Q2 -I will use a restaurant website for the example, the restaurant will have a lot of models but the main ones that I see are (Customer, Order, Restaurant,Product) I will use the KFC website for this.

A screenshot of a food advertisement

Description automatically generated

The first model is Restaurant:

public class Restaurant {

@NotEmpty(message = "Restaurant ID should not be empty")

@Size(min = 3, max = 20, message = "Restaurant ID must be between 3 and 20 characters")

private String restaurantId;

@NotEmpty(message = "Name cannot be empty")

@Size(min = 3, max = 50, message = "Name length must be between 3 and 50 characters")

private String name;

@NotEmpty(message = "Address cannot be empty")

@Size(min = 10, max = 200, message = "Address length must be between 10 and 200 characters")

private String address;

@Pattern(regexp = "^05\\d{8}$", message = "Phone number must start with '05' and contain exactly 10 digits.")

private String phoneNumber;

@NotNull(message = "Opening date cannot be null")

@Past(message = "Opening date must be in the past")

private LocalDate openingDate;

@Min(value = 0, message = "Number of employees must be a non-negative number")

@Max(value = 1000, message = "Number of employees cannot exceed 1000")

private int numberOfEmployees;

@Size(max = 1000, message = "Additional information length must be up to 1000 characters")

private String additionalInformation;

}

How it will lock :

@NotEmpty(message = "Restaurant ID should not be empty")  
@Size(min = 3, max = 20, message = "Restaurant ID must be between 3 and 20 characters")  
private String restaurantId;  
  
@NotEmpty(message = "Name cannot be empty")  
@Size(min = 3, max = 50, message = "Name length must be between 3 and 50 characters")  
private String name;  
  
@NotEmpty(message = "Address cannot be empty")  
@Size(min = 10, max = 200, message = "Address length must be between 10 and 200 characters")  
private String address;  
  
@Pattern(regexp = "^05\\d{8}$", message = "Phone number must start with '05' and contain exactly 10 digits.")  
private String phoneNumber;  
  
@NotNull(message = "Opening date cannot be null")  
@Past(message = "Opening date must be in the past")  
private LocalDate openingDate;  
  
@Min(value = 0, message = "Number of employees must be a non-negative number")  
@Max(value = 1000, message = "Number of employees cannot exceed 1000")  
private int numberOfEmployees;  
  
@Size(max = 1000, message = "Additional information length must be up to 1000 characters")  
private String additionalInformation;

Customer model :

@NotEmpty(message = "Customer ID should not be empty")

@Size(min = 3, max = 20, message = "Customer ID must be between 3 and 20 characters")

private String customerId;

@NotEmpty(message = "Name cannot be empty")

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

private String name;

@NotEmpty(message = "Email should not be empty")

@Email(message = "Invalid email format")

private String email;

@Pattern(regexp = "^05\\d{8}$", message = "Phone number must start with '05' and contain exactly 10 digits.")

private String phoneNumber;

@NotNull(message = "Date of birth cannot be null")

@Past(message = "Date of birth must be in the past")

private LocalDate dateOfBirth;

@Positive(message = "must be should be a positive number ")

@Max(value = 10000, message = "Loyalty points cannot exceed 10000")

private int loyaltyPoints;

}

How it will lock :

@NotEmpty(message = "Customer ID should not be empty")  
@Size(min = 3, max = 20, message = "Customer ID must be between 3 and 20 characters")  
private String customerId;  
  
@NotEmpty(message = "Name cannot be empty")  
@Size(min = 2, max = 50, message = "Name length must be between 2 and 50 characters")  
private String name;  
  
@NotEmpty(message = "Email should not be empty")  
@Email(message = "Invalid email format")  
private String email;  
  
@Pattern(regexp = "^05\\d{8}$", message = "Phone number must start with '05' and contain exactly 10 digits.")  
private String phoneNumber;  
  
@NotNull(message = "Date of birth cannot be null")  
@Past(message = "Date of birth must be in the past")  
private LocalDate dateOfBirth;  
  
@Min(value = 0, message = "Loyalty points must be a non-negative number")  
@Max(value = 10000, message = "Loyalty points cannot exceed 10000")  
private int loyaltyPoints;

Product model :

public class Product {

@NotEmpty(message = "Product ID should not be empty")

@Size(min = 3, max = 20, message = "Product ID must be between 3 and 20 characters")

private String productId;

@NotEmpty(message = "Product name cannot be empty")

@Size(min = 3, max = 100, message = "Product name length must be between 3 and 100 characters")

private String productName;

@DecimalMin(value = "0.0", inclusive = false, message = "Price must be a positive number")

@Digits(integer = 5, fraction = 2, message = "Price must have up to 5 integer digits and 2 decimal places")

private double price;

@NotEmpty(message = "Category cannot be empty")

@Pattern(regexp = "burger|pizza|drink|side", message = "Category must be 'burger', 'pizza', 'drink', or 'side'")

private String category;

@Positive(message = "Stock must be a non-negative number")

private int stock;

@Size(max = 500, message = "Description length must be up to 500 characters")

private String description;

}

How it will lock :

@NotEmpty(message = "Product ID should not be empty")  
@Size(min = 3, max = 20, message = "Product ID must be between 3 and 20 characters")  
private String productId;  
  
@NotEmpty(message = "Product name cannot be empty")  
@Size(min = 3, max = 100, message = "Product name length must be between 3 and 100 characters")  
private String productName;  
  
@DecimalMin(value = "0.0", inclusive = false, message = "Price must be a positive number")  
@Digits(integer = 5, fraction = 2, message = "Price must have up to 5 integer digits and 2 decimal places")  
private double price;  
  
@NotEmpty(message = "Category cannot be empty")  
@Pattern(regexp = "burger|pizza|drink|side", message = "Category must be 'burger', 'pizza', 'drink', or 'side'")  
private String category;

@Positve (message = "Stock must be a non-negative number”)  
private int stock;  
  
@Size(max = 500, message = "Description length must be up to 500 characters")  
private String description;

Model Order :

public class Order {

@NotEmpty(message = "Order ID should not be empty")

@Size(min = 5, max = 20, message = "Order ID must be between 5 and 20 characters")

private String orderId;

@NotEmpty(message = "Customer ID cannot be empty")

@Size(min = 3, max = 20, message = "Customer ID must be between 3 and 20 characters")

private String customerId;

@NotNull(message = "Order date cannot be null")

private LocalDateTime orderDate;

@Positive(value = "0.0", inclusive = false, message = "Total amount must be a positive number")

private double totalAmount;

@Pattern(regexp = "pending|completed|cancelled", message = "Status must be 'pending', 'completed', or 'cancelled'")

private String status;

@Min(value = 1, message = "Order priority must be at least 1")

@Max(value = 5, message = "Order priority cannot exceed 5")

private int priority;

@NotEmpty(message = "Order type cannot be empty")

@Pattern(regexp = "PICKUP|DRIVE\_THROUGH", message = "Order type must be 'PICKUP' or 'DRIVE\_THROUGH'")

private String orderType;

}

How it will lock

public class Order {  
 @NotEmpty(message = "Order ID should not be empty")  
 @Size(min = 5, max = 20, message = "Order ID must be between 5 and 20 characters")

private String orderId;  
 @NotEmpty(message = "Customer ID cannot be empty")  
 @Size(min = 3, max = 20, message = "Customer ID must be between 3 and 20 characters")

private String customerId;  
 @NotNull(message = "Order date cannot be null")

private LocalDateTime orderDate;  
 @Positive(value = "0.0", inclusive = false, message = "Total amount must be a positive number")  
 private double totalAmount;

@Pattern(regexp = "pending|completed|cancelled", message = "Status must be 'pending', 'completed', or 'cancelled'")

private String status;

@Min(value = 1, message = "Order priority must be at least 1")  
 @Max(value = 5, message = "Order priority cannot exceed 5")

private int priority;

@NotEmpty(message = "Order type cannot be empty")  
 @Pattern(regexp = "PICKUP|DRIVE\_THROUGH", message = "Order type must be 'PICKUP' or 'DRIVE\_THROUGH'")  
 private String orderType;  
}