SEPHORA

https://www.sephora.me/sa-ar

Made by Asrar Fallatah

What is Sephora ?

Sephora is one of the leading global beauty company retailers. It is recognized for revolutionizing the way people shop for cosmetics, skincare, haircare, and fragrances. It was founded in France in 1969 and later acquired by the LVMH company. Sephora has now grown to be one of the most influential brands in the beauty industry. What makes Sephora unique is its "open-sell" concept, which allows their customers to freely test, compare, and explore products from a wide range of famous international brands, along with Sephora's own private label. Sephora now has thousands of stores around the world and a strong online shop website with secure payment, which has enabled them to set trends, introduce emerging beauty brands, and provide a personalized shopping experience through trained beauty experts and in-store services. Today, Sephora is known not just as a store, but as a beauty destination that inspires confidence, creativity, and self-expression.

Why Sephora ?

I chose Sephora as the domain for my validation checklist because it represents a real environment in the world that I can relate to and where data accuracy, inventory validation, and customer experience are critical. Sephora deals with thousands of products from different brands each has its their own different shades, ingredients, expiration dates, and pricing. This makes proper validation essential whether for product data, stock availability, online orders, or customer information.

Additionally Sephoras strong reputation for quality and consistency aligns perfectly with the concept of validation. Even a small data error like incorrect product size, wrong prices, or missing ingredient details can lead to customer dissatisfaction, financial loss, or safety issues.

By choosing Sephora, I can create a validation checklist that demonstrates:

1. How structured data validation improves customer trust
2. How proper checks ensure product accuracy and prevent mistakes
3. How validation directly affects the user experience in both physical stores and online

Sephora's fast-paced retail environment makes it an ideal real-life example to apply validation rules in a meaningful context.

After carefully reviewing and studying SEPHORA's website I believe SEPHORA's developer team may used logic like this :

- **MODEL 1 : Customer**

CustomerClass {

@notNull(messege = "email can not be embpty !")

// is used to make sure mandatory fields are not left empty, because Sephora cannot create an account or process an order without this information.

@Email(messege = " you must enter a valid email !")

//is to ensures that the customer enters a correctly formatted email so that Sephora can send order confirmations, password resets, and promotional emails.

private String customerEmail ;

@notNull(messege = "social status can not be embpty !")

// is used to make sure mandatory fields are not left empty, because Sephora cannot create an account or process an order without this information.

@Pattern(regexp = "Ms | Mrs | Mr", message = "please enter your social status correctly ")

// this guarantees that only spesific values are accepted, which keeps the data consistent across the system, especially when printing invoices or sending messages.

private String customerScoialStatus;

@notNull(messege = "First name can not be embpty !")

// is used to make sure mandatory fields are not left empty, because Sephora cannot create an account or process an order without this information.

@Size(min = 1 , message = "first name can not be less than 1 character ")

// ensures that customers do not enter invalid short names which could affect delivery or account lookups.

private String customerFirstName;

@notNull(messege = “last name can not be embpty !”)

// is used to make sure mandatory fields are not left empty, because Sephora cannot create an account or process an order without this information.

@Size(min = 1 , message = "last name can not be less than 1 character ")

// ensures that customers do not enter invalid short names which could affect delivery or account lookups.

private String customerlastName;

@notNull(messege = “phone number can not be embpty !”)

// is used to make sure mandatory fields are not left empty, because Sephora cannot create an account or process an order without this information.

@Size(min = 10 , max = 10, message = " Phone number must contain 10 digits numbers only ! ")

private String customerPhoneNumber;

@notNull(messege = “Password can not be embpty !”)

// is used to make sure mandatory fields are not left empty, because Sephora cannot create an account or process an order without this information.

@Pattern(regexp = "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#$%^&+=])(?=\\S+$).{8,}$",

message = "Password must be at least 8 characters, contain at least 1 uppercase letter, 1 lowercase letter, 1 digit, and 1 special character")

// to make sure that strong password rules are maintained(uppercase, lowercase, digits, and special characters) to protect users’ accounts .

private String customerPassword;

@Pattern(regexp = "true", message = " user must be 18 and above ! ")

// it makes sure the user meets the age requirement, since Sephora sells products not suitable for minors, and different legal regulations apply depending on age.

private boolean isEighteenandAbove ;

}

- **MODEL 2 : Brand**

BrandClass {

@NotNull(message = "Brand name can not be empty!")

@Size(min = 2, max = 50, message = "Brand name must be between 2 and 50 characters")

private String brandName;

@NotNull(message = "Brand category can not be empty!")

@Pattern(regexp = "Makeup|Skincare|Haircare|Fragrance|Tools",

message = "Brand category must be one of the following: Makeup, Skincare, Haircare, Fragrance, Tools"
)

private String brandCategory;

@NotNull(message = "Country of origin can not be empty!")

@Size(min = 2, message = "Country name must be at least 2 characters")

private String countryOfOrigin;

@NotNull(message = "Status can not be empty!")

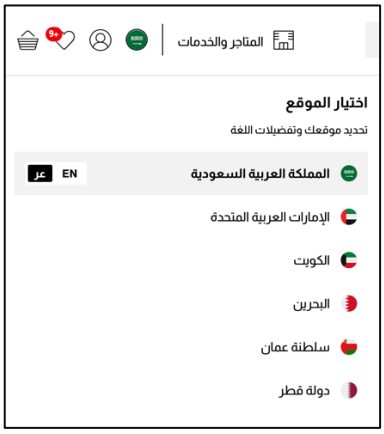
@Pattern(regexp = "Active|Inactive",

message = "Brand status must be either Active or Inactive")

private String brandStatus;

}

- **MODEL 3 : Country**



CountryClass {

@NotNull(message = "Country name cannot be empty!")

@Size(min = 2, message = "Country name must be at least 2 characters")

private String countryName;

@NotNull(message = "Country code cannot be empty!")

@Pattern(regexp = "^[A-Z]{2,3}$",

message = "Country code must follow the ISO standard (Example: SA, US, FR)")

private String countryCode;

@NotNull(message = "Currency cannot be empty!")

@Pattern(regexp = "USD|SAR|EUR|GBP|AED",

message = "Currency must be a valid code such as USD, SAR, EUR, GBP, or AED")

private String currency;

@NotNull(message = "Shipping availability cannot be empty!")

@Pattern(regexp = "Available|Not Available",

message = "Shipping must be either: Available or Not Available")

private String shippingAvailability; }