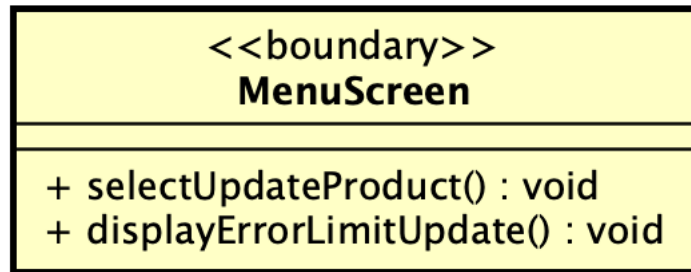


UpdateProductToStore

1. Design for class “MenuScreen”



1.1. Attribute Design:

None

1.2. Operation Design:

STT	Name	Return Type	Description (Purpose)
1	selectUpdateProduct()	void	Triggers the process of updating an existing product
2	displayErrorLimitUpdate()	void	Displays an error message when update limit is exceeded

- **selectUpdateProduct Method**

- *Parameter:* None

- *Exception:*

- NavigationException if the system fails to navigate to the product update screen.

- *How to use parameters/attributes:*

This method is executed when the user selects the option to update a product from the menu. It interacts with the ProductController to start the product update process.

- **displayErrorLimitUpdate Method**

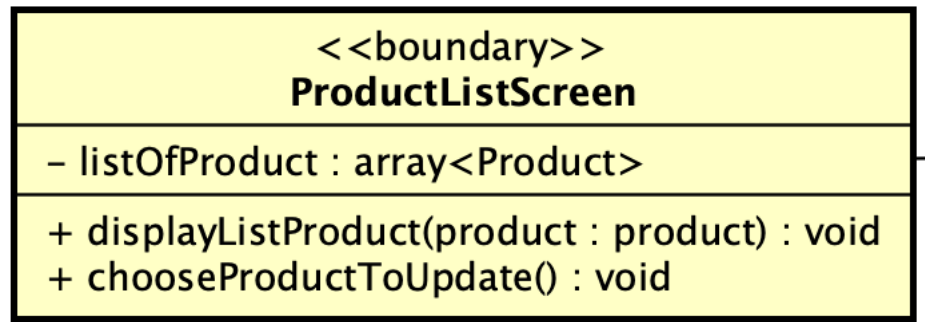
- *Parameter:* None

- *Exception:* None

- *How to use parameters/attributes:*

This method is called when a user exceeds the allowed number of product updates. It displays an appropriate error message to notify the user.

2. Design for class “ProductListScreen”



2.1. Attribute Design:

STT	Name	Data type	Default value	Description
1	listOfProduct	array<Product>		List of products

2.2. Operation Design

STT	Name	Return type	Description (purpose)
1	displayListProduct()	void	Displays the list of available products
2	chooseProductToUpdate()	void	Allows the user to select a product for updating

▪ **displayListProduct Method**

- *Parameter:*

- product: Product - The product to be displayed in the list.

- *Exception: None*

- *How to use parameters/attributes:*

This method is responsible for displaying the list of available products in the system. It presents information about each product to the user.

▪ **chooseProductToUpdate Method**

- *Parameter:None*

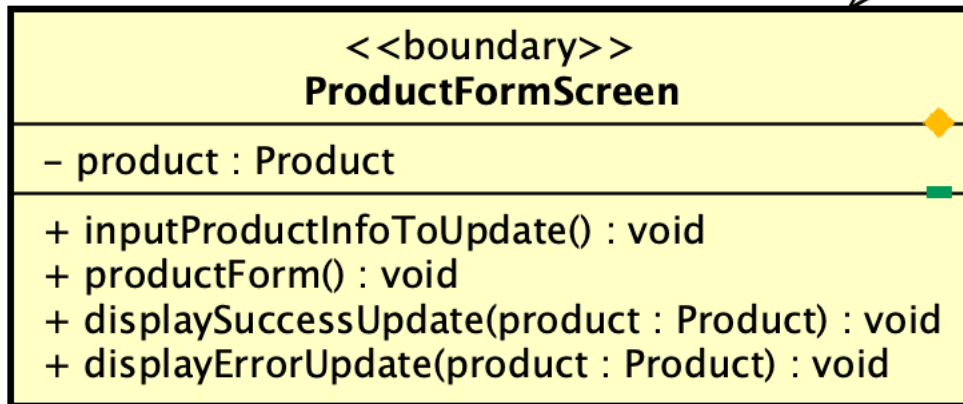
- *Exception:*

- SelectionException if the user selects an invalid product.

- *How to use parameters/attributes:*

This method enables the user to choose a product from the list for updating. It interacts with ProductController to process the update request.

3. Design for class “ProductFormScreen”



3.1. Attribute Design:

STT	Name	Data type	Default Value	Description
1	product	Product	Null	Represents the product being processed in the form

3.2. Operation Design

STT	Name	Return type	Description (purpose)
1	inputProductInfoToUpdate()	Void	Collects product information for updating
2	productForm()	Void	Displays the product form for input
3	displaySuccessUpdate(product)	Void	Displays a success message for a specific product
4	displayErrorUpdate(product)	void	Displays an error message for a specific product

▪ **inputProductInfoToUpdate Method**

- *Parameter:* None
- *Exception:*
- InputException if input data is invalid.
- *How to use parameters/attributes:*

This method gathers user input regarding product updates and processes it before submission.

- **productForm Method**
- *Parameter:* None
- *Exception:* None
- *How to use parameters/attributes:*

This method initializes and displays the product form for user interaction.

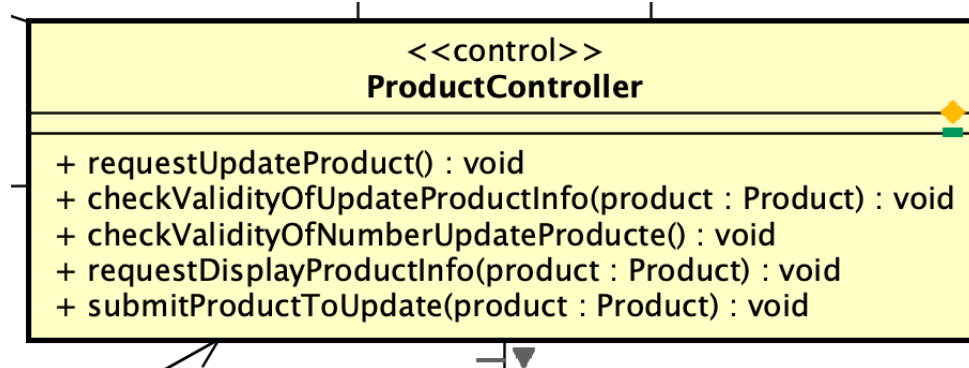
- **displaySuccessUpdate Method**
- *Parameter:*
- Product
- *Exception:*None
- *How to use parameters/attributes:*

This method provides feedback to the user when an update operation is successful.

- **displayErrorUpdate Method**
- *Parameter:*
- Product
- *Exception:*
- DisplayException if the error message cannot be shown.
- *How to use parameters/attributes:*

This method notifies the user in case of an error during the product update process.

4. Design for class “ProductController”



4.1. Attribute Design:

STT	Name	Data type	Default Value	Description
1	product	Product	Null	Represents the product being processed in the form

4.2. Operation Design

STT	Name	Return type	Description (purpose)
1	requestUpdateProduct()	Void	Initiates the product update process
2	checkValidityOfUpdateProductInfo(product)	Void	Validates product update information
3	checkValidityOfNumberUpdateProduct()	Void	Checks if the number of updates allowed is exceeded
4	requestDisplayProductInfo(product)	Void	Requests display of product information
5	submitProductToUpdate(product)	void	Submits the updated product for processing

▪ requestUpdateProduct Method

- *Parameter:* None
- *Exception:*

- UpdateException if the update process cannot be initiated.

- *How to use parameters/attributes:*

This method initiates the product update process by verifying prerequisites.

- **checkValidityOfUpdateProductInfo Method**

- *Parameter:*

- product: Product - The product to validate.

- *Exception:*

- ValidationException if product details are invalid.

- *How to use parameters/attributes:*

This method ensures that the product update information follows required constraints.

- **checkValidityOfNumberUpdateProduct Method**

- *Parameter:None*

- *Exception:*

- LimitExceededException if the allowed update limit is exceeded.

- *How to use parameters/attributes:*

This method verifies if further updates are permitted based on current constraints.

- **requestDisplayProductInfo Method**

- *Parameter:*

- product: Product - The product whose information is requested.

- *Exception:None*

- *How to use parameters/attributes:*

This method retrieves and presents details of a specified product.

- **submitProductToUpdate Method**

- *Parameter:*

- product: Product - The product to update.

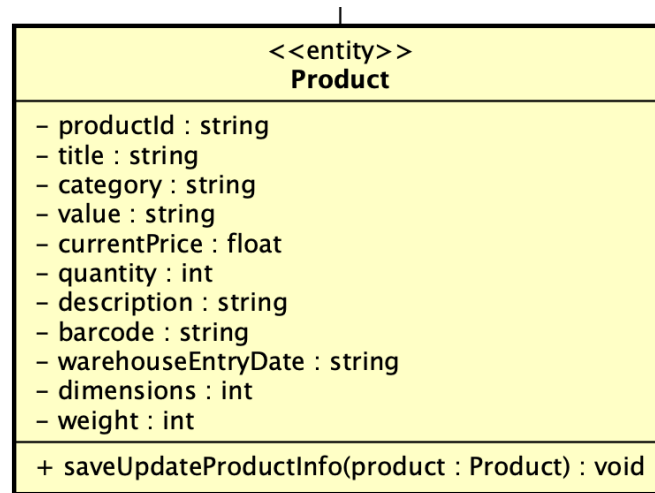
- *Exception:*

- UpdateException if the submission process fails.

- *How to use parameters/attributes:*

This method finalizes and submits the updated product for processing.

5. Design for class “Product”



5.1. Attribute design

STT	Name	Date type	Default value	Description
1	title	string	None	Title of the product
2	category	string	None	Category of the product
3	value	float	0.0	Value of the product
4	currentPrice	float	0.0	Current price of the product
5	quantity	int	0	Quantity available in stock
6	description	string	None	Description of the product
7	barcode	string	None	Barcode number
8	warehouseEntryDate	date	None	Date when product was added to warehouse
9	dimensions	int	0	Dimensions of the product
10	weight	Int	0	Weight of the product
11	productId	string	Genersted by system	Unique ID for the product

5.2. Operation Design

STT	Name	Return type	Description (purpose)
1	saveUpdateProductInfo(product)	void	Saves the updated product information

▪ **saveUpdateProductInfo Method**

- *Parameter:*

- product: Product - The product to save/update.

- *Exception:*

- SaveException if the saving process fails.

- *How to use parameters/attributes:*

This method stores the product details after an update or modification.