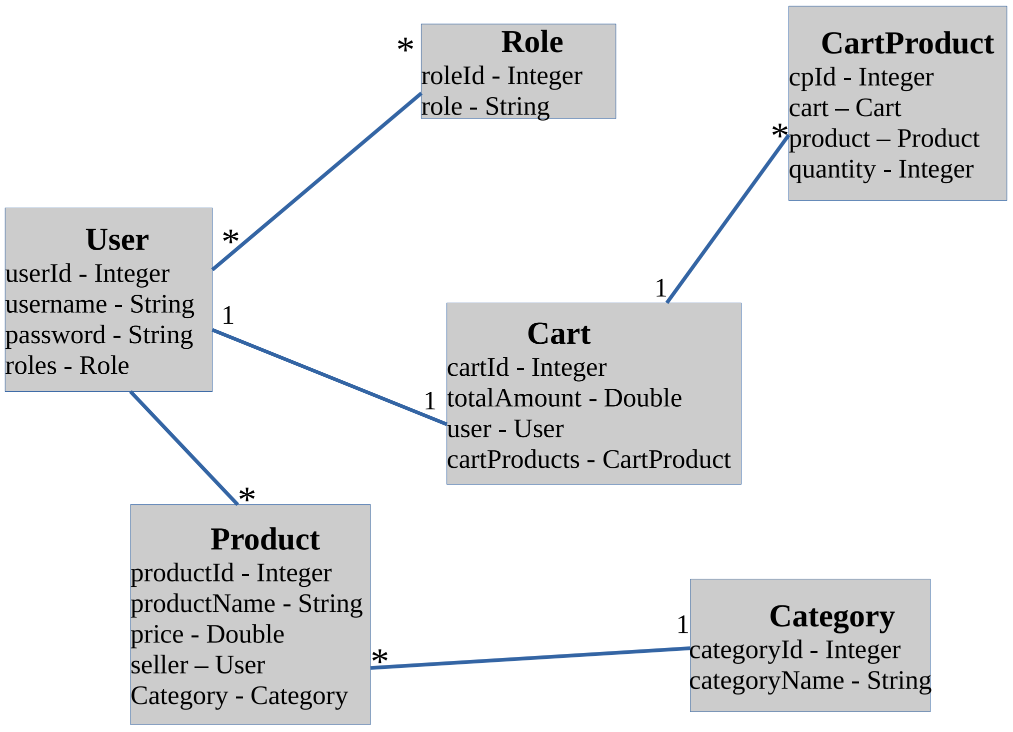
McDiffyStore is a vendor who sell various products ranging across different categories. They hired you to develop a distributed e-commerce platform to move their business online. As an initial MVP you are required to develop a restful API backend application in springboot.
Here is the requirement for the application.

Database models are already created and initialised as below:



DB initialised with following default data:

<u>Categories:</u>	<u>Role</u>	<u>User</u>	<u>Cart</u>	<u>UserRole</u>
categoryId, categoryName 'Fashion' 'Electronics' 'Books' 'Groceries' 'Medicines'	**role** ‘CONSUMER’ ‘SELLER’	**username, password** 'jack', 'pass_word' ‘bob', 'pass_word' 'apple', 'pass_word' ‘glaxo’, 'pass_word	**totalAmount, userId** 20, 1 0, 2	**userId, role** 1, 1 2, 1 3, 2 4, 2

<u>Product</u>	<u>CartProduct</u>
price, productName, categoryId, sellerId 29190, 'Apple iPad 10.2 8th Gen WiFi iOS Tablet', 2, 3 10, 'Crocin pain relief tablet', 5, 4	**cartId, productId, quantity** 1, 2, 2

Your job is to create the following APIs, use JWT authentication with roles to protect consumer and seller specific endpoints. The JWT is included in the header with key **JWT** and value as jwt token.

- Consumers can search, add, update and delete items in cart.
- Sellers can add, update and delete products to the database.
- APIs preceeding with /api/public are public APIs and can be accessed by anyone.
- APIs preceeding with /api/auth/consumer are authenticated and consumer APIs.
- APIs preceeding with /api/auth/seller are authenticated and seller APIs
- if authenticated endpoints are accessed without JWT, return 401.
- if a consumer endpoint is accessed with seller JWT or viceverse, return 403.

<u>Below are public endpoints:</u>

/api/public/product/search – GET - This endpoint takes a query parameter 'keyword' and returns all the matching products containing the keyword either in productName or categoryName.	
Example: */api/public/product/search?keyword="tablet"*	**returns**: *[{"productId":1,"productName":"Apple iPad 10.2 8th Gen WiFi iOS Tablet","price":29190.0,"category":{"categoryName":"Electronics"}}, {"productId":2,"productName":"Crocin pain relief tablet","price":10.0,"category":{"categoryName":"Medicines"}}]*
/api/public/product/search	**status** 400

***/api/public/login* -** POST - takes username and password in json body, authenticates and returns JWT. Can authenticate both consumer and seller.	
request body - *{"username": "bob","password": "pass_word"}*	**status** *200*, **response body** – *eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c*
request body - *{"username": "bob","password": "password"}*	**status** 401

<u>Below are all authenticated endpoints:</u>

***/api/auth/consumer/cart* -** *GET* - returns the consumer's cart.	
/api/auth/consumer/cart	**status** *200*, **response body** – *{"cartId":1,"totalAmount":20.0,"cartProducts":[{"cpId":1,"product":{"productId":2,"productName":"Crocin pain relief tablet","price":10.0,"category":{"categoryName":"Medicines"}},"quantity":2}]}*

/api/auth/consumer/cart - *POST* – takes a Product json in request body and adds it to the consumer’s cart.	
Request body - *{"productId":3,"category":{"categoryName":"Electronics","categoryId":"2"},"price":"98000.0","productName":"iPhone 12"}*	status *200*
If product already in cart is added again	status 409

/api/auth/consumer/cart - *PUT* – takes a CartProduct json in request body and updates the quantity of the product in cart. If product is not in cart, add the product to cart with supplied quantity. If the quantity of the product is zero then delete the product from the cart.	
Request body - *{"product":{"productId":3,"category":{"categoryName":"Electronics","categoryId":"2"},"price":"98000.0","productName":"iPhone 12"},"quantity":3}*	status *200*

/api/auth/consumer/cart - *DELETE* – takes a Product json in request body and removes the product from the cart.	
Request body - *{"productId":3,"category":{"categoryName":"Electronics","categoryId":"2"},"price":"98000.0","productName":"iPhone 12"}*	status *200*

/api/auth/seller/product - *GET* – return all the products owned by seller. ***/api/auth/seller/product/{productId} -*** *GET* – return the product identified by the supplied path parameter productId.	
/api/auth/seller/product	**status** *200,* ***response body*** *-* *[{"productId":1,"productName":"Apple iPad 10.2 8th Gen WiFi iOS Tablet","price":29190.0,"category":{"categoryName":"Electronics"}}]*
/api/auth/seller/product/1	**status** 200, **response body -** *{"productId":1,"productName":"Apple iPad 10.2 8th Gen WiFi iOS Tablet","price":29190.0,"category":{"categoryName":"Electronics"}}*

/api/auth/seller/product - *POST* – takes a product json in request body and saves it to database.	
Request body - *{"productId":3,"category":{"categoryName":"Electronics","categoryId":"2"},"price":"98000.0","productName":"iPhone 12 Pro Max"}*	**status** *201,* ***redirect url*** *(created URI)-* *http://localhost/api/auth/seller/product/3*

***/api/auth/seller/product - ****PUT* – takes a product json in request body with mandatory product id and updates the product in the database.	
Request body - *{"productId":3,"category":{"categoryName":"Electronics","categoryId":"2"},"price":"98000.0","productName":"iPhone 12 Pro Max"}*	**status** *200*

***/api/auth/seller/product/{productId} - ****DELETE* – takes a productId path parameter and deletes the product from the database. If product not owned by seller return 404	
/api/auth/seller/product/1	**status** *200*
/api/auth/seller/product/2	**status** 404

Take a look at the testcases to understand more on how the validation works.
Good Luck and Start Coding!