

# CLASS DESIGN

## 1. Class “PlaceRushOrderController”

<<control>> PlaceRushOrderController	
- card : Card - cart : Cart - deliveryInfo : RushDeliveryInfo - invoice : Invoice	
+ requestToPlaceRushOrder(items : Cart) : boolean - checkIfItemsSupportRushOrder() : boolean - checkIfShippingAddressSupportsRushOrder() : boolean + updateNewShippingInfo(newInfo : RushDeliveryInfo) : void	

### Attribute

#	Name	Data type	Default value	Description
1	card	Card	NULL	Represent the card used for payment
2	cart	Cart	NULL	Represent the cart of items which applied rush order
3	deliveryInfo	RushDeliveryInfo	NULL	Represent information of rush order
4	invoice	Invoice	NULL	Represent invoice of rush order

### Operation

#	Name	Return type	Description (purpose)
1	requestToPlaceRushOrder	boolean	Request to place a rush order
2	checkIfItemsSupportRushOrder	boolean	Check if all items in cart are supported rush order
3	checkIfShippingAddressSupportsRushOrder	boolean	Check if shipping address is supported rush order
4	updateNewShippingInfo	void	Update new shipping information of Rush order

Parameter:

- items – the list of items which Customer chose to request a rush order
- newInfo – new delivery information

Exception: None

**Method**                      None

**State**                        None

## 2. Class “Invoice”

<<entity>> Invoice	
- cart : Cart - card : Card - deliveryInfo : RushDeliveryInfo - totalAmount : double - shippingFee : double	
+ calculateNewShippingFee() : double + Invoice(card : Card, card : Card, deliveryInfo : RushDeliveryInfo) : void + saveInvoice() : void	

### Attribute

#	Name	Data type	Default value	Description
1	card	Card	NULL	Represent the card used for payment
2	cart	Cart	NULL	Represent the cart of items which applied rush order
3	deliveryInfo	RushDeliveryInfo	NULL	Represent information of rush order
4	totalAmount	Double	NULL	Represent total amount of rush order
5	shippingFee	double	NULL	Represent shipping fee of rush order

### Operation

#	Name	Return type	Description (purpose)
1	calculateNewShippingFee	double	Calculate new shipping fee according to new delivery information
2	Invoice	void	Constructor of Invoice
3	saveInvoice	void	Save the final invoice of rush order

#### Parameter:

- card – the credit card used for payment and construct invoice
- cart – rush order cart use to construct invoice
- deliveryInfo - rush order delivery information use to construct invoice

Exception: None

**Method**                      None

**State**                        None

### 3. Class “RushDeliveryInfo”

<<entity>> RushDeliveryInfo	
- receiverName : String - address : String - phoneNumber : String - deliveryTime : String	
+ createRushDeliveryInfo(name : String, address : String, phone : String, time : String) : void + update(newInfo : RushDeliveryInfo) : void - checkDeliveryTime(time : String) : void	

#### **Attribute**

#	Name	Data type	Default value	Description
1	receiverName	String	NULL	Represent name of receiver
2	address	String	NULL	Represent address to delivery
3	phoneNumber	String	NULL	Represent phone number of receiver
4	deliveryTime	String	NULL	Represent date time to delivery

#### **Operation**

#	Name	Return type	Description (purpose)
1	createRushDeliveryInfo	void	Initialize detailed rush delivery information
2	update	void	Update rush delivery information

#### *Parameter:*

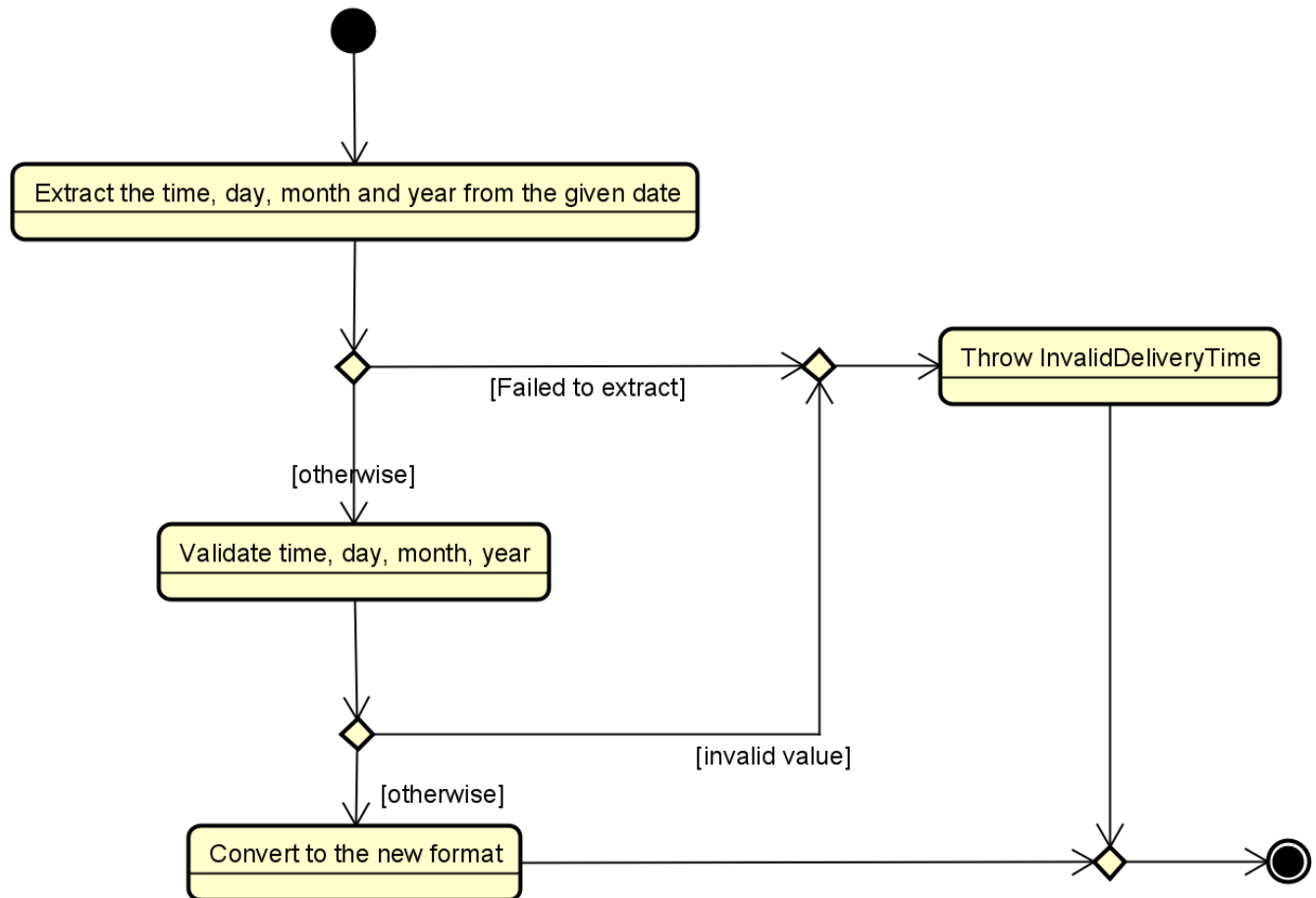
- name – name of receiver
- address – address of receiver
- phone – phone number of receiver
- time – date time that Customer want to delivery in format “HH:MM DD/MM/YY”

*Exception:* None

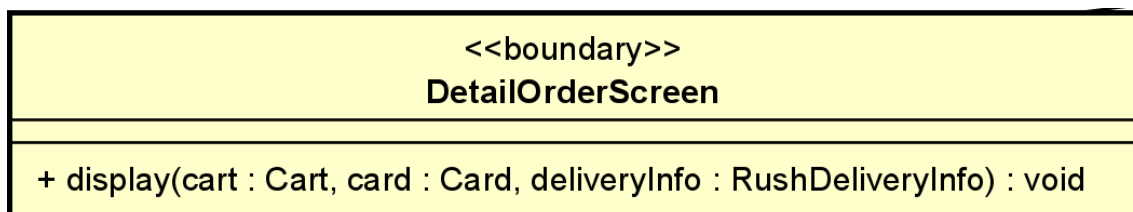
**State**                      None

#### **Method**

- checkDeliveryTime: Given the String “time” representing the delivery date time in the format " HH:MM DD/MM/YY ", this method convert it into the required format “hhmm ddmmyy.”  
The algorithm is illustrated as follows



#### 4. Class “DetailOrderScreen”



#### Attribute

#### Operation

#	Name	Return type	Description (purpose)
1	display	void	Display detailed cart items, credit card and delivery information of rush order

#### Parameter:

- card – the credit card used for payment
- cart – rush order cart
- deliveryInfo - rush order delivery information

Exception: None

#### Method

None

#### State

None