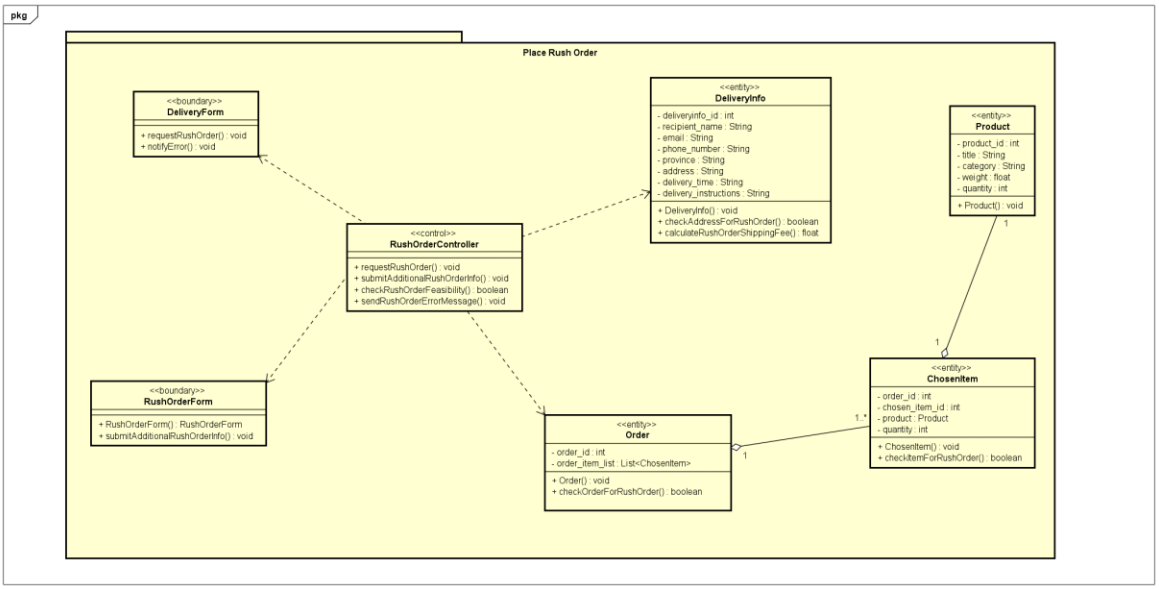
Use Case: Place Rush Order

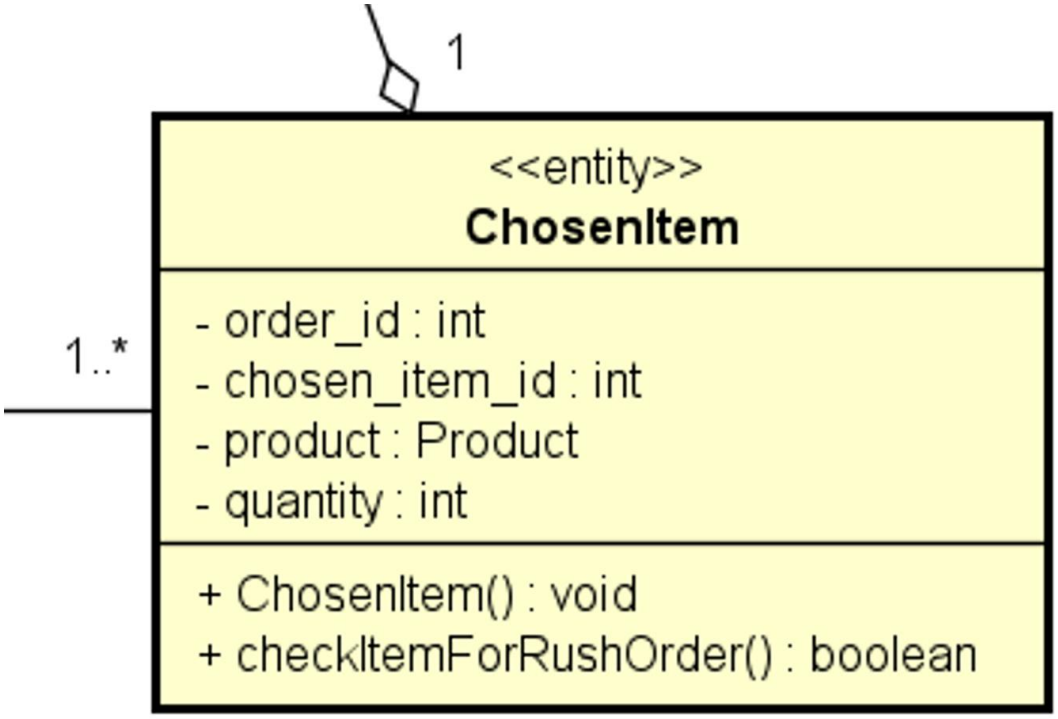
- **Place Rush Order Class Diagram and Relationship**



- **Design Class**
  - **Entity**

    - **ChosenItem Class**

      UML Diagram:



Attributes Design:

#	Name	Data type	Default value	Description
1	order_id	Integer		An Integer identifies the order to which this item belongs
2	chosen_item_id	Integer		An Integer for unique identifier for the specific chosen item
3	product	Product		A Product type that represents the product associated with this chosen item
4	quantity	Integer	0	An Integer that represent the quantity of the specified chosen product

Operations Design:

#	Name	Return type	Description (purpose)
1	ChosenItem	void	This is a constructor that initializes the ChosenItem object
2	checkItemForRushOrder	boolean	A method that check for the feasibility of the chosen item

States:

- Eligible: Chosen item is eligible for Rush Order
- Ineligible: Chosen item is not eligible for Rush Order

Parameter:

- Product product: Product object whose weight will be used to check for Rush Order option
- int quantity: quantity of the Product object that the Customer chooses that will be used to check for Rush Order option

Exception:

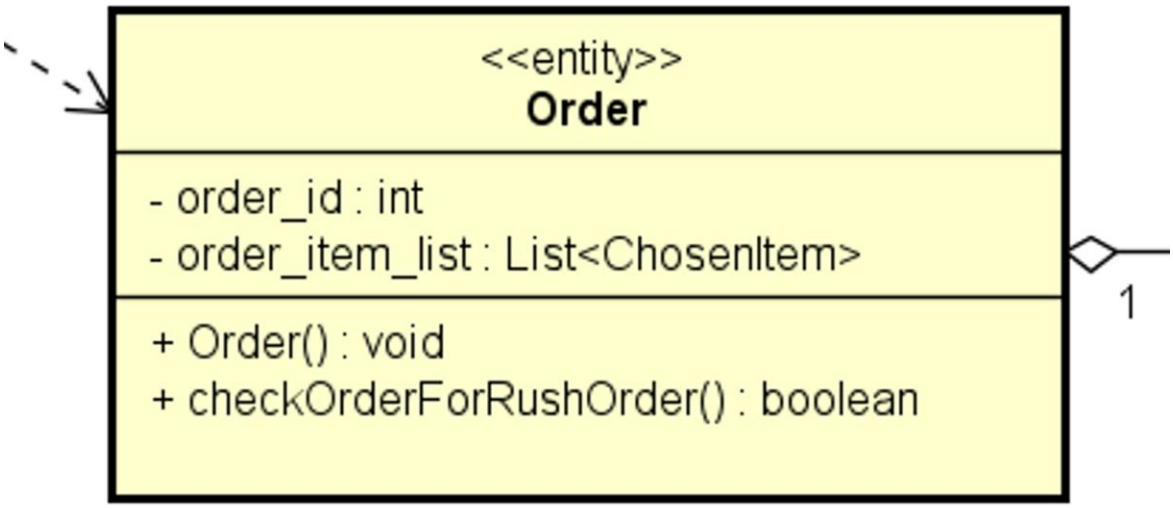
- invalidQuantityException: Thrown if `quantity` is negative or zero

Method:

- checkItemForRushOrder(Product product, int quantity) : boolean
    - Flow:
        - Check if the chosen item allows rush orders based on its weight and quantity
        - If the total weight condition is met, return true
        - Else return false

• Order Class

UML Diagram:



Attributes Design:

#	Name	Data type	Default value	Description
1	order_id	Integer		This is a unique identifier for the order, stored as an integer
2	order_item_list	List<ChosenItem>		This represents a list of ChosenItem objects that track products added to the cart

Operations Design:

#	Name	Return type	Description (purpose)
1	Order	void	This is a constructor that initializes the Order object
2	checkOrderForRushOrder	boolean	A method that examines whether this order is eligible to opt for Rush Order option

States:

- Eligible: Chosen item is eligible for Rush Order
- Ineligible: Chosen item is not eligible for Rush Order

Parameter:

- List<ChosenItem> order_item_list: the list of chosen items that will be checked for the feasibility of the Rush Order option

Exception:

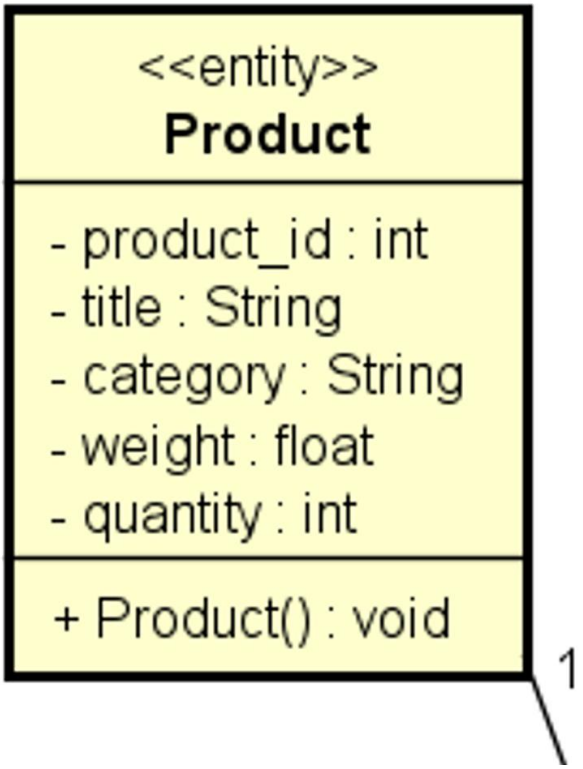
- noItemChosen: Thrown if `order_item_list` is empty

Method:

- checkOrderForRushOrder (List<ChosenItem> order_item_list) : boolean
    - Flow:
        - Doing a while loop through order_item_list to checks if any items in the order is eligible for Rush Order
        - If one item or more is available return True
        - Else return False

• Product Class

UML Diagram:



Attributes:

#	Name	Data type	Default value	Description
1	product_id	Integer		An integer that represents a unique identifier for the product
2	title	String		A String that represents the name or title of the product
3	category	String		A String that represents the classification or category the product belongs to
4	weight	float		A float that represents the weight of the product
5	quantity	integer	0	An integer that represent the available stock

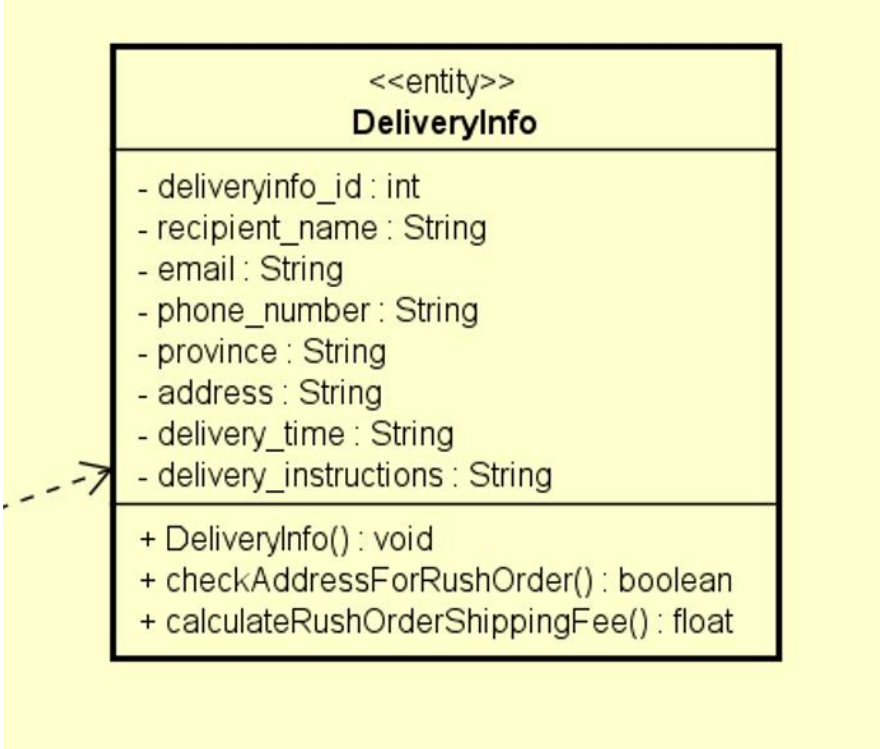
				or inventory count of the product

Operations:

#	Name	Return type	Description (purpose)
1	Product	void	This is a constructor that initializes the Product object

- ## DeliveryInfo Class

UML Diagram:



Attributes:

#	Name	Data type	Default value	Description
1	deliveryinfo_id	Integer		An integer that represent unique identifier for

				the delivery record
2	recipient_name	String		An String that represent the fullname of the person receiving the delivery
3	email	String		An String that represent the email address of the recipient for communication
4	phone_number	String		An String that represent the recipient's phone number for contact
5	province	String		An String that represent the province or state where the delivery is to be made
6	address	String		An String that represent the full address for delivery
7	delivery_time	String		An String that represent the delivery time of the Rush Order option
8	delivery_instructions	String		An String that represent the delivery instructions of the Rush Order option

Operations:

#	Name	Return type	Description (purpose)
1	DeliveryInfo	void	Constructor method for initializing a DeliveryInfo object
2	checkAddressForRushOrder	boolean	A method that checks whether the address of the recipient is available for Rush Order option

3	calculateRushOrderShippingFee	float	A method that calculates the shipping fee of the order with Rush Order option

Parameter:

- String province : this param will be used for checking the validity of the Rush Order option
- String address : this param will be used for checking the validity of the Rush Order option and for Rush Order Shipping Fee Calculation

Exception:

- addressNotAvailable: Thrown if `address` or `province` is empty or not found

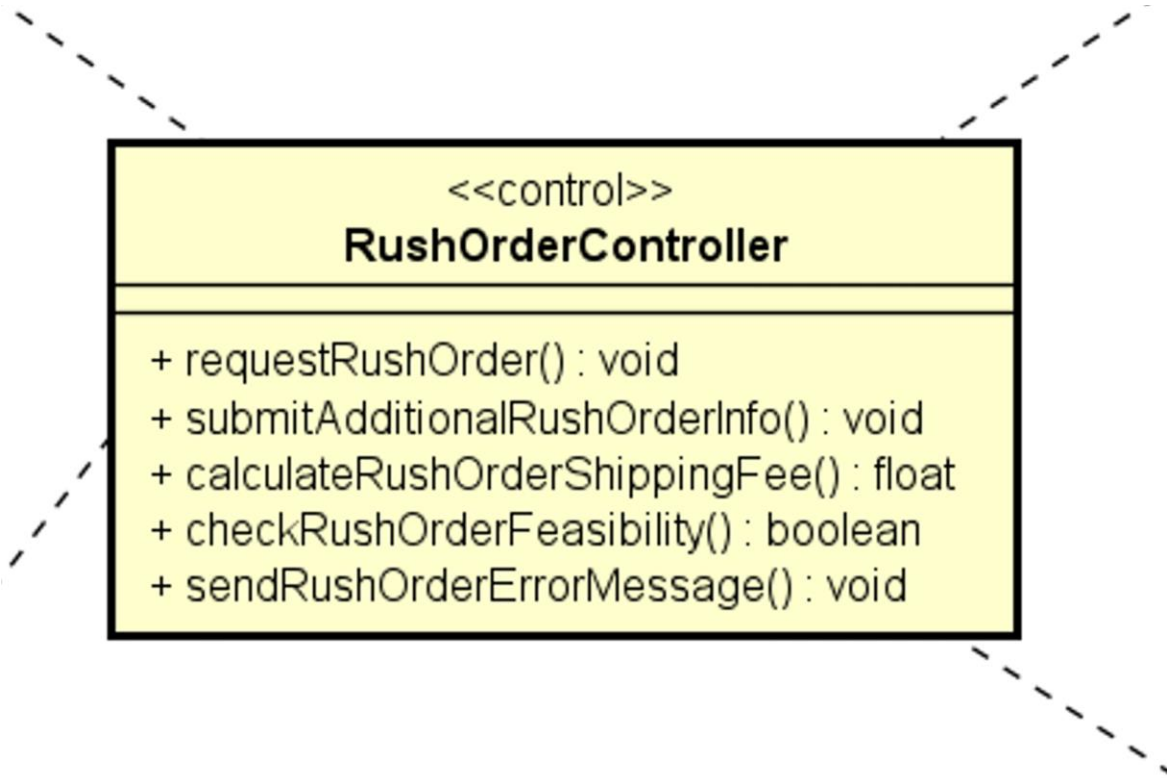
Method:

- CheckAddressForRushOrder (String province, String address) : boolean
    - Flow:
        - Checks for the address condition for Rush Order option
        - If all are valid then return True
        - If not then return False
- calculateRushOrderShippingFee (String address) : float
    - Flow:
        - Calculates the shipping fee for Rush Order option using the distance between the product distribution place and the address of the recipient

- **Controller**

    - **PlaceOrderController Class**

    UML Diagram:



The RushOrderController is responsible for managing the process of handling rush orders in an e-commerce or order management system. It acts as an intermediary between the user interface (UI) and the underlying business logic, ensuring that rush orders are processed efficiently while considering feasibility, additional information, and shipping costs. It performs the following key tasks:

- **Handling Rush Order requests**
- **Checking the feasibility of Rush Orders**
- **Sending error messages for failed Rush Order requests**
- **Submitting additional Rush Order details**
- **Calculating the Rush Order Shipping Fee**