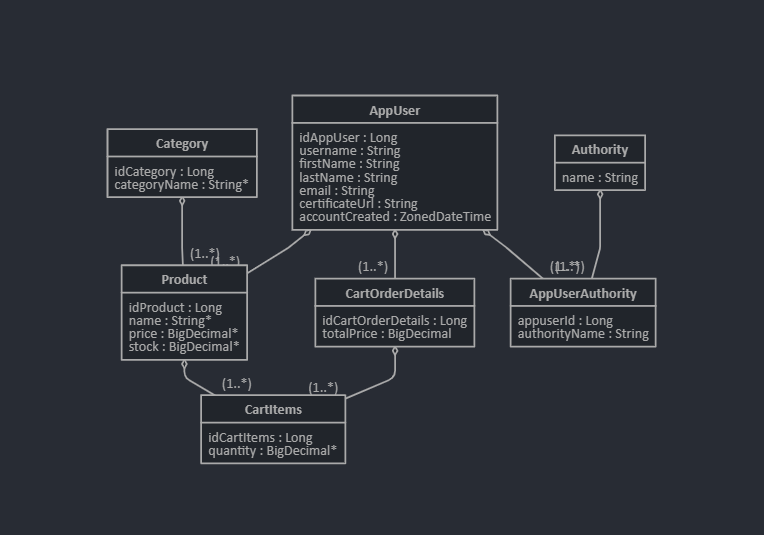
**LocalGrower**

Voinea Alexandru-Daniel-Gabriel

Grupa 405

Modelul bazei de date:

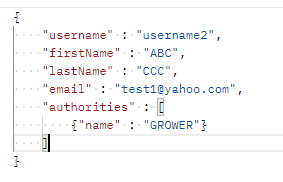
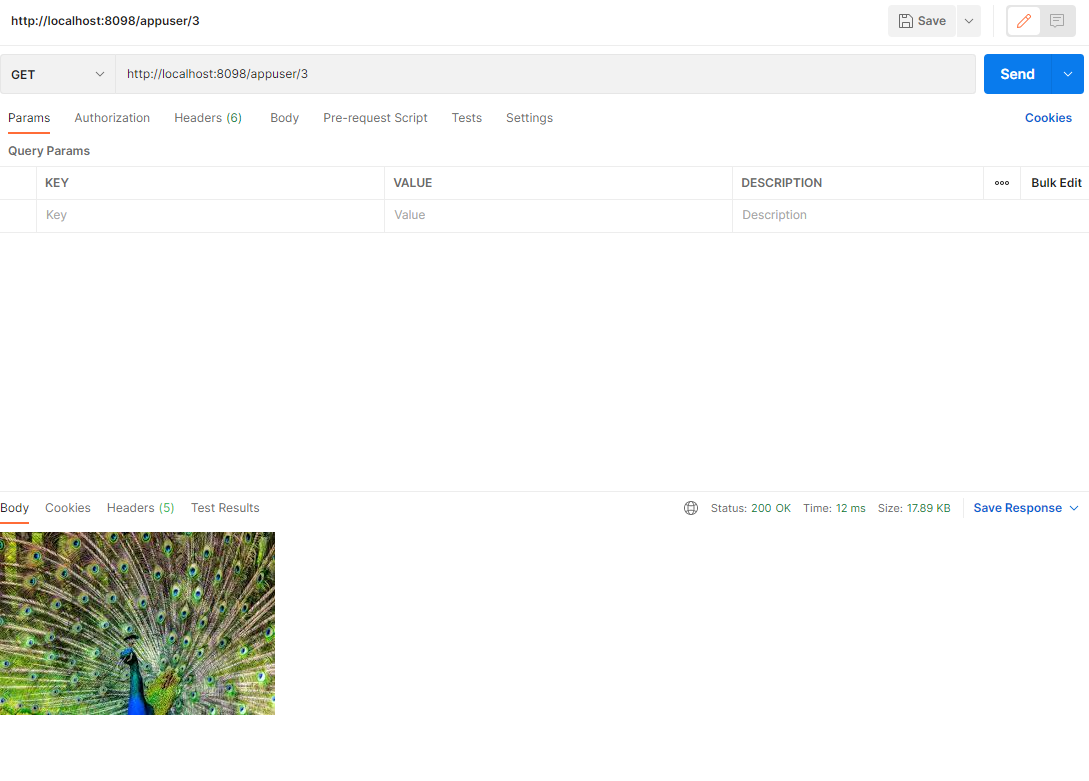


Functionalitati principale:

1. Avem nevoie de o functionalitate prin care producatorii locali sa isi poata incarca certificatul, in baza caruia urmeaza a primii drepturi de producatori
2. Dupa ce un producator si-a primit drepturile, acesta poate sa-si incarcare produsele in aplicatie
3. Avem nevoie de o functionalitate prin care un utilizator isi poate adauga produsele dorite in cos
4. Avem nevoie de o functionalitate prin care un utilizaztor isi poate finaliza o comanda
5. Avem nevoie de diverse functionalitati prin care un administrator sau un producator poate modifica diverse specificatii ale unui produs

Metode din controler:

AppUserController:

* POST : (curl --location --request POST '<http://localhost:8098/appuser>' \
* --header 'Content-Type: application/json' \). Primește pe body un obiect de tipul AppUserDTO de forma :  și creează o instanță de tipul AppUser al carei idAppUser este autogenerat; De asemenea, prin aceasta metoda se creeaza si o legatura intre entitatile AppUser si Authorities.
* PUT : ( curl --location --request PUT '<http://localhost:8098/appuser/role/3>'). In cazul in care entitate AppUser corespondenta id-ul dat in input are un certificatUrl nenul, atunci va face update pe entitate si ii va garanta acesteia drepturi de producator(GROWER);
* GET: (curl --location --request GET '<http://localhost:8099/appuser>') : Acesta metoda returneaza toti utilizatorii din aplicatie;
* POST: ( curl --location --request POST '<http://localhost:8099/appuser/upload/certificate?id=3>' \ --form '[file=@"/C:/Users/Alex/Desktop/test.jfif](mailto:file=@"/C:/Users/Alex/Desktop/test.jfif)"'). Primește ca RequestParam un id de tip Long si in file de tip MultipartFile si insereaza pentru un utilizator practic certificator.
* GET: (curl --location --request GET '<http://localhost:8098/appuser/3>'). Afiseaza un certificat incarcat de un utilizator. 

CategoryController:

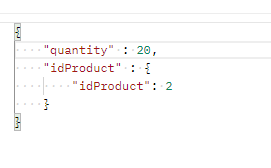
* POST <http://localhost:8099/category/cat> : Primește pe body un CategoryDTO { „categoryName” : „legume” } și creează o categorie noua cu numele dat;
* PUT [http://localhost:8099/category/1](http://localhost:8099/artists1) : Primeste un PathVariable idCategory de tipul Long si un body de tipul CategoryDTO ;
* GET [http://localhost:8099/category](http://localhost:8099/artists/{na) : Returneaza toate categoriile ;
* GET [http://localhost:8099/category/1](http://localhost:8099/artists/member/%7bname%7d) : Returneaza categoria pentru id-ul specificat ca PathVariable(respectiv 1)

ProductController:

* POST : (curl --location --request POST '<http://localhost:8098/product>' \--header 'Content-Type: application/json' \ ) : Primește pe body un ProductDTO de forma : si insereaza un produs nou
* PUT (curl --location --request PUT '<http://localhost:8098/product/2>' \
* --header 'Content-Type: application/json' \) . Primeste pe input un id ca PathVariable si un body de tipul ProductDTO : 
* GET http://localhost:8099/product : Returneaza toate produsele;
* GET [http://localhost:8099/product/1](http://localhost:8099/artists/member/%7bname%7d) : Returneaza produsul cu id-ul specificat ca PathVariable(respectiv 1)

CartItemsController:

-POST : (curl --location --request POST '<http://localhost:8098/cart-items/add/product/1>' \ --header 'Content-Type: application/json' \) si primeste pe body un obiect de tipul CartItemsDTO:

Astfel, se insereaza un produs in cosul de cumparaturi.

CartOrderDetailsController:

-PUT: (curl --location --request PUT '<http://localhost:8098/cart-order-details/1>' ) . Aceasta metoda are rolul de a face update pe cosul de cumparaturi cu id-ul dat ca PathVariable. Astfel, dupa ce apelam aceasta metoda vom modifica statusul in ‘FINALIZAT’.

