

Java Backend Development

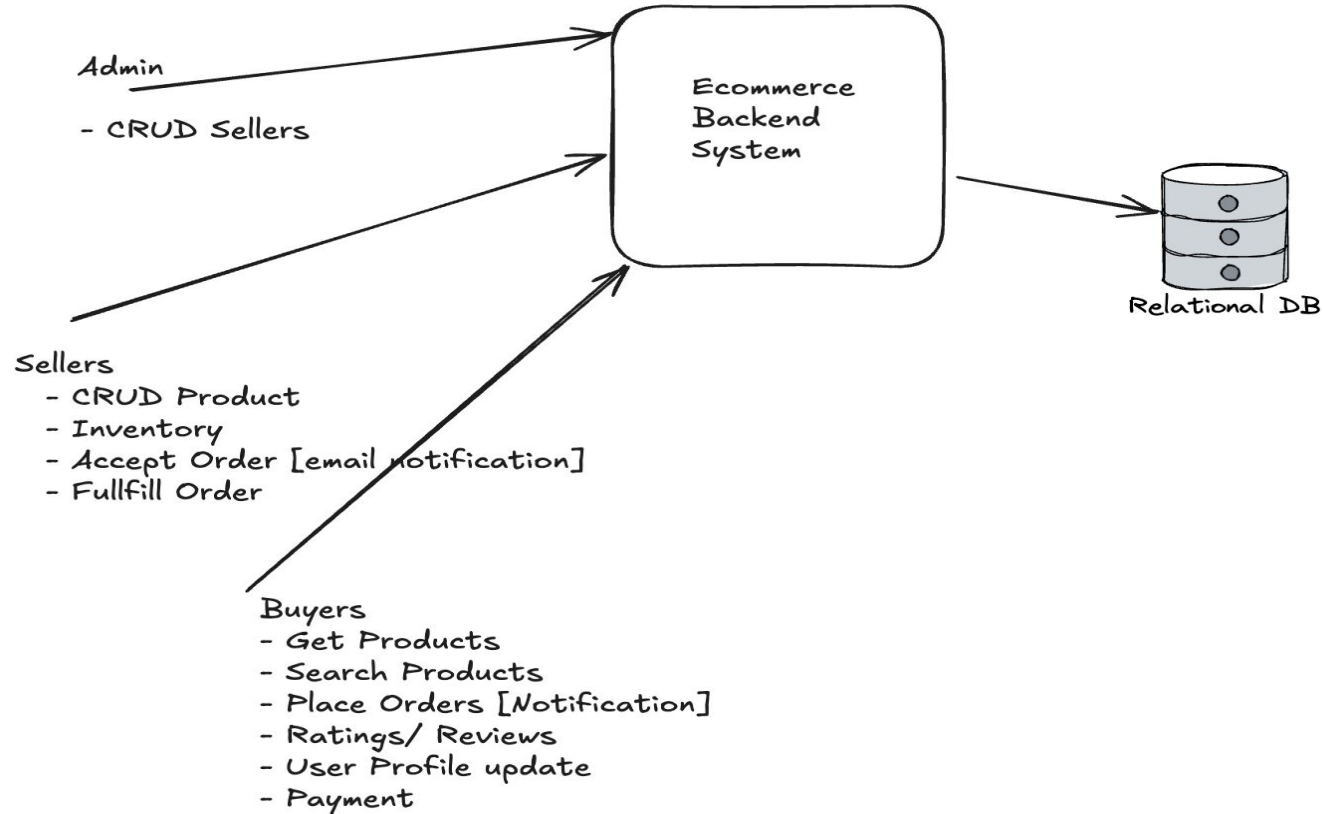
Live-85

lecture-14

Agenda

- Minor Project-1 Completion
 - Admin APIs
 - Seller APIs
 - Customer APIs
 - Pagination
 - CSV upload
 - Image upload
 - Scheduled Job
- Run application jar
- -Xms & -Xmx Flags

Ecommerce Backend System



Functional Requirements

Admin

- CRUD Sellers

Sellers

- CRUD Products
- Accept Order
- Fulfil Order

Buyers

- Search Products
- Add product to Order
- Place Order
- Order History
- Profiles
- Payments

Tables

User (id,name, email, password, role[ADMIN, SELLER, CUSTOMER], created_at, updated_at)

Company (id, name, number, is_active, user_id, created_at, updated_at)

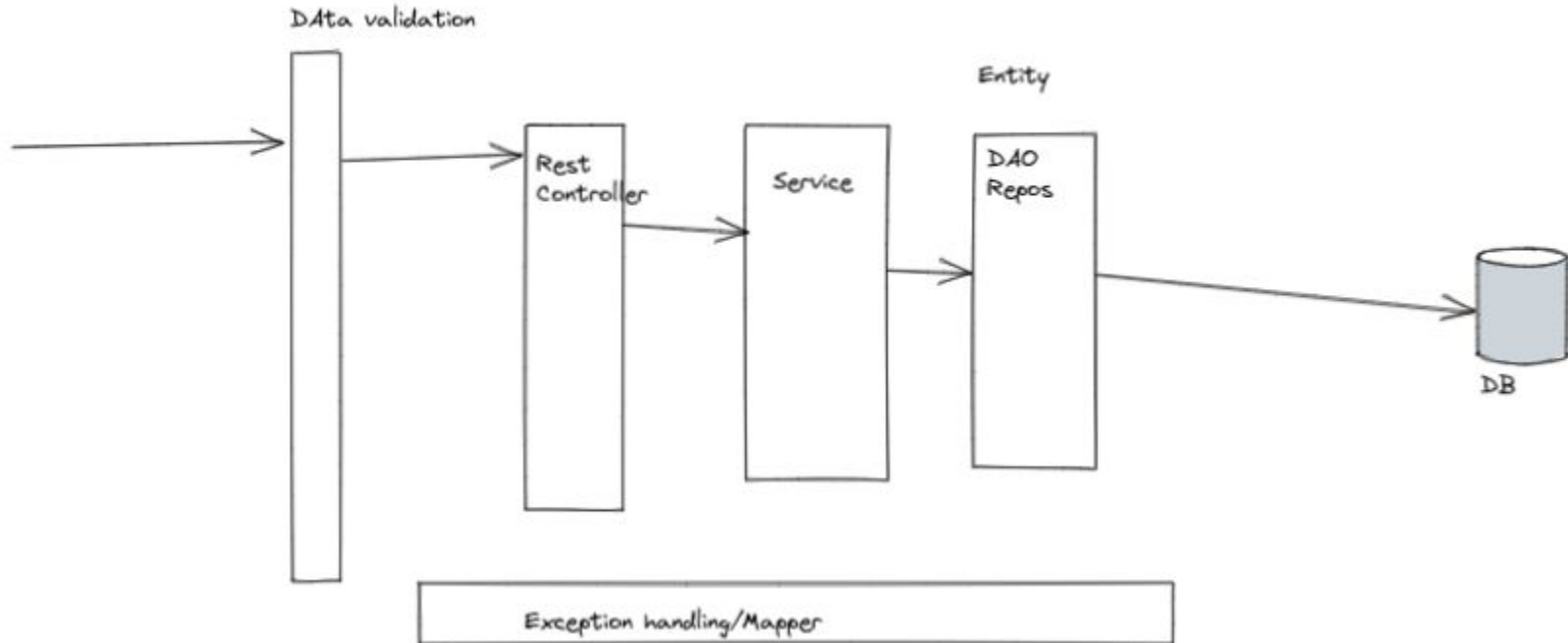
Product(id, name, description, price, stock, company_id, is_active, category_id, created_at, updated_at)

Order (id, user_id, total_amount, status(DRAFT,PLACED,ACCEPTED,SHIPPED,DELIVERED), created_at, updated_at)

Order_Item (id, order_id, product_id, quantity, price)

Category(id, name, description)

API Request flow



Admin APIs

Add a Seller

POST /api/admin/sellers

Request body: Seller details (e.g., name, email, phone)

Response: Seller ID or success message

View All Sellers

GET /api/admin/sellers

Response: List of all sellers

Delete a Seller

DELETE /api/admin/sellers/{sellerId}

Response: Success or error message

Seller APIs

Add a Product

`POST /api/seller/products`

Request body: Product details (e.g., name, description, price, stock, category)

Response: Product ID or success message

View All Products by Seller

`GET /api/seller/products`

Response: List of all products added by the seller

Update Product Details

`PUT /api/seller/products/{productId}`

Request body: Fields to update (e.g., price, stock)

Response: Success or error message

Delete a Product

`DELETE /api/seller/products/{productId}`

Response: Success or error message

Customer APIs

Browse Products

GET /api/products?keyword=laptop

Response: List of available products

View Product Details

GET /api/products/{productId}

Response: Product details

Add Order Item

POST /api/customers/order-item

Response: OrderId

View Order History

GET /api/customers/orders

Response: List of previous orders

Pagination

Pagination is often helpful when we have a large dataset and we want to present it to the user in smaller chunks.

PagingAndSortingRepository: By extending PagingAndSortingRepository repository interface can get findAll(Pageable pageable) and findAll(Sort sort) methods for paging and sorting.

```
Pageable pageable = Pageable.ofSize(pageSize).withPage(pageNo);
```