# CLASS SPECIFICATION

## General class diagram

## The Entity package

### Media

* + 1. **The Media class**

Text

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data Type* | *Default value* | *Description* |
| 1 | Id | Int | NULL | The id of the media |
| 2 | Price | Int | NULL | The price of the media |
| 3 | Category | String | NULL | The category that the media belongs to |
| 4 | Value | Int | NULL | The actual value of the media |
| 5 | Title | String | NULL | The name of the media |
| 6 | Quantity | Int | NULL | The number of the media’s copies left |
| 7 | ImageURL | String | NULL | The link to the image of the media |

**Operation**

**Method**

**State**

* + 1. **The Book class**

**Text

Description automatically generated with medium confidence**

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data Type* | *Default value* | *Description* |
| 1 | Author | String | NULL | The author of the book |
| 2 | CoverType | String | NULL | The type of the book cover |
| 3 | Publisher | String | NULL | The publisher of the book |
| 4 | numOfPages | Int | NULL | The number of pages in the book |
| 5 | language | String | NULL | The language in which the book was written in |
| 6 | publishDate | DateTime | NULL | The day the book was published |
| 7 | bookCategory | String | NULL | The category of the book |

**Operation**

**Method**

**State**

* + 1. **The CD class**

**Text

Description automatically generated with medium confidence**

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data Type* | *Default value* | *Description* |
| 1 | Artist | String | NULL | The artist who sang the song in the CD |
| 2 | RecordLabel | String | NULL | The record label which is responsible for recording the CD |
| 3 | MusicType | String | NULL | The type of the song in the CD |
| 4 | releaseDate | DateTime | NULL | The day the CD was released |

**Operation**

**Method**

**State**

* + 1. **The DVD class**

**A picture containing table

Description automatically generated**

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data Type* | *Default value* | *Description* |
| 1 | DiscType | String | NULL | The type of the video in the DVD |
| 2 | Director | String | NULL | The director of the video |
| 3 | Runtime | String | NULL | The total duration of the video |
| 4 | Studio | String | NULL | The studio which is responsible for recording the video |
| 5 | Subtitle | String | NULL | The language of the subtitle |
| 6 | releaseDate | DateTime | NULL | The day the video was released |

**Operation**

**Method**

**State**

### Cart

* + 1. The Cart class

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data Type* | *Default value* | *Description* |
| 1 | MediaList | List<Media> | NULL | The list of media in the cart |

**Operation**

**Method**

1. **checkAvaiability()**

Check whether the quantity of media left in the storage is enough to meet the number of media ordered in the cart.

*+ Parameter*

None

*+ Exception*

* MediaNotAvailableException

1. **empty()**

Empty the cart after the customer complete the transaction.

+ *Parameter*

None

*+ Exception*

None

**State**

### Order

* + 1. **The Order class**

**Table

Description automatically generated**

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data Type* | *Default value* | *Description* |
| 1 | Cart | Cart | NULL | The cart ordered by customer |
| 2 | Destination | String | NULL | The shipping destination |
| 3 | rushOrder | Boolean | FALSE | Whether the customer place a rush order |

**Operation**

**Method**

1. **saveOrder()**

Save the order after the customer complete the transaction

*+ Parameter*

None

*+ Exception*

None

1. **setDeliveryInfo()**

Change the delivery info according to customer’s requests

+ *Parameter*

None

*+ Exception*

None

**State**

### Card

* + 1. **The Card class**

Text

Description automatically generated with low confidence

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data Type* | *Default value* | *Description* |
| 1 | CardNumber | String | NULL | The number of the card |
| 2 | CardHolderName | String | NULL | The name of the card’s owner |
| 3 | IssuingBank | String | NULL | The original bank of the card |
| 4 | expirationDate | DateTime | NULL | The expiration date of the card |
| 5 | CartSecurityCode | Int | NULL | The security password |

**Operation**

**Method**

**State**

### Payment

* + 1. **The PaymentTransaction class**

Diagram

Description automatically generated with low confidence

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data Type* | *Default value* | *Description* |
| 1 | TransactionID | String | NULL | The ID of the transaction |
| 2 | Card | String | NULL | The card involved in the transaction |
| 3 | Amount | Int | NULL | The amount of money |
| 4 | TransactionTime | DateTime | NULL | The time the transaction was executed |
| 5 | Contents | String | NULL | The content of the transaction |
| 6 | ErrorMessage | String | NULL | The error message if any error occurs |

**Operation**

**Method**

1. **saveTransaction():**

Save the transaction to the database after it is done

*+ Parameters:*

None

*+ Exception:*

None

**State**

* + 1. **The Invoice Class**

**Diagram

Description automatically generated**

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data Type* | *Default value* | *Description* |
| 1 | Id | Int | NULL | The ID of the invoice |
| 2 | Amount | Int | NULL | The amount of money |
| 3 | Order | Order | NULL | The customer’s order |

**Operation**

**Method**

1. **saveInvoice():**

Save the invoice to the database after it is done

*+ Parameters:*

None

*+ Exception:*

None

**State**

## The Controller package

### Base Controller

Diagram

Description automatically generated

**Attribute**

**Operation**

**Method**

**State**

### Place Order Controller

Text

Description automatically generated

**Attribute**

**Operation**

1. **PlaceOrder():**

Create new order based on customer’s cart

+ *Parameters:*

* Cart: customer’s cart

*+ Exception:*

None

1. **checkAvailability():**

Check the availability of all products in customer’s cart

*+ Parameters:*

* Cart: customer’s cart

*+ Exception:*

* MediaNotAvailableException

1. **processDeliveryInfo()**

Process the delivery info inputed by customer

+ *Parameters:*

* *Shipping form: The form returned by delivery screen*

*+ Exception:*

None

1. **calculateShippingFees()**

Calculate the shipping fees base on the shipping information provided by customer

*+ Parameters:*

None

*+ Exception:*

None

1. **validateDeliveryInfo()**

Check whether the delivery information provided by customer is valid

*+ Parameters:*

* Shipping form: The form returned by the delivery screen

*+ Exception:*

* InvalidDeliveryInfoException

1. **confirmInvoice()**

Confirm the invoice

+ *Parameters:*

None

*+ Exception:*

None

**Method**

**State**

### Place Rush Order Controller

Text

Description automatically generated

**Attribute**

**Operation**

1. **checkConstraints():**

Check whether the items in customer’s cart and his or her location satisfy the conditions needed for a rush delivery.

*+ Parameters:*

* Shipping form: The form returned by the shipping screen

*+ Exception:*

* PlaceOrderException

1. **placeRushOrder():**

Place a rush order

+ *Parameters:*

* Shipping form: The form returned by the shipping screen
* Cart: customer’s cart

+ *Exception:*

* PlaceOrderException

1. **calculateShippingFees()**

Calculate the shipping fees base on the shipping information provided by customer

*+ Parameters:*

None

*+ Exception:*

None

1. **processRushDeliveryInfo()**

Process the additional information provided by customer

*+ Parameters:*

* Additional shipping form: The form returned by the delivery screen

*+ Exception:*

* InvalidDeliveryInfoException

1. **confirmInvoice()**

Confirm the invoice

+ *Parameters:*

None

*+ Exception:*

None

**Method**

**State**

### Payment Controller

Diagram

Description automatically generated

**Attribute**

**Operation**

1. **payOrder():**

Process the transaction and call the Interbank interface to proceed the transaction

*+ Parameters:*

* None

*+ Exception:*

* None

**Method**

**State**

### View Cart Controller

A picture containing diagram

Description automatically generated

**Attribute**

**Operation**

**Method**

**State**