

1. Design for class “ProductListScreen”

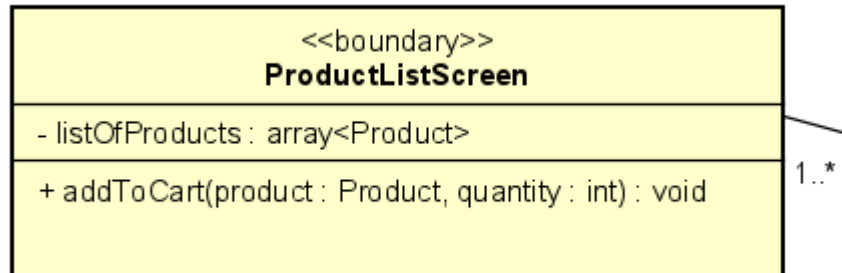


Table 1. Attribute design

#	Name	Data type	Default value	Description
1	listOfProducts	Array<Product>	empty array []	A list of products currently in the cart. Each element is a Product object.

Table 2: Operation design

#	Name	Return type	Description (purpose)
1	addToCart	Void	Adds a product to the shopping cart.

Method: addToCart

- **Parameters:**
 - product: Product → The product to be added to the cart.
 - quantity: int → The number of units of the product to be added.
- **Exceptions:**
 - Throws InvalidProductException if the product is unavailable or the quantity is invalid.
- **How to use parameters/attributes (Algorithm to implement operation):**

1. Call CartController.addToCart(product, quantity).
2. Handle any exceptions (product unavailable).

2. Design for class “ProductDetailScreen”

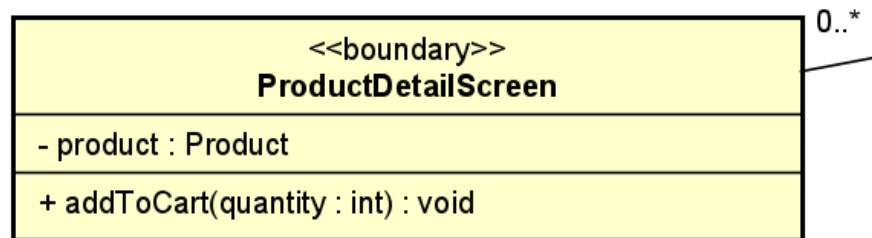


Table 1. Attribute design

#	Name	Data type	Default value	Description
1	product	Product	null	Stores the product details being displayed.

Table 2: Operation design

#	Name	Return type	Description (purpose)
1	addToCart	Void	Adds a product to the shopping cart.

Method: addToCart

- **Parameters:**
 - quantity: int → The number of units of the product.
- **Exceptions:**
 - Throws InvalidProductException if the product is unavailable or the quantity is invalid.
- **Algorithm:**

1. Calls CartController.addToCart(self.product, quantity).
2. Handles exceptions related to stock availability.

3. Design for class “CartController”

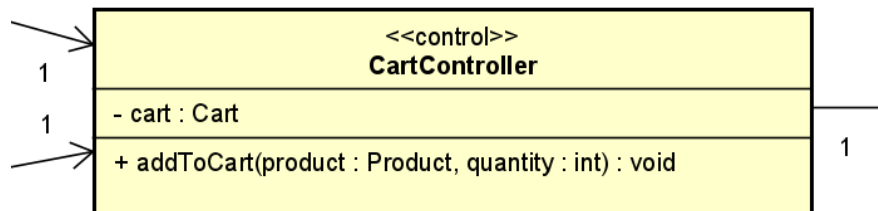


Table 1. Attribute design

#	Name	Data type	Default value	Description
1	cart	Cart	null	Reference to the shopping cart.

Table 2: Operation design

#	Name	Return type	Description (purpose)
1	addToCart	Void	Adds a product to the shopping cart.

Method: addToCart

- **Parameters:**
 - product: Product → The product to be added.
 - quantity: int → The number of units.
- **Exceptions:**
 - Throws InvalidProductException if the product is unavailable or the quantity is invalid.
- **Algorithm:**

1. Calls Cart.addToCart(product, quantity).
2. Handles exceptions related to stock availability.

4. Design for class “Cart”

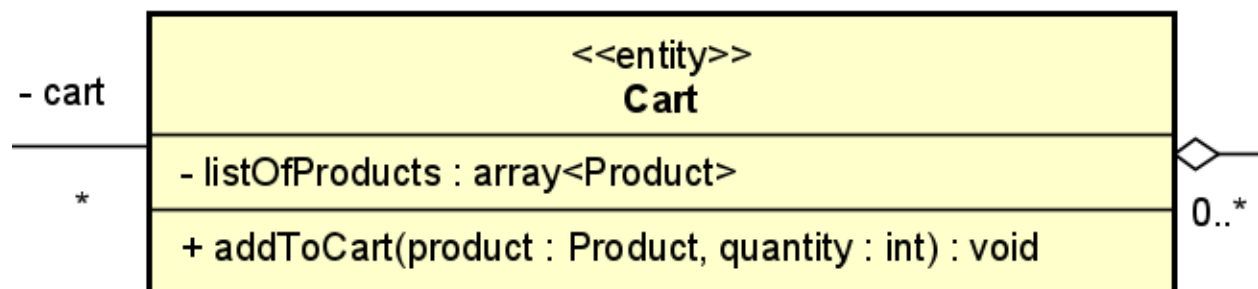


Table 1. Attribute design

#	Name	Data type	Default value	Description
1	listOfProducts	array<Product>	[]	Stores the products added to the cart.

Table 2: Operation design

#	Name	Return type	Description (purpose)
1	addToCart	Void	Adds a product to the shopping cart.

Method: addToCart

- **Parameters:**
 - product: Product → The product to be added.
 - quantity: int → The number of units.

- **Exceptions:**

- Throws InvalidProductException if the product is unavailable or the quantity is invalid.

- **Algorithm:**

1. Calls product.checkProductAvailability(quantity).
2. If available, adds the product to listOfProducts.
3. If not available, throws an InvalidProductException.

5. Design for class “Cart”

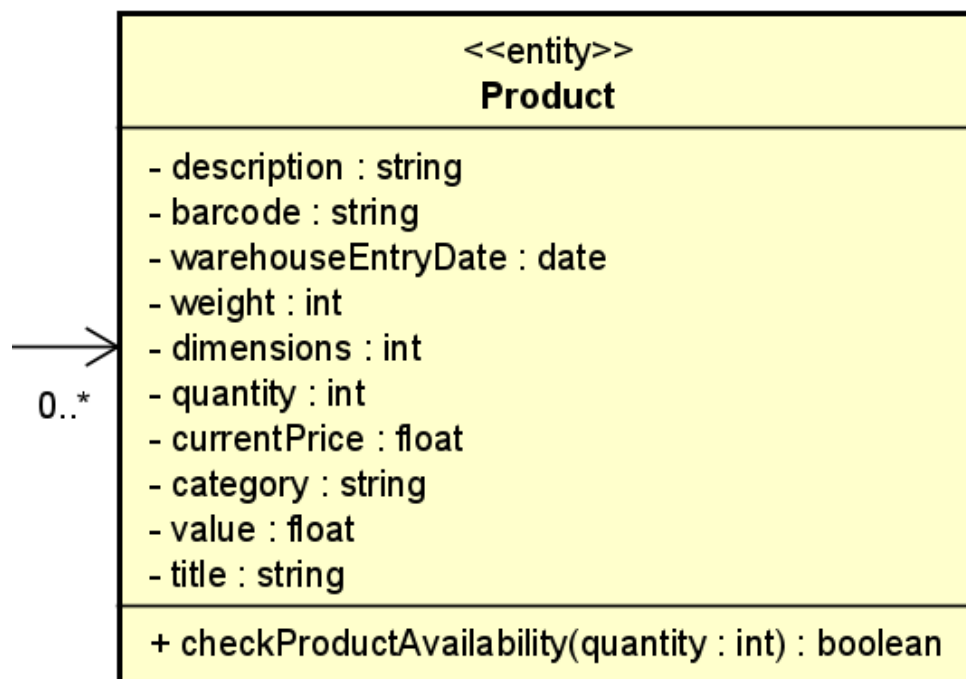


Table 1. Attribute design

#	Name	Data type	Default value	Description
1	description	string	""	Description of the product.

2	Barcode	String	""	Unique identifier of the product
3.	warehouseEntryDate	Date	Null	Date when the product entered the warehouse
4	Weight	Int	0	Product weight
5	Dimensions	Int	0	Available quantity
6	Quantity	Int	0	Available quantity
7	currentPrice	Float	0.0	Price per unit
8	Category	String	""	Product category
9	Value	Float	0.0	Product value
10	Title	String	""	Product title

Table 2: Operation design

#	Name	Return type	Description (purpose)
1	checkProductAvailability	Boolean	Checks if the product is available in the given quantity

Method: checkProductAvailability

- **Parameters:**

- quantity: int → The number of units required.

- **Exceptions:**

- None

- **Algorithm:**

1. If self.quantity >= quantity:
 - Return true.
2. Else:

- Return false.