
8. transactionHash เป็น sring
9. getNFTByOwnerValidate ที่มีการสร้าง query ที่เป็น object ที่มี address เป็น string
10. getNFTCreatedByOwnerValidate ที่มีการสร้าง query ที่เป็น object ที่มี address เป็น string
11. getNFTByTokenIdValidate ที่มีการสร้าง query ที่เป็น object ที่มี tokenId เป็น number
12. listingForSaleValidate ที่มีการสร้าง query ที่เป็น object ที่มี ownerAddress เป็น string

และมีการสร้าง body ที่เป็น object ที่มี id เป็น string

1. unlistingForSaleValidate ที่มีการสร้าง query ที่เป็น object ที่มี ownerAddress เป็น string

และมีการสร้าง body ที่เป็น object ที่มี id เป็น string

1. updateCollectionOfNftValidate ที่มีการสร้าง body ที่เป็น object ที่มี
2. id เป็น string
3. collectionId เป็น string
4. updateOwnerNFTValidate ที่มีการสร้าง body ที่เป็น object ที่มี
5. id เป็น string
6. contract เป็น string
7. deleteNFTByTokenIdValidate ที่มีการสร้าง query ที่เป็น object ที่มี id เป็น string

authRoute จะมีเฉพาะ router get ดังภาพ



**รุปที่ 38 authRoute**

API สำหรับการ Get มีดังนี้

1. Path /message จะมีการเรียกใช้ module messageController ใน controller ของ auth และมีการเรียกใช้ module loginMessageValidate ซึ่งเป็น validate ของ auth
2. Path /jwt จะมีการเรียกใช้ module authJWTController ใน controller ของ auth และมีการเรียกใช้ module authJWTValidate ซึ่งเป็น validate ของ auth
3. Path /requestAccessToken จะมีการเรียกใช้ module requestAccessToken ใน controller ของ auth และมีการเรียกใช้ module requestAccessTokenValidate ซึ่งเป็น validate ของ auth

โดย route นี้มีการเรียกใช้ controller สำหรับการ get ข้อมูลดังนี้

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 39 messageController**

1. messageController โดยจะมีการรับ address ที่อยู่ใน request query และทำการเรียกใช้ module loginMessage ซึ่งเป็น service ของ auth และใส่ argument เป็น address และส่ง response กลับไป
2. authJWTController โดยจะมีการรับ address , signature ที่อยู่ใน request query และทำการเรียกใช้ module authJWT ซึ่งเป็น service ของ auth และใส่ argument เป็น address , signature และส่ง response กลับไป
3. requestAccessToken โดยจะมีการรับ address ที่อยู่ใน request query และทำการเรียกใช้ module requestAccessToken ซึ่งเป็น service ของ auth และใส่ argument เป็น และส่ง response กลับไป

โดย controller เหล่านี้มีการเรียกใช้ service ที่ใช้สำหรับการ get ทั้งหมดดังนี้

A picture containing text, screenshot, font, number

Description automatically generated

**รูปที่ 40 service ที่ใช้ในการ get**

1. loginMessage จะมีการทำงานคือการ getMessage ที่จะ sign กับ metamask หากมีข้อมูล user อยู่แล้วแต่หากไม่มีก็จะทำการเพิ่มข้อมูลลงไปใน database ของ user

A screen shot of a computer

Description automatically generated with medium confidence

**รูปที่ 41 การเพิ่ม database ของ user**

1. authJWT จะมีการรับ address เข้ามาและทำการ generate jwt โดย address และจะได้ access token และ refresh token



**รูปที่ 42 การ generate jwt**

และทำการ return token ทั้งสองกลับไป

1. requestAccessToken จะมีการรับ address เข้ามาและทำการ generate access token อันใหม่และทำการ return ข้อมูลกลับไปโดย route ของ auth นั้นจะมีการเรียกใช้ authvalidation ดังรูป



**รูปที่ 43 authValidation**

ซึ่งใน authValidation นั้นจะมีการ validate ดังนี้

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 44 validate ใน authValidation**

* 1. loginMessageValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี address เป็น string
  2. authJWTValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี

1. address เป็น string
2. Signature เป็น string
   1. requestAccessTokenValidate ที่เป็นการสร้าง query จะต้องเป็น object ที่มี address เป็น string

collectionRoute จะแบ่งออกเป็น Get, Post , Delete, Patch ดังภาพ

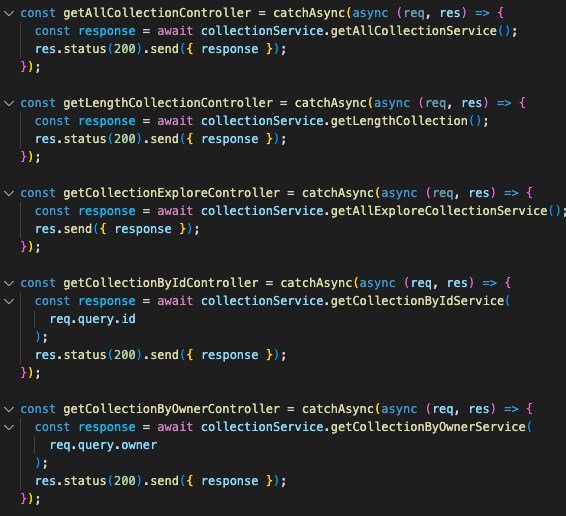


**รูปที่ 45 collectionRoute**

API สำหรับการ Get มีดังนี้

1. Path /length จะมีการเรียกใช้ module getLengthCollectionController ใน controller ของ collection
2. Path / จะมีการเรียกใช้ module getAllCollectionController ใน controller ของ collection และมีการเรียกใช้ module getAllCollectionValidate ซึ่งเป็น validate ของ collection
3. Path /random จะมีการเรียกใช้ module getCollectionExploreController ใน controller ของ collection
4. Path /getCollectionById จะมีการเรียกใช้ module getCollectionByIdController ใน controller ของ collection และมีการเรียกใช้ module getCollectionByIdValidate ซึ่งเป็น validate ของ collection
5. Path /getCollectionByOwner จะมีการเรียกใช้ module getCollectionByOwnerController ใน controller ของ collection และมีการเรียกใช้ module getCollectionByOwnerValidate ซึ่งเป็น validate ของ collection

โดยใน route ของ collection นั้นจะมีการเรียกใช้ controller สำหรับการ get ข้อมูลดังนี้



**รูปที่ 46 controller สำหรับการ get**

1. getAllCollectionController โดยจะมีการเรียกใช้ module getAllCollectionService ซึ่งเป็น service ของ collection และทำการส่ง response กลับไป
2. getLengthCollectionController โดยจะมีการเรียกใช้ module getLengthCollection ซึ่งเป็น service ของ collection และทำการส่ง response กลับไป
3. getCollectionExploreController โดยจะมีการเรียกใช้ module getAllExploreCollectionService ซึ่งเป็น service ของ collection และทำการส่ง response กลับไป
4. getCollectionByIdController โดยจะมีการรับตัวแปร id จาก request query และมีการเรียกใช้ module getCollectionByIdService และทำการใส่ใน argument คือ id โดยmodule นี้เป็น service ของ collection และทำการส่ง response กลับไป
5. getCollectionByOwnerController โดยจะมีการรับตัวแปร owner จาก request query และมีการเรียกใช้ module getCollectionByOwnerService และทำการใส่ใน argument คือ owner โดย module นี้เป็น service ของ collection และทำการส่ง response กลับไปซึ่ง controller ก็ได้มีการเรียกใช้ข้อมูลใน service ของ collection ดังนี้

A screenshot of a computer program

Description automatically generated with low confidence

**รูปที่ 47 service ของ collection**

1. getLengthCollection โดย Service นี้จะมีการ get ข้อมูลของ collection โดยจะส่งกลับไปเป็น length ความยาวของ collection
2. getAllCollectionService โดย service นี้จะมีการ get ข้อมูลทั้งหมดของ collection ทุก collection และมีเพิ่ม nft image เข้าไปด้วย
3. getAllExploreCollectionService โดย service นี้จะมีการ get ข้อมูลแบบสุ่มเพื่อส่งข้อมูลของ collection จำนวน 4 collection พร้อมกับ nft image กลับไปด้วย
4. getCollectionByIdService จะมีการรับ id เข้ามาเพื่อใช้ในการค้นหา collection ที่ตรงกับ id และทำการ return ข้อมูลของ collection นั้นกลับไป
5. getCollectionByOwnerService จะมีการรับ address ของ owner เข้ามาและหา collection ทั้งหมดของ owner เจ้าของ address และทำการ return list ของ collection กลับไป

API สำหรับการ Post มีดังนี้

1. Path / จะมีการเรียกใช้ module createCollectionController ใน controller ของ collection และมีการเรียกใช้ module createCollectionValidate ซึ่งเป็น validate ของ collection และมีการ validate jwt สำหรับการเขียนข้อมูลลง database โดย path นี้มีการเรียกใช้งาน controller สำหรับการ post ข้อมูลดังนี้

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 48 createCollectionController**

1. createCollectionController โดยจะมีการรับตัวแปรจาก request body และมีการเรียกใช้ module createCollectionService และทำการใส่ใน argument คือ request body โดย module นี้เป็น service ของ collection และทำการส่ง response กลับไปซึ่ง controller ก็ได้มีการเรียกใช้ข้อมูลใน service ของ collection ดังนี้

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 49 createCollectionService**

1. createCollectionService โดยจะมีการรับ body ที่มี owner, collectionName, description เข้ามาและทำการสร้าง collection และตามด้วยการเพิ่มข้อมูลของ collectionId เข้าไป และ return ข้อมูลที่สร้างขึ้นมากลับไป

API สำหรับการ Delete มีดังนี้

1. Path / จะมีการเรียกใช้ module deleteCollectionByIdController ใน controller ของ collection และมีการเรียกใช้ module deleteCollectionByIdValidate ซึ่งเป็น validate ของ collection และมีการ validate jwt สำหรับการลบข้อมูลออกจาก database โดย path นี้มีการเรียกใช้งาน controller สำหรับการ Delete ข้อมูลดังนี้

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 50 deleteCollectionByIdController**

1. deleteCollectionByIdController โดยจะมีการรับตัวแปร id จาก request query และมีการเรียกใช้ module deleteCollectionByIdService และทำการใส่ใน argument คือ id โดย module นี้เป็น service ของ collection และทำการส่ง response กลับไปซึ่ง controller ก็ได้มีการเรียกใช้ข้อมูลใน service ของ collection ดังนี้

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 51 deleteCollectionByIdService**

1. deleteCollectionByIdService โดยจะมีการรับ id เข้ามาเพื่อทำการค้นหา collection ที่ต้องการจะลบและทำการลบข้อมูลจาก database และทำการ return string กลับไปบอกผลการลบข้อมู,

API สำหรับการ Patch มีดังนี้

1. Path / จะมีการเรียกใช้ module updateCollectionController ใน controller ของ collection และมีการเรียกใช้ module updateCollectionValidate ซึ่งเป็น validate ของ collection และมีการ validate jwt สำหรับการอัพเดทข้อมูลใน database โดย path นี้มีการเรียกใช้งาน controller สำหรับการ Patch ข้อมูลดังนี้

A picture containing screenshot, text, font

Description automatically generated

**รูปที่ 52 updateCollectionController**

1. updateCollectionController โดยจะมีการรับตัวแปรจาก request body และมีการเรียกใช้ module updateCollectionService และทำการใส่ใน argument คือ request body โดย module นี้เป็น service ของ collection และทำการส่ง response กลับไปซึ่ง controller ก็ได้มีการเรียกใช้ข้อมูลใน service ของ collection ดังนี้

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 53 updateCollectionService**

1. updateCollectionService โดยจะมีการรับ id จาก body เพื่อใช้ค้นหา collection ที่ต้องการจะ update ข้อมูล หากเจอจะทำการนำ collectionName, description จาก body เพื่อใช้อัพเดทข้อมูลใน collectionValidation นั้นจะมีการ validate ดังนี้

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 54 createCollectionValidate**

1. createCollectionValidate ที่มีการสร้าง body จะต้องเป็น object ที่มี
2. owner เป็น string
3. collectionName เป็น string
4. description เป็น string
5. getAllCollectionValidate ที่มีการสร้าง query จะต้องเป็น object
6. getCollectionByIdValidate ที่เป็นการสร้าง query จะต้องเป็น object ที่มีowner เป็น string
7. deleteCollectionByIdValidate ที่เป็นการสร้าง body จะต้องเป็น object ที่มี
8. id เป็น string
9. collectionName เป็น string
10. description เป็น string

userRoute จะแบ่งออกเป็น get, post ดังนี้



**รูปที่ 55 userRoute**

API สำหรับการ get ข้อมูลดังภาพ

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 56 api สำหรับการ get ข้อมูล**

1. Path / โดย Service นี้จะมีการ get ข้อมูลของ user โดยจะมีการเรียกใช้ module getAllUserController ของ user controller และมีการเรียกใช้ validate สำหรับการ validate getAllUsers
2. Path /getUserByAddress โดย Service นี้จะมีการ get ข้อมูลของ user โดยจะมีการเรียกใช้ module getUserByAddressController ของ user controller และมีการเรียกใช้ validate สำหรับการ validate getUserByAddress

โดย path นี้มีการเรียกใช้งาน controller สำหรับการ get ข้อมูลดังนี้

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 57 getAllUserController**

1. getAllUserController จะมีการเรียกใช้ module getAllUsersService ซึ่งเป็น service ของ user และจะทำการส่ง response กลับไป
2. getUserByAddressController โดยจะมีการรับ address จาก request query และมีการเรียกใช้ module getUserByAddressService และทำการนำ address ใส่ใน argument โดย module นี้เป็น service ของ user และทำการส่ง response กลับไป ซึ่ง controller ก็ได้มีการเรียกใช้ข้อมูลใน service ของ user ดังนี้

A picture containing text, screenshot, software, font

Description automatically generated

**รูปที่ 58 getAllUserService**

1. getAllUsersService จะมีการทำงานคือการ get ข้อมูลของ user ทั้งหมด และทำการ return ข้อมูลทั้งหมดกลับไป
2. getUserByAddressService จะมีการทำงานคือทำการรับ address เข้ามาเพื่อทำการ filter user ที่ตรงกับ address และทำการ return ข้อมูลกลับไป

API สำหรับการ post ข้อมูลดังภาพ

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 59 path/unfriendList**

1. Path /unfriendList จะมีการเรียกใช้ module unfriendListController ใน controller ของ user และมีการเรียกใช้ module unfriendList ซึ่งเป็น validate ของ user และมีการ validate jwt สำหรับการเขียนข้อมูลลง database
2. Path /addFavoriteNFT จะมีการเรียกใช้ module addFavoriteNFTController ใน controller ของ user และมีการเรียกใช้ module addFavoriteNFT ซึ่งเป็น validate ของ user และมีการ validate jwt สำหรับการเขียนข้อมูลลง database
3. Path /removeFavoriteNFT จะมีการเรียกใช้ module removeFavoriteNFTController ใน controller ของ user และมีการเรียกใช้ module removeFavoriteNFT ซึ่งเป็น validate ของ user และมีการ validate jwt สำหรับการเขียนข้อมูลลง database
4. Path /editInfoUser จะมีการเรียกใช้ module editInfoUserController ใน controller ของ user และมีการ validate jwt สำหรับการเขียนข้อมูลลง database
5. Path /editImageProfile จะมีการเรียกใช้ module editImageProfileController ใน controller ของ user และมีการเรียกใช้ module editImageProfile ซึ่งเป็น validate ของ user และมีการ validate jwt สำหรับการเขียนข้อมูลลง database
6. Path /editImageBackground จะมีการเรียกใช้ module editImageBackgroundController ใน controller ของ user และมีการเรียกใช้ module editImageBackground ซึ่งเป็น validate ของ user และมีการ validate jwt สำหรับการเขียนข้อมูลลง databaseโดย path นี้มีการเรียกใช้งาน controller สำหรับการ post ข้อมูลดังนี้

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 60 addFreindListController**

1. addFriendListController โดยจะมีการรับตัวแปร address จาก request query และ frinedAddress จาก request body และมีการเรียกใช้ module addFrinendListService และทำการใส่ใน argument คือ address , friendAddress โดย module นี้เป็น service ของ user และทำการส่ง response กลับไป
2. unfriendListController โดยจะมีการรับตัวแปร address จาก request query และ frinedAddress จาก request body และมีการเรียกใช้ module unfriendListService และทำการใส่ใน argument คือ address , friendAddress โดย module นี้เป็น service ของ user และทำการส่ง response กลับไป
3. addFavoriteNFTController โดยจะมีการรับตัวแปร address จาก request query และมีการเรียกใช้ module addFavoriteNFTService และทำการใส่ใน argument คือ address , request body โดย module นี้เป็น service ของ user และทำการส่ง response กลับไป
4. removeFavoriteNFTController โดยจะมีการรับตัวแปร address จาก request query และ tokenId จาก request body และมีการเรียกใช้ module removeFavoriteNFTService และทำการใส่ใน argument คือ address , tokenId โดย module นี้เป็น service ของ user และทำการส่ง response กลับไป
5. editInfoUserController โดยจะมีการรับตัวแปร address จาก request query และมีการเรียกใช้ module editInfoUserService และทำการใส่ใน argument คือ address , request body โดย module นี้เป็น service ของ user และทำการส่ง response กลับไป
6. editImageProfileController โดยจะมีการรับตัวแปร address จาก request query และ profileImage จาก request body และมีการเรียกใช้ module editImageProfileService และทำการใส่ใน argument คือ address , profileImage โดย module นี้เป็น service ของ user และทำการส่ง response กลับไป
7. editImageBackgroundController โดยจะมีการรับตัวแปร address จาก request query และ backgroundImage จาก request body และมีการเรียกใช้ module editImageBackgroundService และทำการใส่ใน argument คือ address , backgroundImage โดย module นี้เป็น service ของ user และทำการส่ง response กลับไป

ใน userValidation นั้นจะมีการ validate ดังนี้

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 61 getUserByAddressValidate**

1. getUserByAddressValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี address เป็น string
2. addFriendListValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
3. address เป็น stringและสร้าง body ที่จะต้องเป็น object ที่มี
4. friendAddress เป็น string
5. unfriendListValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
6. address เป็น string และสร้าง body ที่จะต้องเป็น object ที่มี
7. friendAddress เป็น string
8. addFavoriteNFTValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
9. address เป็น string และสร้าง body ที่จะต้องเป็น object ที่มี
10. tokenId เป็น number
11. removeFavoriteNFTValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
12. address เป็น string และสร้าง body ที่จะต้องเป็น object ที่มี
13. tokenId เป็น number
14. editInfoUserValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
15. address เป็น string และสร้าง body ที่จะต้องเป็น object ที่มี
16. name เป็น string
17. bio เป็น string
18. twitter เป็น string
19. instagram เป็น string
20. contact เป็น string
21. editImageProfileValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
22. address เป็น string และสร้าง body ที่จะต้องเป็น object ที่มี
23. profileImage เป็น string
24. editImageBackgroundValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
25. address เป็น string และสร้าง body ที่จะต้องเป็น object ที่มี
26. backgroundImage เป็น string

searchRoute จะมีเฉพาะ router get ดังนี้

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 62 path โดย Service**

1. Path / โดย Service นี้จะมีการ get ข้อมูลทั้งหมดของการ search ทั้ง collection, user, NFT โดยจะมีการเรียกใช้ module allSearchController ของ search controller และมีการเรียกใช้ validate สำหรับการ validate getAllSearchValidate
2. Path /getNFTSearch โดย Service นี้จะมีการ get ข้อมูลของการ search โดยจะมีการเรียกใช้ module nftsSearchController ของ search controller และมีการเรียกใช้ validate สำหรับการ validate getNFTsSearchValidate
3. Path /getUserSearch โดย Service นี้จะมีการ get ข้อมูลของการ search โดยจะมีการเรียกใช้ module usersSearchController ของ search controller และมีการเรียกใช้ validate สำหรับการ validate getUsersSearchValidate
4. Path /getCollectionSearch โดย Service นี้จะมีการ get ข้อมูลของการ search โดยจะมีการเรียกใช้ module collectionSearchController ของ search controller และมีการเรียกใช้ validate สำหรับการ validate getCollectionSearchValidate

โดย path นี้มีการเรียกใช้งาน controller สำหรับการ get ข้อมูลดังนี้

A picture containing text, screenshot, software

Description automatically generated

**รูปที่ 63 allSearchController**

1. allSearchController โดยจะมีการรับ keyword จาก request query และจะมีการเรียกใช้ module getAllSearchListService และทำการนำ keyword ใส่ใน argument โดย module นี้เป็น service ของ search และทำการส่ง response กลับไป
2. nftsSearchController โดยจะมีการรับ keyword จาก request query และจะมีการเรียกใช้ module getNFTsSearchService และทำการนำ keyword ใส่ใน argument โดย module นี้เป็น service ของ search และทำการส่ง response กลับไป
3. usersSearchController โดยจะมีการรับ keyword จาก request query และจะมีการเรียกใช้ module getUsersSearchService และทำการนำ keyword ใส่ใน argument โดย module นี้เป็น service ของ search และทำการส่ง response กลับไป
4. collectionsSearchController โดยจะมีการรับ keyword จาก request query และจะมีการเรียกใช้ module getCollectionsSearchService และทำการนำ keyword ใส่ใน argument โดย module นี้เป็น service ของ search และทำการส่ง response กลับไป

ซึ่ง controller ก็ได้มีการเรียกใช้ข้อมูลใน service ของ user ดังนี้

A screenshot of a computer program

Description automatically generated with low confidence

**รูปที่ 64 controller service**

1. getAllSearchListService จะมีการทำงานคือการ get ข้อมูลของการ search ข้อมูลทั้ง NFT, user, collection และทำการ return ข้อมูลทั้งหมดกลับไป
2. getNFTsSearchService จะมีการทำงานคือการ get ข้อมูลของการ search ข้อมูลทั้ง NFT และทำการ return ข้อมูลทั้งหมดกลับไป
3. getUsersSearchService จะมีการทำงานคือการ get ข้อมูลของการ search ข้อมูลทั้ง user และทำการ return ข้อมูลทั้งหมดกลับไป
4. getCollectionsSearchService จะมีการทำงานคือการ get ข้อมูลของการ search ข้อมูลทั้ง collection และทำการ return ข้อมูลทั้งหมดกลับไป

ใน searchValidation นั้นจะมีการ validate ดังนี้

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 65 getAllSearchValidate**

1. getAllSearchValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
2. keyword เป็น string
3. getNFTsSearchValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
4. keyword เป็น string
5. getUsersSearchValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
6. keyword เป็น string
7. getCollectionsSearchValidate ที่มีการสร้าง query จะต้องเป็น object ที่มี
8. keyword เป็น string

ทั้งหมดเป็น route ทั้งหมดที่มีอยู่ในระบบของ backend แต่ต่อไปจะอธิบายถึงส่วนที่มีการเรียกใช้เหมือนกันในระบบคือ ส่วนของ config, middlewares,utils

1. config

A picture containing text, screenshot, font, number

Description automatically generated

**รูปที่ 66 getAllSearchValidate**

Folder config จะมีดังนี้ไฟล์ abi.json ซึ่งจะมี event ของ สำหรับการดักอ่าน log ของ transactionHash และจะมีไฟล์ config.js ที่จะมีการกำหนดค่าต่างๆ ที่ใช้ในทางระบบและมีไฟล์ firebase.js ที่จะต้อง config ไฟล์สำหรับการติดต่อกับระบบ firebase

1. middlewares

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 67 getAllSearchValidate**

Folder middlewares จะมีไฟล์ดังนี้ authJWT.js ที่จะมีการ validate token ที่แนบมาในส่วนของ header และไฟล์ validate จะมีการทำ validate ตัวของ params, query, body

1. utils

A screenshot of a computer

Description automatically generated with medium confidence

**รูปที่ 68 utils**

Folder utils จะมี

1. addressHelper ที่ใช้ในการ validate address ว่าเป็น address ของ crypto waller จริงหรือเปล่า



**รูปที่ 69 isValidEthAddress**

1. catchAsync จะเป็นการทำ catch error หากเกิด error ขึ้น

A picture containing text, screenshot, font, line

Description automatically generated

**รูปที่ 70 catchAsync**

1. jwtHelper จะแบ่งออกเป็นดังนี้ jwtGenerate, jwtRefreshTokenGenerate, jwtVerifyToken

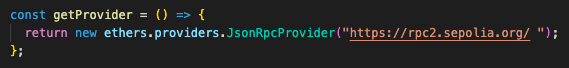
A picture containing text, screenshot, software

Description automatically generated

**รูปที่ 71 jwtGenerate**

โดยฟังก์ชัน jwtGenerate จะมีการรับ address และทำการ generate access token และทำการ return access token กลับไป ฟังก์ชัน jwtRefreshTokenGenerate จะมีการรับ address และทำการ generate access token และทำการ return access token กลับไป และสุดท้ายคือ jwtVerifyToken จะมีการ verify token และreturn ผลลัพธ์กลับไป

1. provider จะทำการ return provider กลับไป



**รูปที่ 72 getProvider**

1. randomHelper จะมีการรับ ความยาวของเลข และ จำนวนที่มากที่สุด และทำการ return list ของ number กลับมาแบบสุ่ม

A picture containing text, screenshot, display, software

Description automatically generated

**รูปที่ 73 randomNumber**

1. signatureHelper จะมีการรับ address, signature, messageToSign เพื่อ signature และ address

A picture containing text, screenshot, software

Description automatically generated

**รูปที่ 74 isValidSignature**

1. **Smart contract**

smart contract โดยพัฒนาโดยใช้ remix ในการพัฒนาโดยเขียนเป็นภาษา solidity และ deploy ไว้ในระบบ etheruem sepolia testnet

A screenshot of a computer program

Description automatically generated with low confidence

**รูปที่ 75 โครงสร้าง folder ของ smart contract**

ซึ่งจะเห็นได้ว่าจะมี folder ดังภาพโดยจะเริ่มจากการเข้าไปที่ folder scripts



**รูปที่ 76 folder scripts**

จากภาพจะเห็น file deploy\_ethers.js ซึ่งจะเป็นไฟล์สำหรับการ deploy smart contract ในระบบ blockchain

A picture containing text, screenshot

Description automatically generated

**รูปที่ 77 script สำหรับการ deploy**

ในส่วนนี้จะมีการ deploy smart contract จาก path ของ artifact ตามที่กำหนดไว้ในบรรทัดที่ 11 จากการกำหนดชื่อของ contractName ในบรรทัดที่ 6 และจะมีการเรียกตัวของ metadata ตาม path ของ artifacts ที่ได้กำหนดไว้ ในบรรทัดที่ 13 ในบรรทัดที่ 15 จะเป็นการ get ข้อมูลของผู้ที่ทำการ deploy smart contract และข้อมูลของ blockchain network จาก crypto wallet และทำการสร้าง contract factory และทำการ deploy smart contract

ส่วนต่อไปเป็นส่วนของ contract ที่มีการเรียกในการ deploy contract

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 78 folder ของ contracts**

จากการเรียกในการ deploy จะไปที่ folder artifacts ที่อยู่ใน folder contracts ซึ่งจะมีไฟล์ json ของ smart contract หลังจากการ complie smart contract โดยที่ไฟล์ทั้งหมดจะเป็นไฟล์ abi กับ bytecode ของ smart contract ดังภาพ

A screenshot of a computer

Description automatically generated with medium confidence

**รูปที่ 79 artifacts ของ contracts**

ซึ่งข้อมูลใน artifacts ทั้งหมดนี้มาจาก 2 ไฟล์ smart contract ดังนี้



**รูปที่ 80 file smart contract**

1. NFT

nft contract นี้จะมี constructor คือ name และ symbol ของ contract ดังนี้

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 81 constructor ของ NFT contract**

A picture containing text, screenshot

Description automatically generated

**รูปที่ 82 การกำหนดตัวแปรใน NFT contract**

จากภาพเป็นการกำหนดตัวแปรที่จะใช้เก็บข้อมูลลงใน blockchain และในไฟล์ของ nft contract นี้จะมีทั้งหมด 31 function, 1 event

A picture containing text, font, screenshot, design

Description automatically generated

**รูปที่ 83 event สำหรับการ mint NFT**

1 event คือ event ของการ mint โดยที่สร้างไว้สำหรับการดักจับ logs ของ event ของ transactionHash เพื่อใช้แสดงในส่วนของ history ของ NFT และ function ที่ใช้ใน nft contract นั้นจะมีทั้งหมดดังนี้

A picture containing text, screenshot, font, line

Description automatically generated

**รูปที่ 84 function \_incrementCounterTransfers**

* 1. \_incrementCounterTransfers ใช้สำหรับการนับจำนวนการขายของ nft เพื่อใช้สำหรับการแบ่งผลประโยชน์ครั้งแรกและครั้งต่อๆ ไป โดยจะมีรับ input เป็น tokenId และไม่มี output

A picture containing text, font, screenshot

Description automatically generated

**รูปที่ 85 function getCounterTransfers**

1. getCounterTransfers ใช้สำหรับการดูจำนวนการซื้อขายของ nft ซึ่งจะรับ input เป็น id ของ token ที่ต้องการและจะมีการ return จำนวนการซื้อขายของ nft

A picture containing text, screenshot, software, font

Description automatically generated

**รูปที่ 86 function tokenURI**

1. tokenURI ซึ่งจะรับ input เป็น id ของ token ที่ต้องการ และจะมีการ return ข้อมูล URI ของ nft

A screen shot of a computer

Description automatically generated with medium confidence

**รูปที่ 87 function mint**

1. mint ซึ่ง function นี้จะมีหน้าที่เป็นตัวกลางระหว่างการ set ข้อมูล uri และการสร้างข้อมูลอื่นๆ ของ nft ซึ่งมีการรับ input อยู่ 3 ตัวคือ array ของ address ของ collaborator, array ของ percent ของแต่ละ collaborator และ uri ของรูปภาพ และไม่มี output

A screen shot of a computer program

Description automatically generated with low confidence

**รูปที่ 88 function \_safeMint**

1. \_safeMint โดย function นี้จะหน้าที่เป็น ตัวกลางก่อนการ mint ซึ่งมีการรับ input สำหรับการ mint ดังนี้ to, collaborator, collaboratorPercent, tokenId เพื่อส่งต่อไปให้ function ของการสร้าง nft และ function นี้ไม่มีการ output

A screen shot of a computer

Description automatically generated with medium confidence

**รูปที่ 89 function \_mint**

1. \_mint โดย function สำหรับการ mint นี้จะเป็นการสร้าง nft ซึ่งมีการรับ input คือ to, collaborator, collaboratorPercent, tokenId และ function นี้ไม่มี output

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 90 function \_setTokenURI**

1. \_setTokenURI เป็น function สำหรับการ set ข้อมูลของ token uri โดยมีการรับ input คือ id ของ token และ uri ของรูปภาพ และไม่มี output

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 91 function balanceOf**

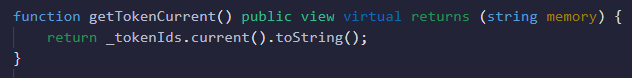
1. balanceOf เป็น function สำหรับการเช็คจำนวน nft ทั้งหมดที่เจ้าของ address เป็นเจ้าของอยู่ โดยที่จะมี input คือ owner และจะมี output คือจำนวน nft ทั้งหมดที่เป็นเจ้าของอยู่

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 92 function owerOf**

1. ownerOf เป็น function สำหรับการเช็คเจ้าของ token โดยที่จะรับ Input เป็น tokenId และจะมี output คือเจ้าของ nft เป็น address



**รูปที่ 93 function getTokenCurrent**

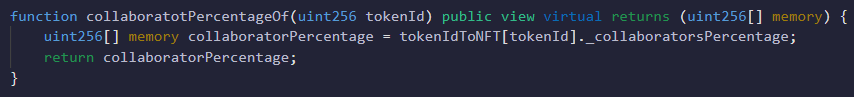
1. getTokenCurrent เป็น function สำหรับการเช็คจำนวน token ในปัจจุบันทั้งหมด โดยที่ไม่มี input แต่จะมี output เป็นจำนวนทั้งหมดของ nft

A picture containing text, font, screenshot

Description automatically generated

**รูปที่ 94 function collaboratorOf**

* 1. collaboratorOf เป็น function สำหรับการเช็ค collaborator ของ token โดยที่จะรับ Input เป็น tokenId และจะมี output คือ address ของ collaborator



**รูปที่ 95 function collaboratorPercentOf**

* 1. collaboratorPercentOf เป็น function สำหรับการเช็ค per ของ collaborator ของ token โดยที่จะรับ Input เป็น tokenId และจะมี output คือ จำนวน percent ของ collaborator

A picture containing screenshot, text

Description automatically generated

**รูปที่ 96 function name**

* 1. name เป็น function สำหรับเช็คชื่อของ nft contract ที่ได้ทำการ set ไว้ใน constructor ซึ่งไม่มี Input แต่จะ return ชื่อของ nft contract

A picture containing screenshot, text

Description automatically generated

**รูปที่ 97 function symbol**

* 1. symbol เป็น function สำหรับเช็ค symbol ของ nft contract ที่ได้ทำการ set ไว้ใน constructor ซึ่งไม่มี Input แต่จะ return symbol ของ nft contract

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 98 function \_tokenURI**

* 1. \_tokenURI เป็น function สำหรับดู uri ของ nft ที่จะมีการเรียกจาก function tokenURI โดยที่จะมีการรับ input เป็น tokenId และมี output เป็น uri ของ tokenId return ไปให้ function tokenURI

A picture containing screenshot, text, font, line

Description automatically generated

**รูปที่ 99 function \_baseURI**

* 1. \_baseURI เป็น function สำหรับ return string ว่างกลับไปให้ function ที่เรียกใช้งาน โดยที่ไม่มี input แต่มี output เป็น string ว่าง

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 100 function approve**

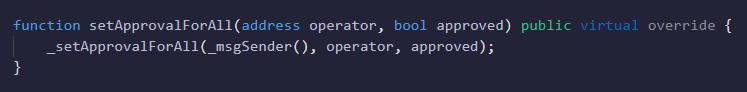
* 1. approve เป็น function สำหรับการเช็คต่างๆ เพื่อที่จะอนุมัติก่อนการโอน token โดยที่จะรับ Input เป็น to, tokenId และจะไม่มี output

A picture containing text, screenshot, font

Description automatically generated

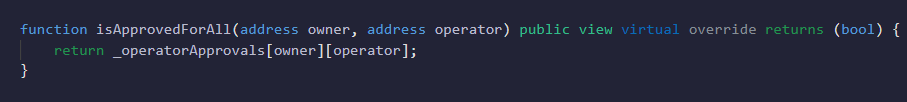
**รูปที่ 101 function getApproved**

* 1. getApproved เป็น function สำหรับ invalid ของ tokenId โดยที่จะมี input เป็น tokenId และจะมีการ และจะมี output เป็น address



**รูปที่ 102 function setApprovealForAll**

* 1. setApprovalForAll เป็น function สำหรับเป็นตัวกลางสำหรับการเรียกใช้ \_setAppovalForAll โดยที่จะมี input เป็น operator, approved แต่ไม่มี output



**รูปที่ 103 function isApprovedForAll**

* 1. isApprovedForAll เป็น function ที่ใช้สำหรับการเช็คการอนุมัติ โดยที่มีการรับinput เข้ามาเช็คคือ owner, operator และมี output เป็นผลการเช็คเป็น bool

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 104 function transferFrom**

* 1. transferForm เป็น function ที่เป็นตัวกลางสำหรับการเรียกใช้ transfer โดยที่จะมีการรับ input เพื่อที่จะส่งไปที่ function \_transfer คือ from, to และ tokenId โดยที่ไม่มี output

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 105 function safeTransferForm**

* 1. safeTransferForm เป็น function ที่เป็นตัวกลางสำหรับการเรียกใช้ safeTransferForm โดยที่จะมีการรับ input เพื่อที่จะส่งไปที่ function safeTransferForm คือ from, to และ tokenId โดยที่ไม่มี output

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 106 function safeTransferFrom**

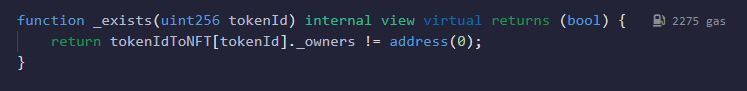
* 1. safeTransferFrom เป็น function ที่เป็นตัวกลางสำหรับการเรียกใช้ \_safeTransfer ซึ่งจะมีการเช็ค require เจ้าของ token โดยที่จะมีการรับ input เพื่อที่จะส่งไปที่ function \_safeTransfer คือ from, to, tokenId และ byte ของ data โดยที่ไม่มี output

A picture containing text, font, screenshot

Description automatically generated

**รูปที่ 107 function \_safeTransfer**

* 1. \_safeTransfer เป็น function ที่เป็นตัวกลางสำหรับการเรียกใช้ \_transfer ซึ่งจะมีการเช็ค ERC721Received โดยที่จะมีการรับ input เพื่อที่จะส่งไปที่ function \_safeTransfer คือ from, to, tokenId และ byte ของ data โดยที่ไม่มี output



**รูปที่ 108 function \_exitsts**

* 1. \_exitsts เป็น function ที่ใช้สำหรับการเช็ค address ของเจ้าของ nft owner ไม่เท่ากับ 0 โดยจะมีการรับ Input เป็น tokenId และมี output เป็น bool

A picture containing screenshot, text, font

Description automatically generated

**รูปที่ 109 function \_isApprovedOrOwner**

* 1. \_isApprovedOrOwner เป็น function ที่ใช้สำหรับการอนุมัติ address โดยที่จะมีการรับ input เป็น sender, tokenId และมี output เป็น bool

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 110 function \_transfer**

* 1. \_transfer เป็น function ที่ใช้สำหรับการเปลี่ยเจ้าของ nft โดยที่จะมีการรับ input เป็น from, to, tokenId และไม่มี output

A picture containing text, font, screenshot, line

Description automatically generated

**รูปที่ 111 function \_approve**

* 1. \_approve เป็น function ที่ใช้สำหรับการ set approvals ของ token โดยที่จะรับ input คือ to, tokenId และไม่มี output

A screen shot of a computer program

Description automatically generated with low confidence  
**รูปที่ 112 function \_setApprovalForAll**

* 1. \_setApprovalForAll เป็น function ที่ใช้สำหรับการ set operatorApprovals โดยที่จะรับ input คือ owner, operator, approved และไม่มี output

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 113 function \_requireMinted**

* 1. \_requireMinted เป็น function ที่ใช้สำหรับการเช็ค invalid ของ token Id โดยที่จะรับ input คือ tokenId และไม่มี output

A picture containing text, screenshot

Description automatically generated

**รูปที่ 114 function \_checkOnERC721Received**

* 1. \_checkOnERC721Received เป็น function ที่ใช้สำหรับการเช็ค ERC 721 Received โดยที่จะรับ input คือ from, to, tokenId, byte ของ data และมี output เป็น bool
     1. marketplace

marketplace contract นี้จะมี constructor คือ owner ของ contract ดังนี้

A screen shot of a computer

Description automatically generated with medium confidence

**รูปที่ 115 constructor ของ marketplace contract**

A picture containing text, screenshot, font, software

Description automatically generated

**รูปที่ 116 การกำหนดตัวแปรใน marketplace contract**

จากภาพเป็นการกำหนดตัวแปรที่จะใช้เก็บข้อมูลลงใน blockchain และในไฟล์ของ marketplace contract นี้จะมีทั้งหมด 6 function, 3 event

A screen shot of a computer code

Description automatically generated with low confidence

**รูปที่ 117 event ใน marketplace contract**

3 event คือ event List, Unlist, Sale โดยที่สร้างไว้สำหรับการดักจับ logs ของ event ของ transactionHash เพื่อใช้แสดงในส่วนของ history ของ NFT

และ function ที่ใช้ใน marketplace contract นั้นจะมีทั้งหมดดังนี้

A picture containing text, screenshot, software

Description automatically generated

**รูปที่ 118 function listedNFTItem**

1. listedNFTItem เป็น function ที่ใช้สำหรับ nft ที่เราทำการสร้างเสร็จแล้วมาทำการลงขายใน marketplace โดยที่ input คือ nftContract, tokenId, price และไม่มี output

A picture containing text, screenshot, software, font

Description automatically generated

**รูปที่ 119 function itemFromTokenId**

1. itemFromTokenId เป็น function ที่ใช้สำหรับการเช็ค itemId ของ token โดยที่จะ input ดังนี้คือ tokenId และมี output คือ itemId ของ token

A picture containing text, screenshot, font, software

Description automatically generated

**รูปที่ 120 function priceFromTokenId**

1. priceFromTokenId เป็น function ที่ใช้สำหรับการเช็ค price ของ token โดยที่จะ input ดังนี้คือ

tokenId และมี output คือ price ของ token

A picture containing text, screenshot, software

Description automatically generated  
**รูปที่ 121 function saleNFTItem**

1. saleNFTItem เป็น function ที่ใช้สำหรับการขาย nft โดยที่จะ input ดังนี้คือ

nftContract, itemId และไม่มี output

A picture containing text, screenshot, font

Description automatically generated

**รูปที่ 122 function unListNFTItem**

1. unListNFTItem เป็น function ที่ใช้สำหรับการยกเลิกการวางขาย nft ในระบบโดยที่จะมี input ดังนี้คือ nftContract, itemId และไม่มี output

A picture containing text, screenshot, software, font

Description automatically generated

**รูปที่ 123 function fetchNFTItem**

1. fetchNFTItem เป็น function การดู nft ทั้งหมดที่ทำการลงขายอยู่ในระบบ โดยที่จะไม่มี input แต่มี output คือ array ของ market item