**I. Define the System**

**Business Domain:** Vinyl Record Store

**1. Define 10 Business Requirements**

**1: User Authentication**

Customers should be able to:

* Log in securely.
* Users can create new accounts.
* View and update their profile details.
* Logout

**2: Product Search and Filtering**

Customers should be able to:

* Search for vinyl records by title, artist, or genre.
* Filter the catalog by price range, release year, and stock availability.

**3: Product Catalog Management**

The app should display a catalog of vinyl records with the following details:

* Title.
* Artist.
* Genre.
* Price.
* Stock availability.
* Release year.

**4: Shopping Cart Functionality**

Users should be able to:

* Add vinyl records to their shopping cart.
* View the items in their cart, including quantities and total price.

**5: Cart**

* Only one active cart at a time
* Persist cart data across sessions until checkout or item removal.

**6: Checkout**

Users should be able to:

* Proceed to checkout from their shopping cart.
* Review their order details before finalizing.

**7: Order Management**

* Orders should include "active" or "completed." Statuses.

**8: Shipping Information**

* Provide shipping addresses during checkout.

**9: Inventory Management for Admins**

Admin users should be able to:

* Add new vinyl records to the catalog.
* Delete vinyl records no longer in stock.
* Update stock levels for existing products.

**10: User roles**

* The app should differentiate between admin and customer accounts, granting admin users exclusive access to inventory management tools.

**2. MVP Features**

**For the MVP (Minimum Viable Product) phase, the app should contain the following five features**:

1. **User Authentication**: Allow users to register, log in, and manage their profiles.
2. **Catalog of Vinyl Records**: Display a catalog of vinyl records with their details.
3. **Shopping Cart**: Allow users to add items to their cart and view their selections.
4. **Order Placement**: Let users place orders and view their order history.
5. **Admin Interface**: Add admin capabilities for adding, updating and deleting vinyl records.

**II. Spring Boot Application Structure**

**5. Entities**

1. **User**: Represents a customer with properties like ID, name, email, password, and role.
2. **Vinyl**: Represents a vinyl record with properties like ID, title, artist, genre, price, stock.
3. **Order**: Represents a customer’s order with properties like order ID, user, order status, and list of items.
4. **OrderItem**: Represents an item in the order, linking to a specific vinyl record and quantity.
5. **ShippingAddress**: Stores shipping information for the user.
6. **Review**: Represents a review left by the user for a vinyl record, with properties like rating and comment.

At least 4 relationships:

1. **User-Order**: One-to-many (A user can have many orders).
2. **Order-Vinyl**: Many-to-many (An order can contain multiple vinyl records, and a vinyl record can appear in multiple orders).
3. **ShippingAddress-User:** Many-to-One
4. **OrderItem-Order**: Many-to-One

**8.Used Thymeleaf**

http://localhost:9093/swagger-ui/index.html#/