



# **Sistem de monitorizare a prețului unei colecții de NFT-uri**

Autor: Maria Onița CREȚIU

Îndrumător: Eugen Cătălin STOICA

# CUPRINS

<b>1. Introducere.....</b>	<b>3</b>
<b>1.1 Cerințe.....</b>	<b>3</b>
<b>1.2 Specificații .....</b>	<b>3</b>
<b>2. Proiectare .....</b>	<b>4</b>
<b>3. Implementare .....</b>	<b>6</b>
<b>4. Testare .....</b>	<b>6</b>
<b>5. Concluzii .....</b>	<b>9</b>

# 1. Introducere

## 1.1 Cerințe

Scopul proiectului este implementarea un sistem distribuit pentru monitorizarea prețului unei colecții de NFT-uri.

Sistemul este compus din următoarele aplicații:

- **Server**

Aplicația server este responsabilă cu salvarea detaliilor unei colecții NFT într-o baza de date. Datele provin prin interogarea API-ului marketplace-ului OpenSea la un interval de 15 minute. De asemenea aplicația primește de la clientul mobile o colecție pentru care se va interoga la un interval de 1 minut prețul minim anterior cu noul preț, iar dacă noul preț este mai mic cu 10%, clientul va fi înștiințat printr-un SMS. Va exista o singura astfel de aplicație în sistem (**1 instanță**).

- **Client**

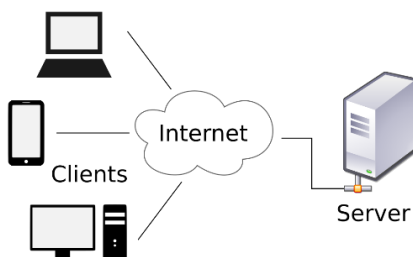
Se implementează pentru Android sau iOS, și aplicația va avea rolul de a trimite colecția NFT la care dorim să îi monitorizăm prețul minim. Colecția trimisă va fi monitorizată o dată la 1 minut, iar în cazul în care noul preț este mai mic cu 10% față de precedentul, clientul va fi înștiințat printr-un SMS. Vor putea exista mai multe astfel de aplicații în sistem (**n instanțe**).

- **Monitor**

Este o aplicație **WEB**, ce permite vizualizarea detaliilor colecțiilor de NFT salvate. Aceste detalii sunt afișate și vor putea fi editate (modificate valori/șterse).

## 1.2 Specificații

Sistemul impletat este format din trei aplicații distincte ce comunică între ele. Arhitectura este de tip client/server, clientul face un request automat la server pentru a face operațiile CRUD pe baza de date. Ambii clienți (monitor și mobile) sunt contactați la același server.



## 2. Proiectare

Sistemul va fi proiectat prin intermediul următoarelor tipuri de diagrame UML:

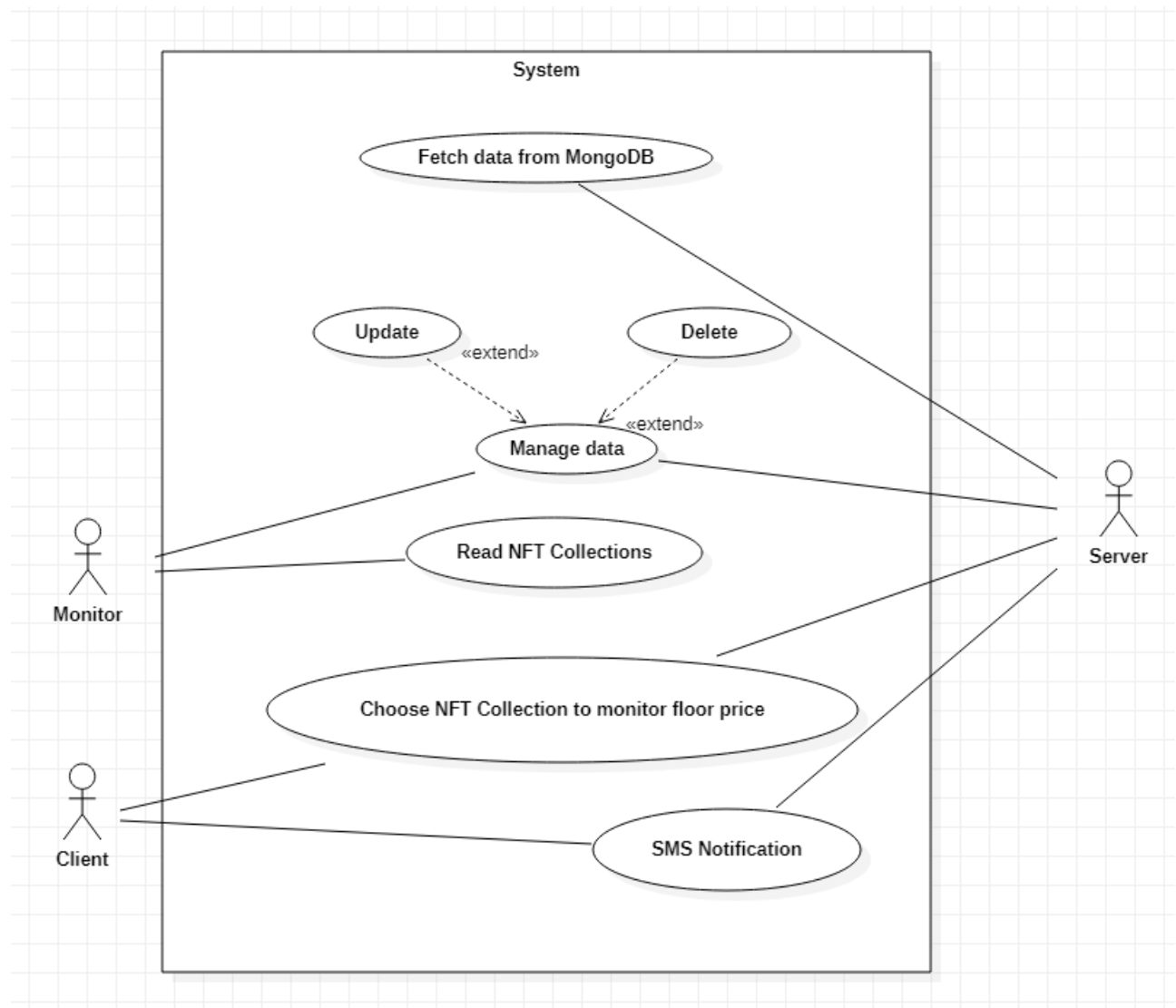


Figura 2.1: Diagrama cazurilor de utilizare

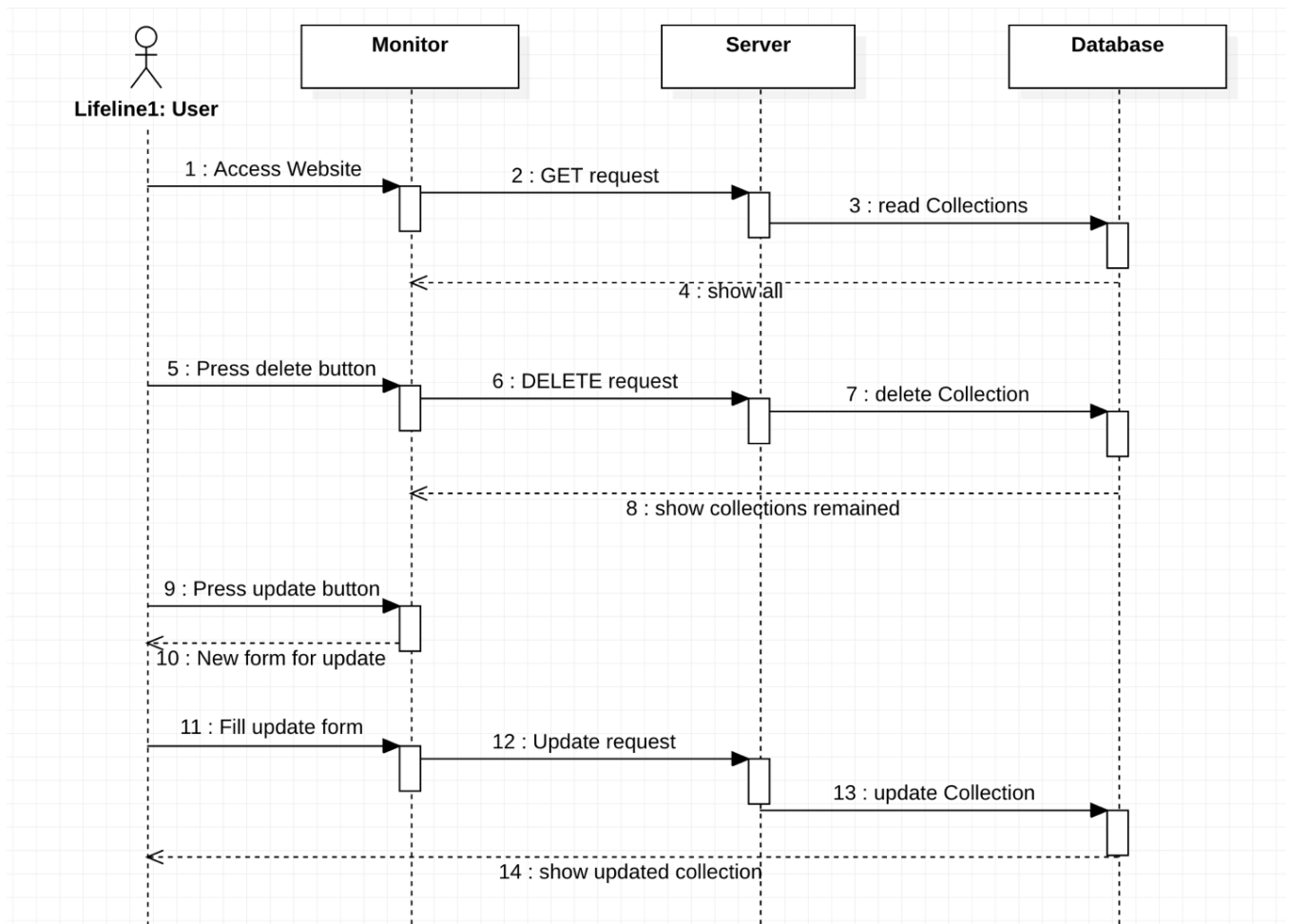


Figura 2.3: Diagrama secvențială

Baza de date a fost conceputa sub forma unei tabele care contine detaliile colectiilor de NFT-uri.

collection	
PK	_id floorPrice averagePrice totalSupply numOwners name totalVolume sevenDaySales date

### 3. Implementare

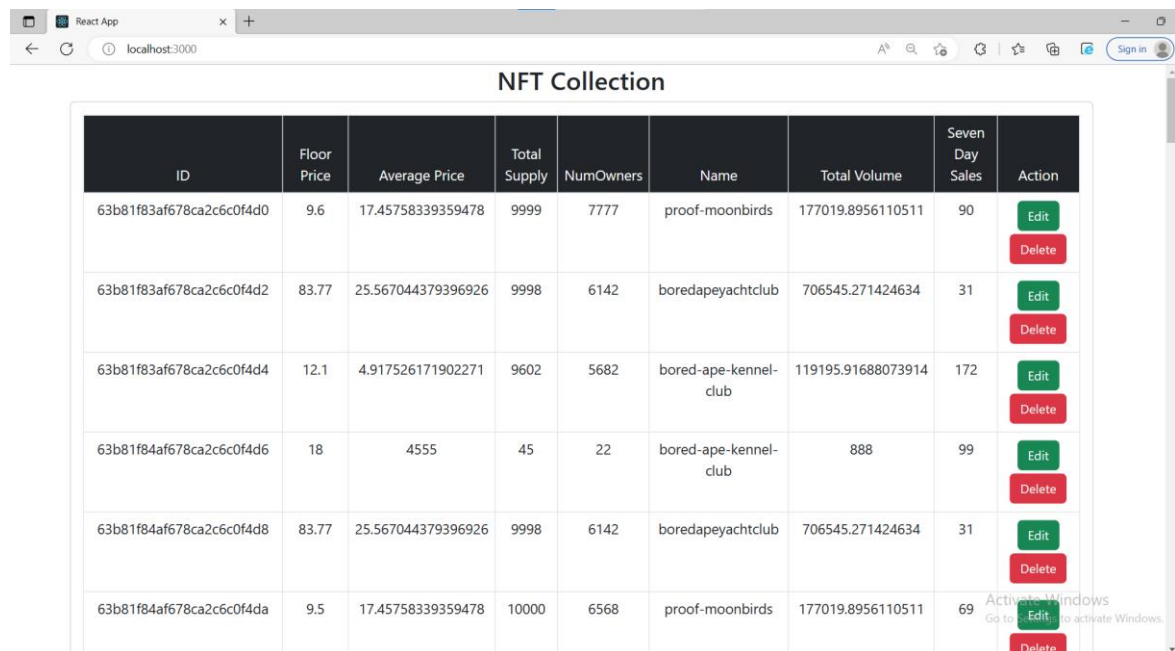
Aplicația este structurată în trei părți distincte : server, monitor și aplicație mobilă.

Serverul este implementat în mediul de execuție Node.js. Acesta execută cod JavaScript în afara unui navigator web. Acesta este sub forma de MVC(Model-View-Controller). Scopul MVC este de a ține separate logica business-ului și interfața utilizator, astfel încat cei care întrețin aplicația să schimbe mult mai ușor o parte, fără a afecta celălalte părți. Modelul conține informațiile (name, floor price, total supply, total volume etc.) și regulile business; view-ul conține elemente din interfața utilizator (texte, input-uri ale formularelor etc); controller-ul gestionează comunicatia dintre model si view(aici se află operațiile CRUD).

Pagina web este dezvoltată cu ajutorul bibliotecii React.

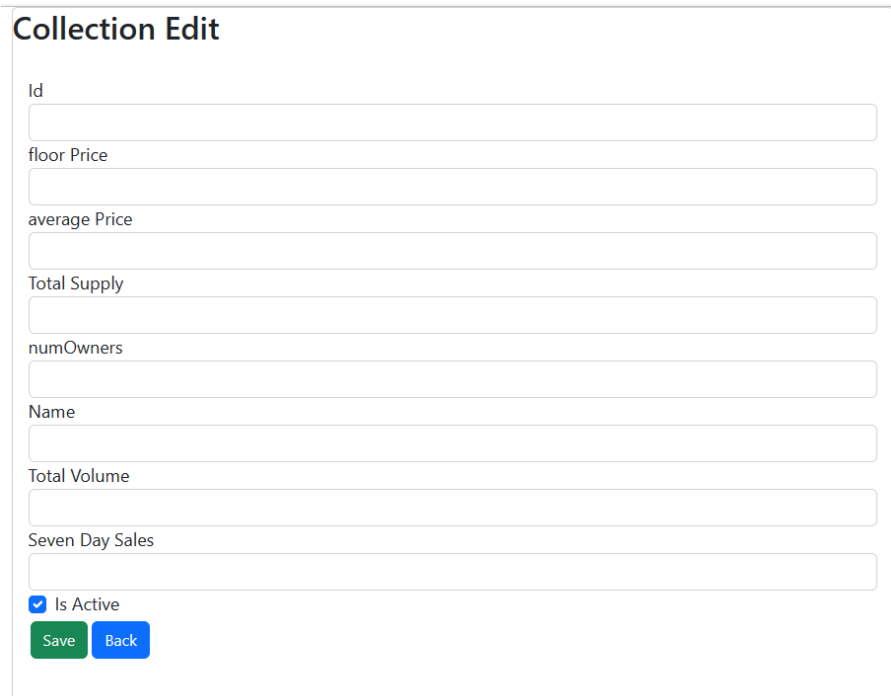


Aceasta permite crearea unui frontend scalabil și ușor de utilizat pentru aplicațiile web. Aici se pot monitoriza colecțiile de NFT-uri salvate, împreună cu detaliile despre ele. Sunt prezentate sub formă de card-uri pentru o vizualizare mai plăcută și se pot șterge sau update.

A screenshot of a web browser displaying a table titled "NFT Collection". The table has columns for ID, Floor Price, Average Price, Total Supply, NumOwners, Name, Total Volume, Seven Day Sales, and Action. The Action column contains "Edit" and "Delete" buttons for each row. The browser's address bar shows "localhost:3000".

ID	Floor Price	Average Price	Total Supply	NumOwners	Name	Total Volume	Seven Day Sales	Action
63b81f83af678ca2c6c0f4d0	9.6	17.45758339359478	9999	7777	proof-moonbirds	177019.8956110511	90	<button>Edit</button> <button>Delete</button>
63b81f83af678ca2c6c0f4d2	83.77	25.567044379396926	9998	6142	boredapeyachtclub	706545.271424634	31	<button>Edit</button> <button>Delete</button>
63b81f83af678ca2c6c0f4d4	12.1	4.917526171902271	9602	5682	bored-ape-kennel-club	119195.91688073914	172	<button>Edit</button> <button>Delete</button>
63b81f84af678ca2c6c0f4d6	18	4555	45	22	bored-ape-kennel-club	888	99	<button>Edit</button> <button>Delete</button>
63b81f84af678ca2c6c0f4d8	83.77	25.567044379396926	9998	6142	boredapeyachtclub	706545.271424634	31	<button>Edit</button> <button>Delete</button>
63b81f84af678ca2c6c0f4da	9.5	17.45758339359478	10000	6568	proof-moonbirds	177019.8956110511	69	<button>Edit</button> <button>Delete</button>

Figura 3.1: Pagina Web



**Collection Edit**

Id

floor Price

average Price

Total Supply

numOwners

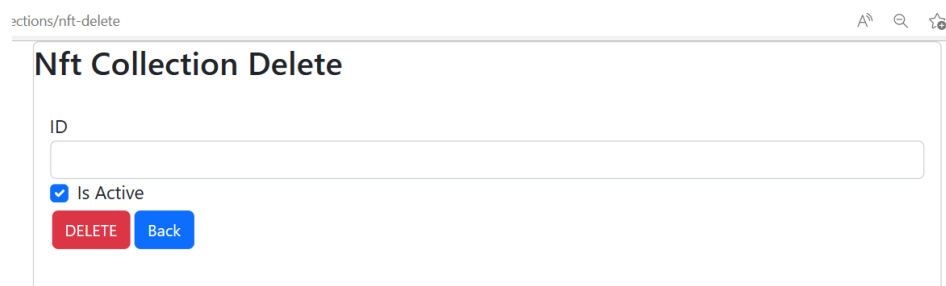
Name

Total Volume

Seven Day Sales

☒ Is Active

Figura 3.2: Formul pentru Update



actions/nft-delete

**Nft Collection Delete**

ID

☒ Is Active

Figura 3.3: Formul pentru Delete

Aplicația mobilă este implementată în Android Studio, folosind terminologiile Java și XML.

Android Studio este mediul oficial de dezvoltare integrată (IDE) pentru sistemul de operare Android Google, construit pe software-ul IntelliJ IDEA al JetBrains și conceput special pentru dezvoltarea Android.

Fiecare proiect din Android Studio conține unul sau mai multe module cu fișiere de cod sursă și fișiere de resurse. Tipurile de module includ:

- Modulele aplicației Android
- module de bibliotecă
- Module Google App Engine

În mod implicit, Android Studio afișează fișierele de proiect în vizualizarea proiectului Android, așa cum se arată în figura 1. Această vizualizare este organizată pe module pentru a oferi acces rapid la fișierele sursă cheie ale proiectului. Toate fișierele de compilare sunt vizibile la nivelul superior, sub Scripturi Gradle.

Fiecare modul de aplicație conține următoarele foldere:

- manifest: Conține AndroidManifest.xml fișierul.
- java: Conține fișierele de cod sursă Java, inclusiv codul de testare JUnit.
- res: Conține toate resursele non-cod, cum ar fi aspecte XML, șiruri de interfață și imagini bitmap.

Rularea aplicației se face cu ajutorul unui emulator oferit de Android Studio.

Emulatorul Android simulează dispozitivele Android pe computerul dvs., astfel încât să vă puteți testa aplicația pe o varietate de dispozitive și niveluri API Android fără a fi nevoie să aveți fiecare dispozitiv fizic.

Arhitectura aplicației mobile:

- Adapters
- Networking
- Activity

Clasa Activity este o componentă crucială a unei aplicații Android, iar modul în care activitățile sunt lansate și reunite este o parte fundamentală a modelului de aplicație al platformei. Spre deosebire de paradigmele de programare în care aplicațiile sunt lansate cu o metodă main(), sistemul Android inițiază codul într-o instanță activity invocând anumite metode de apel invers care corespund unor etape specifice ale ciclului său de viață.

Pentru a gestiona apelurile API REST în aplicație am folosit libraria Retrofit. Retrofit este un client REST sigur pentru Android și Java, care își propune să faciliteze consumul de servicii web RESTful.

Am folosit RecyclerView pentru afișarea eficientă a colecțiilor de NFT-uri. Adaptoarele oferă o legătură de la un set de date specific aplicației la vizualizări care sunt afișate într-un RecyclerView.



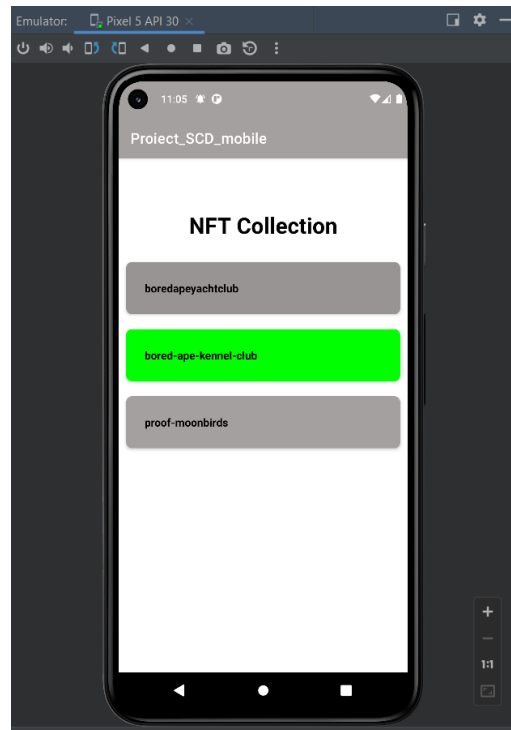


Figura 3.4: Aplicația Mobile

## 4. Testare

Pentru testarea backend-ului am folosit aplicația Postman. Am verificat fiecare endpoint pentru a ne asigura că request-ul la baza de date este realizat cu succes.

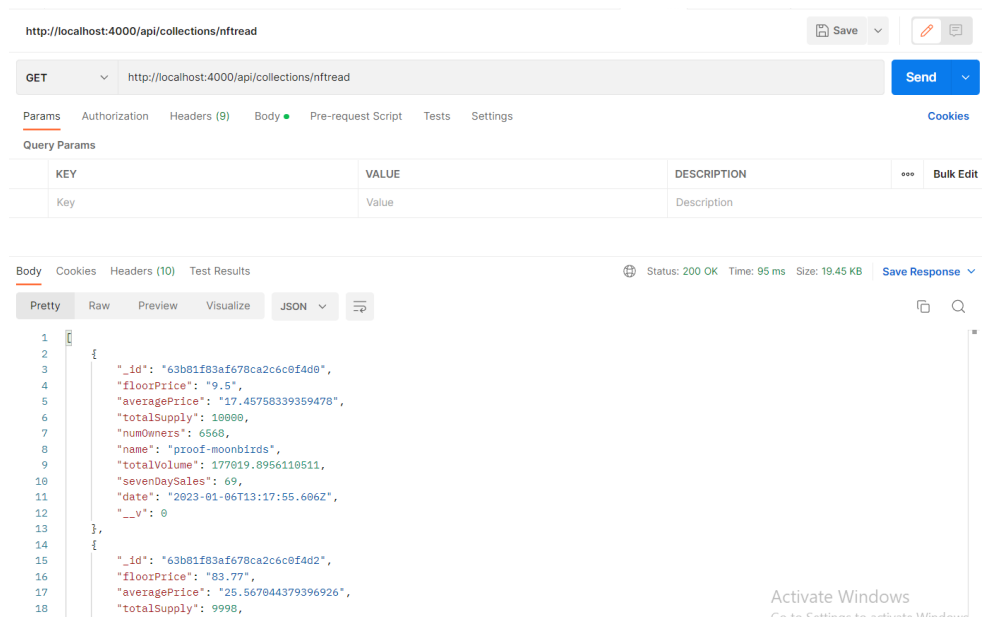


Figura 4.1: Testare GET

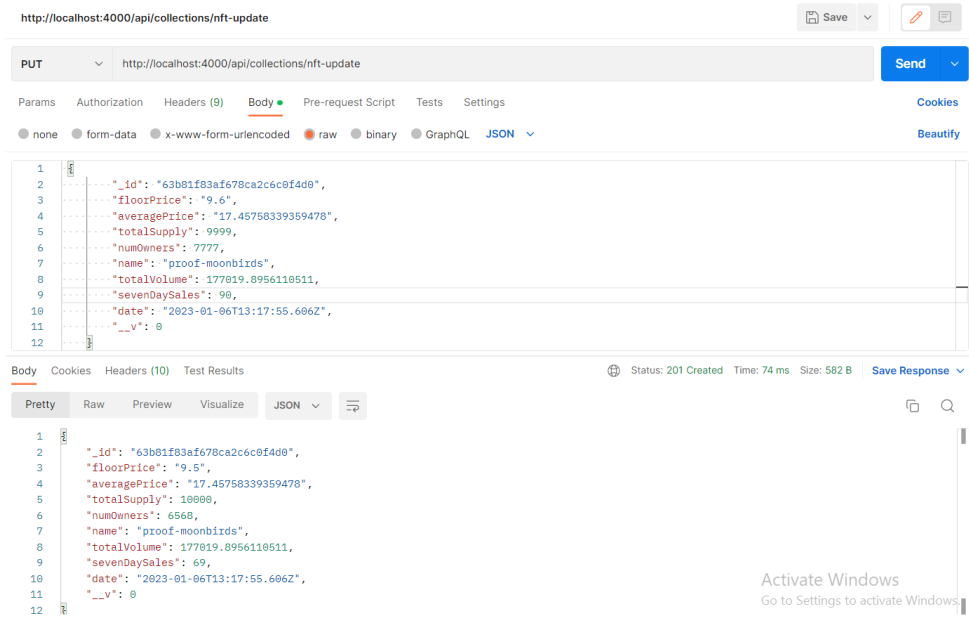


Figura 4.2: Testare PUT

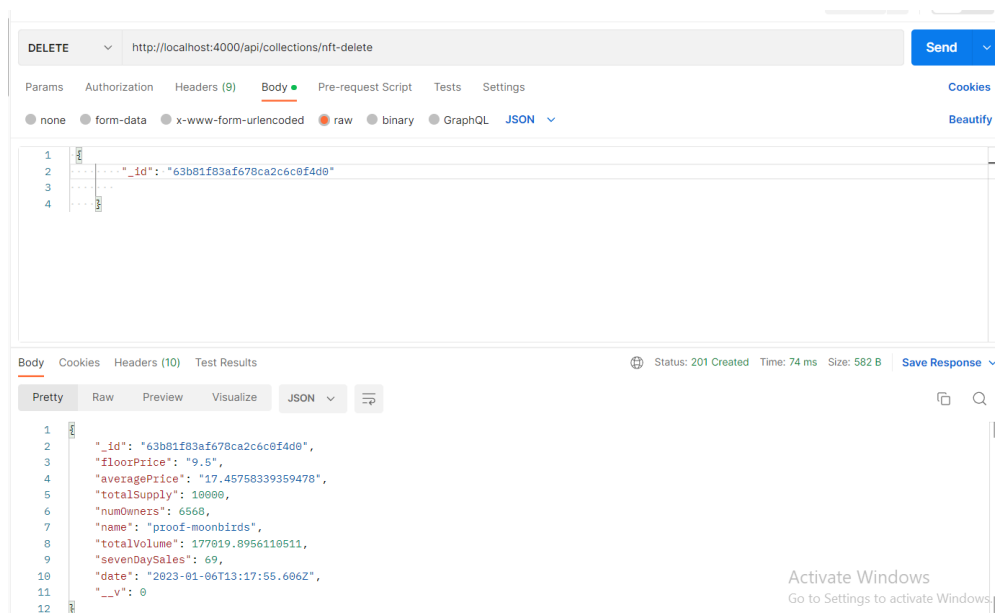
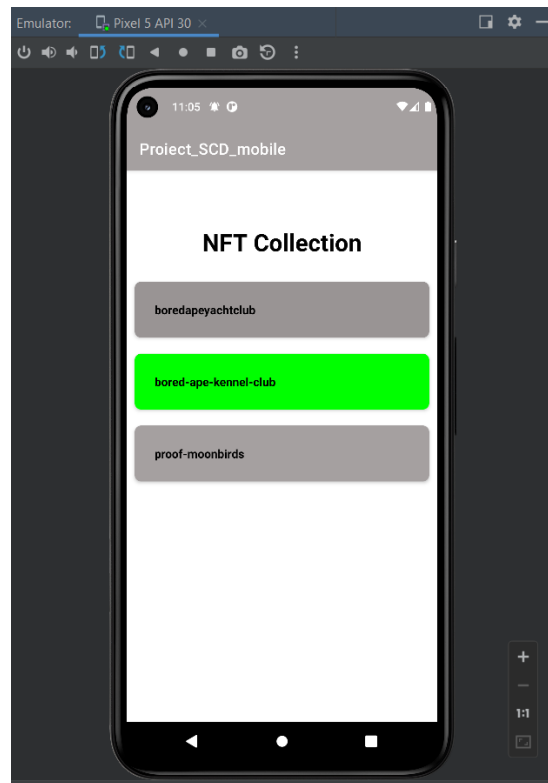


Figura 4.3: Testare DELETE

Pentru testarea frontend-ului am testat mai întâi afișarea detaliilor colecțiilor de NFT-uri. După ce ne-am asigurat că sunt afișate stats-urile corecte am început testarea butoanelor de delete și de update. La început se updatea doar un singur câmp, dar apoi aceasta problema a fost rezolvată și în final toate testările s-au realizat cu succes.

Pentru a testa aplicația mobile am verificat în primul rând ca numele colecțiilor să apară în emulatorul din Android Studio.



## 5. Concluzii

Am realizat o aplicație care monitorizează prețul unor colecții de NFT-uri formata din trei părți distincte: sever, monitor și aplicație mobilă. Aplicația Web permite vizualizarea colecțiilor de NFT-uri și editarea acestora, iar din aplicația mobilă se poate selecta o colecție pentru care se monitorizează prețul și în cazul în care acesta scade cu 10%, clientul este înștiințat printr-un SMS. Toate obiectivele proiectului au fost realizate.