Price Calculator

1.0 Introduction

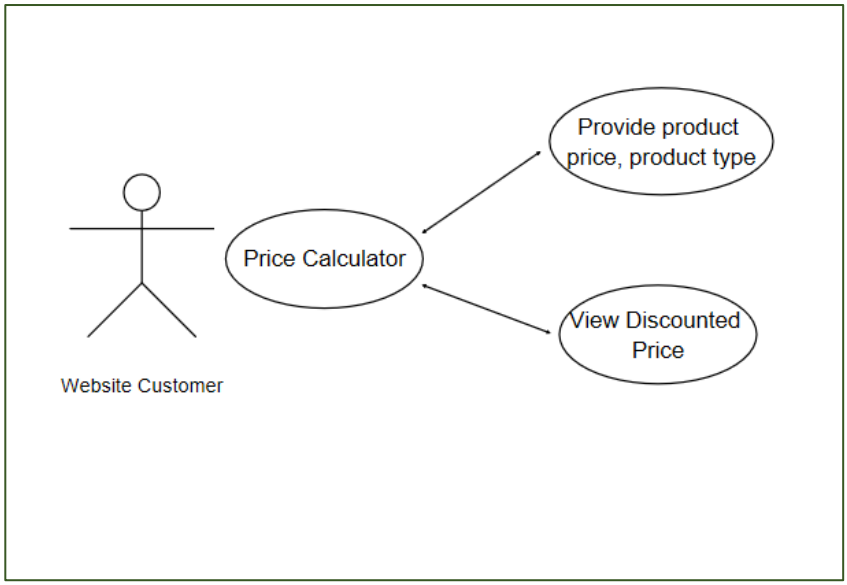
1.1 Functional Requirements

An e-commerce based retail application provides a wide range of products to its customers. The website offers discount on various products.

Price Calculator is an application used by the website customers to calculate the discounted price for any given product.

To automate this calculation process, develop a Spring boot, Spring MVC application. This application needs to get product price, product type and display the discounted price as output.

1.2. Use Case Diagram



2.0 Technical Specifications

2.1 Process Flow

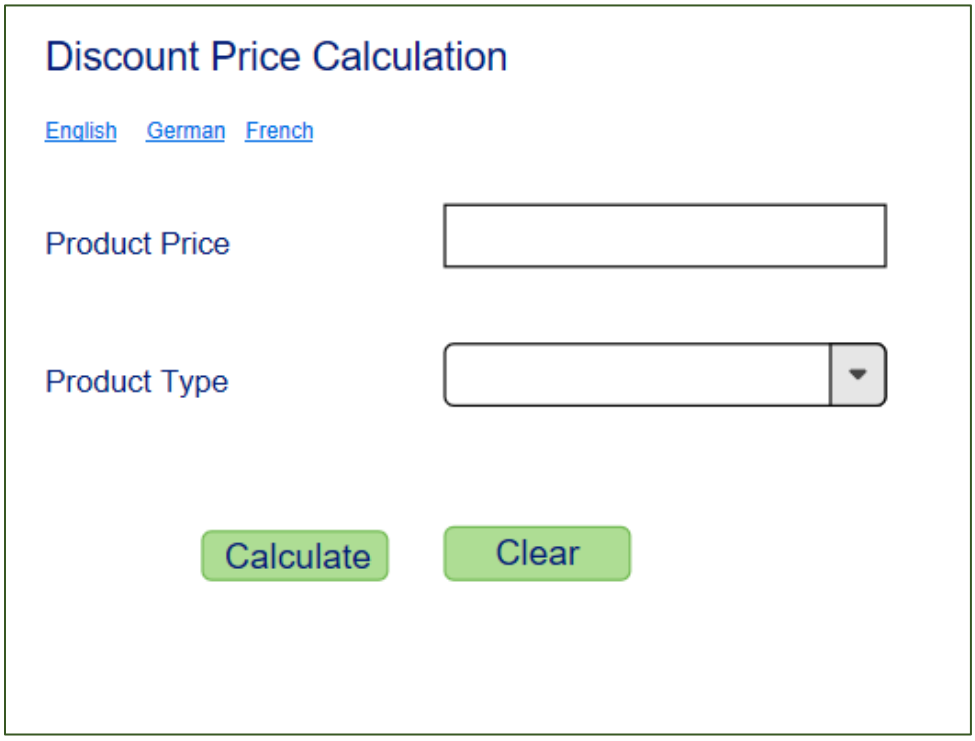
- The first page of the application is calculatediscount.jsp page, this page should have the below form components.

Component	Component Type	Component ID
Product Price	Textbox	productPrice
Product Type	Drop down	productType
Calculate Discount Percentage	Submit	calculateDiscountPrice

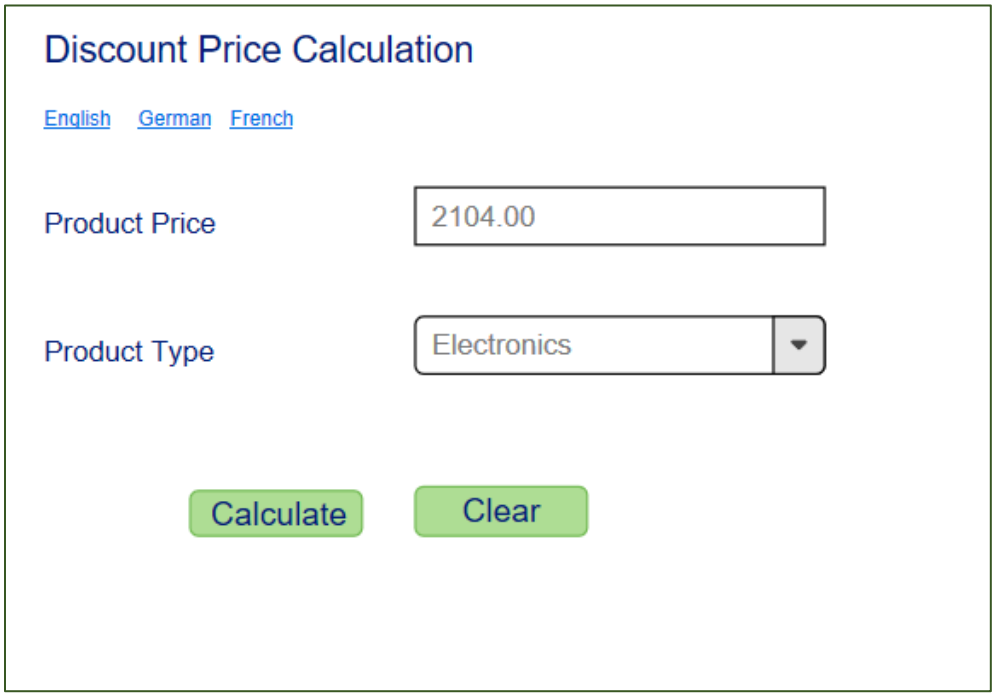
- calculatediscount.jsp should be internationalized. The user is given the options to choose the language (English, German and French) in which the website is to be displayed.
- By default, the page content will be displayed in English. For example, if the user chooses the language as French the label content and the error message in the page should displayed in French. Ensure you implement the concept of Internationalization.
- In the calculatediscount.jsp page, ensure you use the component type and id correctly as specified in the above table.
- The values in the Product type drop-down must be auto populated from the controller with the values as given in the below table. The values should not be populated/hardcoded inside the JSP.

Product Type
Electronics
Toys
Apparels

- Sample calculatediscount.jsp for displaying the page content in English

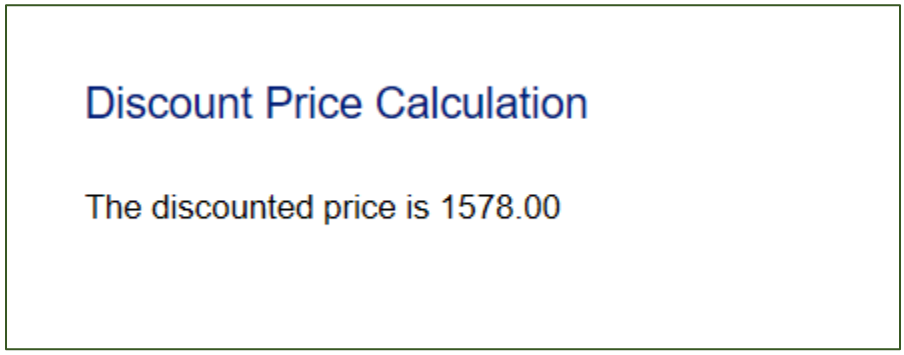


- Key in all the required inputs and click on the Calculate button.



- Validation needs to be performed using **JSR303 Hibernate Validation**- @Valid annotation.
- Business Validation rules are listed in Section2.2

- On Successful validation, discounted price must be calculated and displayed in the finalprice.jsp file.

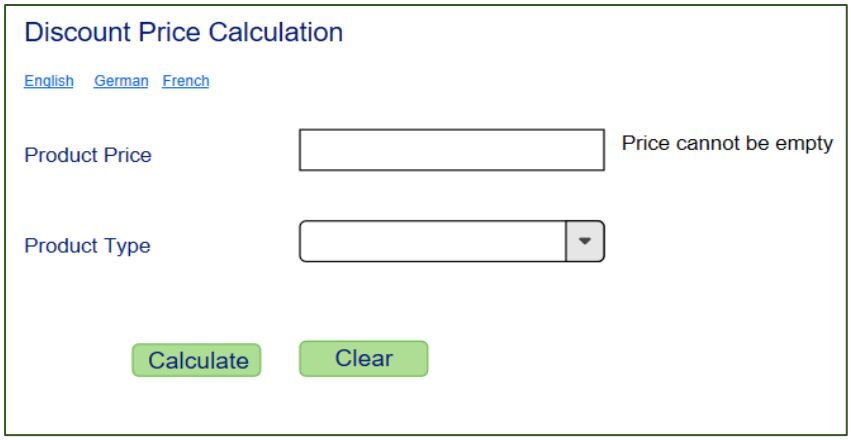


- The message should be displayed, as “The discounted price is <<discountedAmount>>”.
- The result should be rendered inside the <h2> tag. Refer 2.4 for the logic to be followed for doing calculation.

2.2 Business Validation

- Rule: Product Price field is mandatory.
    - Message: If validation fails, UI should display the error message “Price cannot be empty”. This message should be displayed in the language chosen by the user.
- Rule: Product Price should be numeric.
    - Message: If the validation fails, UI should display the error message - “Product Price should be numeric”
- **Note:**

    - The error messages have to be retrieved from the appropriate properties file.
    - Include the required Spring tag for Internationalization to display the label and the error messages.



2.3 Control Flow

- By giving the request /getDiscountedPrice from the browser (Eg: http://localhost/8080/ getDiscountedPrice), the user should be redirected to the calculatediscount.jsp
- User will enter the product price and product type.
- On clicking calculate button validations should be performed and the discounted product price should be displayed in finalprice.jsp.

2.4 Business Rules

- Discount percentage for the different product are given below.

Product Type	Discount Percentage
Electronic	25
Apparels	10
Toys	50

- Discounted Price = Product Price - (Product Price *(Discount percentage /100))

3.0 Component Specification

3.0.1 Controller

- The DiscountController should be configured via annotation as a Controller

DiscountController			
Attribute Name	**Attribute Type**	**Access Specifier**	**Constraints**
discountService	DiscountServiceInterface	private	Use annotation to autowire

Method Name	Method Argument name:type	Return type	RequestMapping URL & Request Method
discountPage	modelAttribute “product”:Product	String	/getDiscountedPrice & GET
calculateDiscount	modelAttribute “product”: Product, map:ModelMap, result:BindingResult	String	/calculateDiscountedPrice & GET
populateProductType		List<String>	Should be annotated with ModelAttribute with name “productTypeList”

- Inside the populateProductType method, you should add all the 3 product types into the list and return the list.
- DiscountService interface should be autowired inside the Controller via annotations
- DiscountServiceImpl must calculate the discounted price.

3.0.2 Service

- DiscountService interface should be configured via annotation as Service

Method Name	Method Argument name:type	Return type	Responsibilities
calculateDiscount	“product”: Product	double	This method should calculate the discounted price based on the product price and type.

- DiscountServiceImpl must implement DiscountService interface.

3.0.3 Model

Class(Model)	Property	Methods
Product	productPrice : double productType : String	//getters and setters

- Include the needed annotations inside the Product class for various validations already mentioned. The error messages should be retrieved from the properties file

3.1 Request Mapping Tasks

/getDiscountedPrice	Redirect the user to calculatediscount.jsp
/calculateDiscountedPrice	• Invoke calculateDiscount method of the Controller, which in turn should invokes the calculateDiscount method of the DiscountServiceImpl • Redirect to finalprice.jsp with the appropriate values depicted in the screen design. • In case of validation errors the user should be redirected to calculatediscount.jsp page

4.0 Overall Design Constraints

1) Implement the code using the best design standards.
2) Use Internationalization for all the labels and messages.
3) Validation needs to be performed using **JSR303 Hibernate Validation**- @Valid annotation.
4) Adhere to the design specifications mentioned in the case study.