I.   Packages

II.   Class diagram
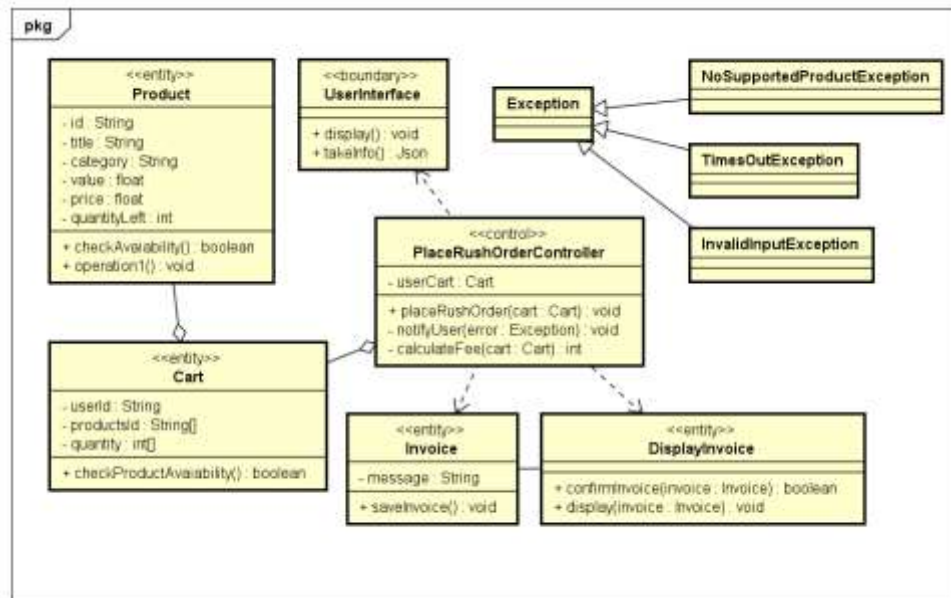
III. Class specification
1.   Class Product

```
              <<entity>>
               Product

          - id : String
          - title : String
          - category : String
          - value : float
          - price : float
          - quantityLeft : int

          + checkAvaiability() : boolean
          + operation1 () : void
```
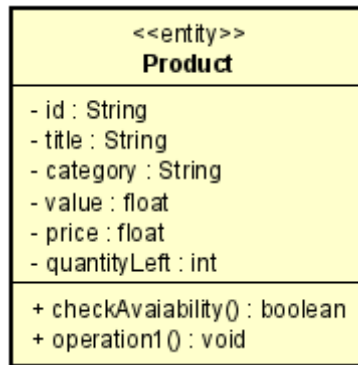
Attributes:

#	Name	Data Type	Default value	Description
1	id	String	Null	//
2	title	String	Null	//
3	category	String	Null	//
4	value	Float	Null	//
5	price	float	null	//
6	quantityLeft	int	Null	Number of product left

Operations:

#	Name	Return Type	Description	Parameter	Exception
1	checkAvaiability	booean	Check wheather this product quantityLeft >0	null	null

Method:

 None

State:

 None

2. Class Cart

```
              <<entity>>
               Cart

          - userId : String
          - productsId : String[]
          - quantity : int[]

          + checkProductAvaiability() : boolean
```
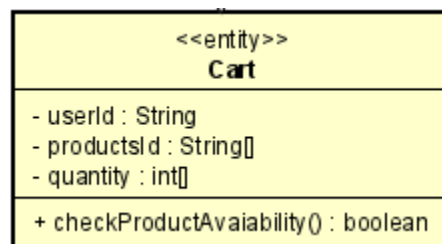
Attributes:

#	Name	Data Type	Default value	Description
1	userId	String	Null	Ower user id
2	productsId	String[]	Null	The products in the cart
3	category	String	Null	//

4	value	Float	Null	//
5	price	float	null	//
6	quantity	Int[]	Null	Number of product putted in the cart corresponding with the productId

Operations:

#	Name	Return Type	Description	Parameter	Exception
1	checkAvaiability	booean	Check wheather each product in the cart is available	null	null

Method:

None

State:

None

3. Class Invoice
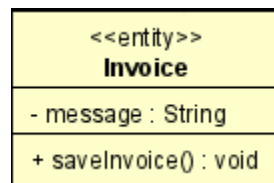
Attributes:

#	Name	Data Type	Default value	Description
1	message	String	null	fee, products supported,…

Operations:

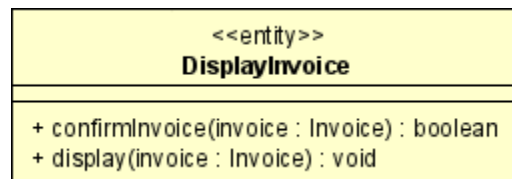| #	Name	Return Type	Description	Parameter	Exception
1	saveInvoice	null	Save this invoice	null	null

Method:

None

State:

None

4. Class DisplayInvoice

Attributes:

#	Name	Data Type	Default value	Description

Operations:

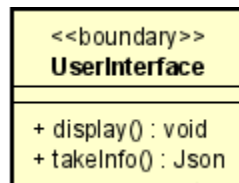| #	Name	Return Type	Description	Parameter	Exception
1	confirmInvoice	booean	Return if the user accept the fee	Invoice – containing the info f the rush order	TimesOutException– if the user take too long to respond
2	display	null	Display the invoice	Invoice – containing ….	null

Method:

None

State:

None

5. Class UserInterface (nothing much to do, since it using same interface of Place Order use case, the following is the simple version of the interface in the Place Order use case)

Attributes:

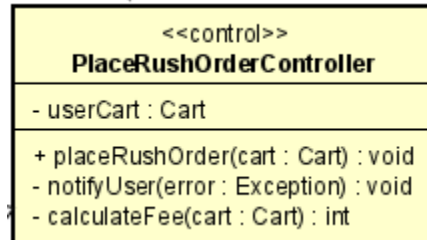| #	Name	Data Type	Default value	Description

Operations:

#	Name	Return Type	Description	Parameter	Exception
1	display	void	Display the interface	null	null
2	takeInfo	Json	Take the info the user inputs	null	InvalidInputException–If the user input is invalid TimesOutException– if the user take too long to respond

Method:

None

State:

None

6. Class PlaceRushOrderController

Attributes:

#	Name	Data Type	Default value	Description
1	userCart	Cart	null	//

Operations:

#	Name	Return Type	Description	Parameter	Exception
1	placeRushOrder	void	Start the place Rush Order process	Cart	NoSupportedProductExcetion – if no product in the cart is supported

Method:

#	Name	Return Type	Description	Parameter	Exception
1	calculateFee	int	Calculating the place rush order fee	cart	null
2	notifyUser	void	Notify User if an error occur	error	null

State:

None