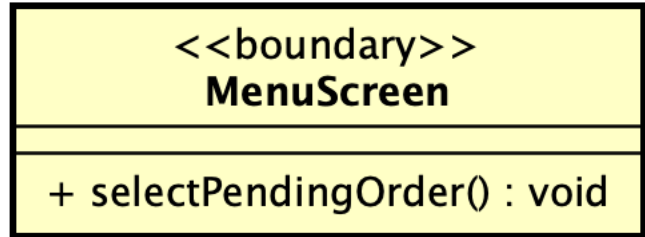


# processPendingOrder

## 1. Design for class “MenuScreen”



1.1. Attribute Design:

None

1.2. Operation Design:

STT	Name	Return Type	Description (Purpose)
1	selectPendingOrder()	void	Triggers allows user to select a pending order

### ▪ selectPendingOrder Method

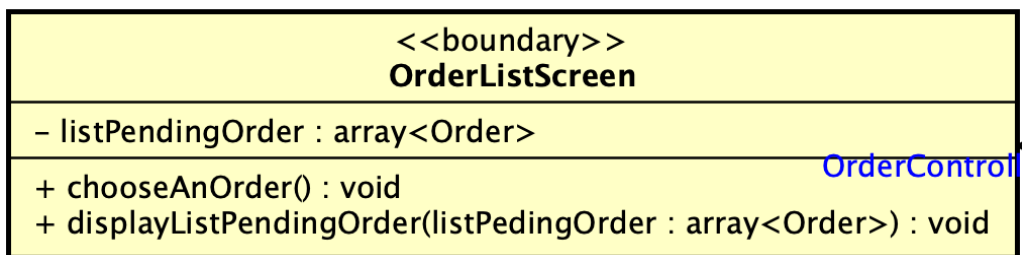
- *Exception:*

- SelectionException if selection fails.

- *How to use parameters/attributes:*

This method enables the user to choose a pending order from the available list.

## 2. Design for class “OrderListScreen”



2.1. Attribute Design:

STT	Name	Data type	Default Value	Description
1	listPendingOrder	Array<Order>	null	Stores the list of pending orders

2.2. Operation Design

STT	Name	Return type	Description
-----	------	-------------	-------------

1	chooseAnOrder()	Void	Allows user to choose an order from the list
2	displayListPendingOrder(listPendingOrder : array<Order>)	void	Displays the list of pending orders

#### chooseAnOrder Method

*Exception:*

- OrderSelectionException if no order is selected or an invalid selection occurs.

*How to use parameters/attributes:*

This method allows the user to select an order from the available list.

#### displayListPendingOrder Method

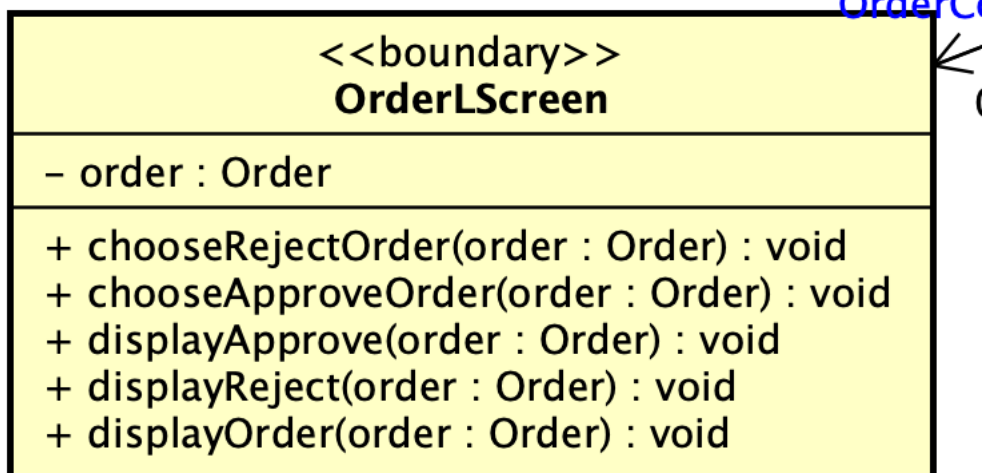
*Exception:*

- DisplayException if the list cannot be retrieved or displayed.

*How to use parameters/attributes:*

This method retrieves and displays all pending orders from listPendingOrder

### 3. Design for class “OrderLScreen”



#### 3.1. Attribute Design

STT	Name	Data type	Default Value	Description
1	order	Order	null	Stores the selected order

### 3.2. Operation Design

STT	Name	Return type	Description (purpose)
1	chooseRejectOrder(order: Order)	Void	Allows user to reject an order
2	chooseApproveOrder(order: Order)	Void	Allows user to approve an order
3	approveOrder(order: Order)	Void	Displays approval confirmation
4	displayReject(order: Order)		Displays rejection confirmation
5	displayOrder(order: Order)	void	Displays order details

#### chooseRejectOrder Method

##### *Exception:*

- OrderProcessingException if the rejection fails.

##### *How to use parameters/attributes:*

This method enables the user to reject an order based on provided order details.

#### chooseApproveOrder Method

##### *Exception:*

- OrderProcessingException if the approval fails.

##### *How to use parameters/attributes:*

This method enables the user to approve an order and proceed to processing.

#### displayReject Method

##### *Exception:*

- DisplayException if the rejection details cannot be retrieved.

*How to use parameters/attributes:*

This method retrieves and displays the rejection details of the selected order.

approveOrder Method

*Exception:*

- ApprovalException if the approval process fails.

*How to use parameters/attributes:*

This method processes and confirms the approval of the selected order.

displayOrder Method

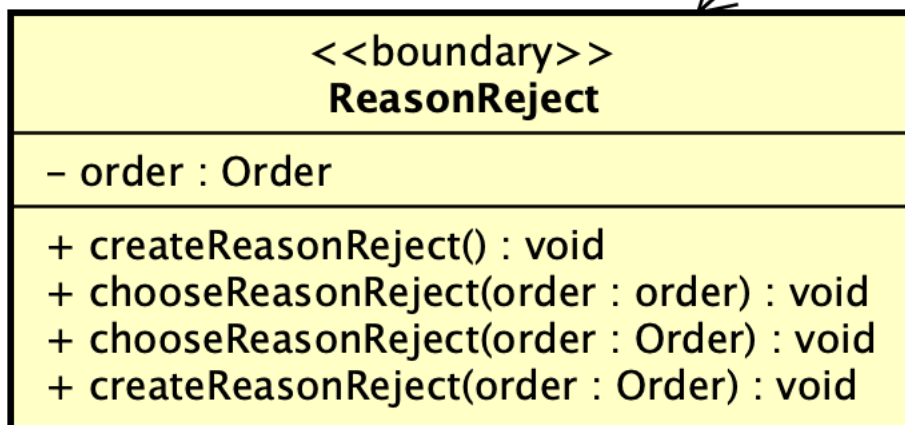
*Exception:*

- DisplayException if the order details cannot be retrieved.

*How to use parameters/attributes:*

This method retrieves and displays details of the selected order.

#### 4. Design for class “ReasonReject”



##### 4.1. Attribute Design:

STT	Name	Data type	Default Value	Description
-----	------	-----------	---------------	-------------

1	order	Order	null	Stores the selected order
---	-------	-------	------	---------------------------

#### 4.2. Operation Design

STT	Name	Return type	Description (purpose)
1	createReasonReject()	Void	Creates a rejection reason
2	chooseReasonReject(order: Order)	Void	Allows user to choose a reason
3	displayReasonReject(order: Order)	void	Displays rejection reason

#### **createReasonReject Method**

##### **Exception:**

- ReasonCreationException if reason creation fails.

##### **How to use parameters/attributes:**

This method allows the user to input and store a reason for rejecting an order.

##### chooseReasonReject Method

##### *Exception:*

- SelectionException if no reason is selected or invalid input occurs.

##### *How to use parameters/attributes:*

This method allows the user to select a predefined reason for rejecting an order.

##### displayReasonReject Method

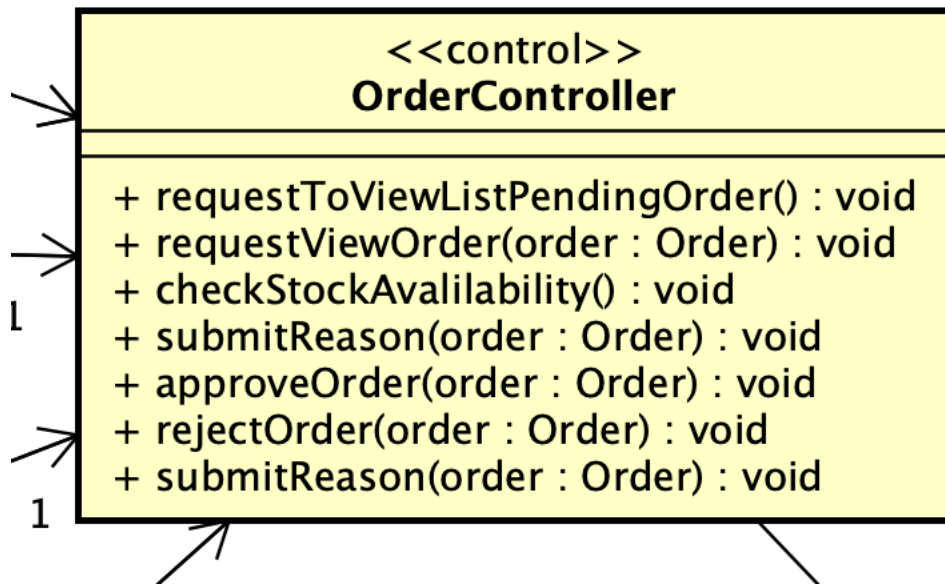
##### *Exception:*

- DisplayException if the reason cannot be displayed.

##### *How to use parameters/attributes:*

This method retrieves and displays the rejection reason associated with the order.

## 5. Design for class “OrderController”



### 5.1. Attribute Design:

None

### 5.2. Operation Design

STT	Name	Return type	Description (purpose)
1	requestToViewListPendingOrder()	Void	Requests to view pending orders list
2	requestViewOrder(order: Order)	Void	Requests details of a specific order
3	checkStockAvailability()	Void	Checks stock availability
4	submitReason(order: Order)	Void	Submits a rejection reason
5	approveOrder(order: Order)	Void	Approves an order

6	rejectOrder(order: Order)	Void	Rejects an order
---	---------------------------	------	------------------

### **requestToViewListPendingOrder Method**

#### **Exception:**

- OrderListException if retrieval fails.

#### **How to use parameters/attributes:**

This method requests and retrieves the list of pending orders.

### **requestViewOrder Method**

#### *Exception:*

- OrderViewException if order details cannot be retrieved.

#### *How to use parameters/attributes:*

This method fetches details of a specific order based on selection.

### **checkStockAvailability Method**

#### *Exception:*

- StockCheckException if stock data is unavailable or inconsistent.

#### *How to use parameters/attributes:*

This method verifies whether the requested order items are available in stock before proceeding.

### **submitReason Method**

#### *Parameter:*

- order: Order - The order being rejected.

#### *Exception:*

- InvalidReasonException if the provided rejection reason is not valid.

*How to use parameters/attributes:*

This method allows the user to submit a reason for rejecting an order.

approveOrder Method

*Parameter:*

- order: Order - The order to be approved.

*Exception:*

- ApprovalException if order approval fails due to validation errors.

*How to use parameters/attributes:*

This method processes the approval of an order and proceeds with fulfillment.

rejectOrder Method

*Parameter:*

- order: Order - The order to be rejected.

*Exception:*

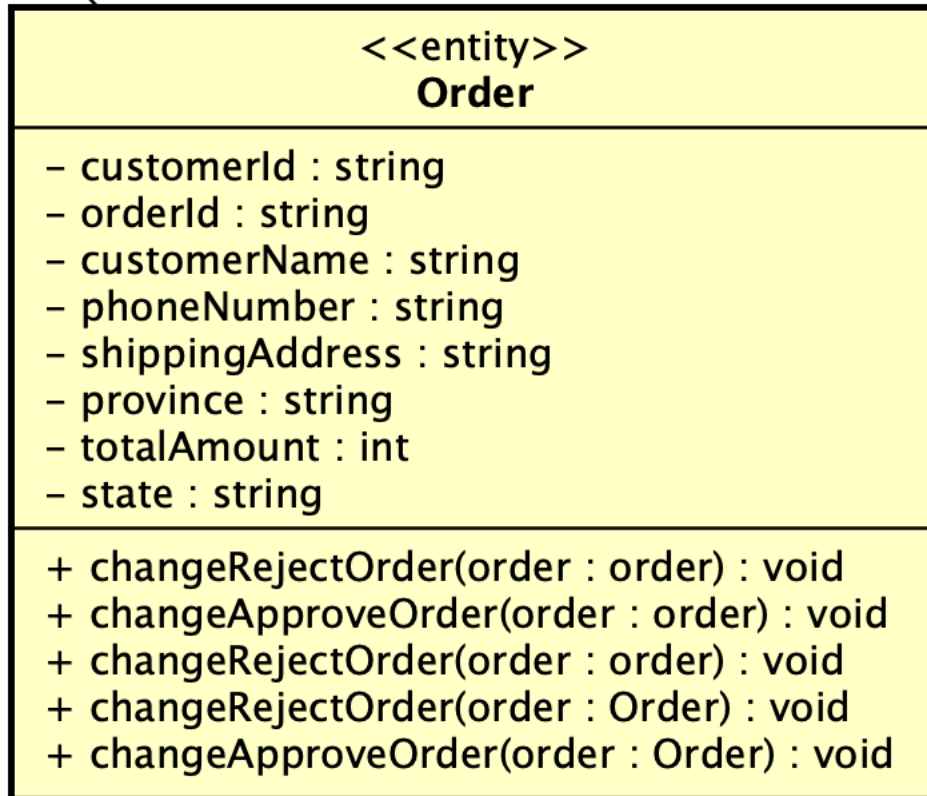
- RejectionException if order rejection fails due to system constraints.

*How to use parameters/attributes:*

This method finalizes the rejection of an order based on user input and validation.

## 6. Design for class “Order”





6.1. Attribute Design:

STT	Name	Data type	Default value	Description
1	orderId	String		Unique identifier for the order
2	customerName	String		Name of the customer placing order
3	totalAmount	Float	0.0	Total amount of the order
4	orderStatus	String	"Pending"	Current status of the order
5	orderDate	String		Date when the order was placed

6	itemList	array		List of items included in the order
---	----------	-------	--	-------------------------------------

## 6.2. Operation Design

STT	Name	Return type	Description (purpose)
1	getOrderDetails()	Order	Retrieves details of the order
2	calculateTotalAmount()	Float	Calculates the total amount of the order
3	updateOrderStatus(status)	Void	Updates the status of the order
4	addItemToOrder(item)	Void	Adds an item to the order
5	removeItemFromOrder(item)	Void	Removes an item from the order

### getOrderDetails Method

#### Exception:

- OrderNotFoundException if the order does not exist.

#### How to use parameters/attributes:

This method retrieves and returns details of a specific order.

#### calculateTotalAmount Method

*Exception:*

- CalculationException if the total amount cannot be calculated.

*How to use parameters/attributes:*

This method iterates through itemList and calculates the total cost.

#### updateOrderStatus Method

*Parameter:*

- status: string - The new status to be updated.

*Exception:*

- InvalidStatusException if the status is not valid.

*How to use parameters/attributes:*

This method updates the **order status** ( **"Pending"**, **"Approved"**, **"Rejected"**).

#### addItemToOrder Method

*Parameter:*

- item: Item - The item to be added to the order.

*Exception:*

- ItemAdditionException if the item cannot be added.

*How to use parameters/attributes:*

This method adds an item to itemList and updates totalAmount accordingly.

#### removeItemFromOrder Method

*Parameter:*

- item: Item - The item to be removed from the order.

*Exception:*

- `ItemRemovalException` if the item is not found or cannot be removed.

*How to use parameters/attributes:*

This method removes an item from `itemList` and recalculates `totalAmount`.