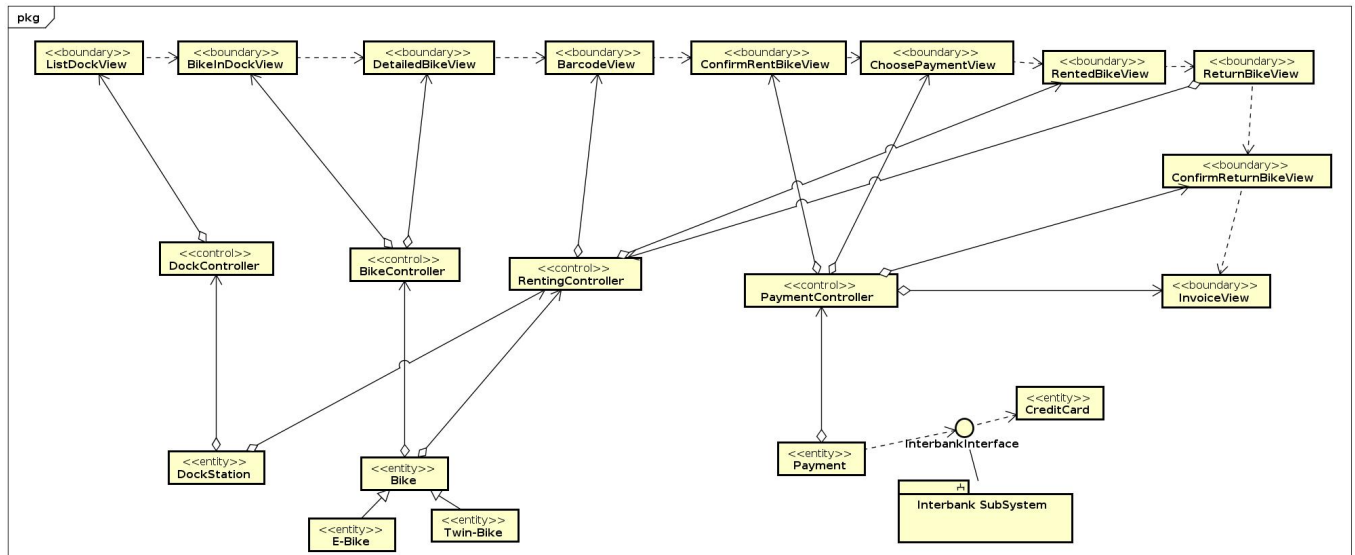


## 1. Class relation



## 2. Class design

### 2.1 Class “ListDockView”



#### Attributes

None

#### Operation

#	Name	Return type	Description
1	showListDockView	void	Display the list dock station view

#### Parameters

DockStation: List<DockStation> - a list of all dock stations

#### Exception

#### Method

#### State

### 2.2 Class “BikeInDockView”



#### Attributes

None

### Operation

#	Name	Return type	Description
1	ShowListBikeView	void	Show the list Bike in dock view

### Parameters

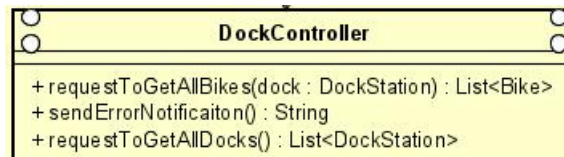
Dock: dockStation – information of the dock

### Exception

### Method

### State

## 2.3 Class “DockController”



### Attributes

None

### Operation

#	Name	Return type	Description
1	requestToGetAllBikes	List<Bike>	Get all bikes from a dock station
2	requestToGetAllDocks	List<DockStation>	Update the cart information
3	sendErrorNotification	String	Send error notification

### Parameters

Dock: dockStation – information of the dock

### Exception

### Method

### State

## 2.4 Class “DockStation”

<<entity>> <b>DockStation</b>
- dockName : String - listBike : List<Bike> - dockArea : String - dockSize : int
+ getDockName() : String + getListBike() : List<Bike> + getDockArea() : String + getDockSize() : int

### Attributes

dockName – name of the dock

listBike – list bike in the dock

dockArea – example: 120x100 (m2)

dockSize – total number of bike can be fit in dock

### Operation

#	Name	Return type	Description
1	getDockName	String	Get name of the dock station
2	getListBike	List<Bike>	Get list bike in the dock station
3	getDockArea	String	Get dock area

### Parameters

### Exception

### Method

### State

## 2.5 Class “PaymentView”

<<boundary>> <b>ChoosePaymentView</b>
+ <<exception>> ChoosePaymentMethod(card : CreditCard) : void

### Attributes

None

### Operation

#	Name	Return type	Description
1	ChoosePaymentMethod	Void	Choose Payment method for payment/refund

### Parameters

card – credit card used for payment/refund

### Exception

PaymentException - If responded with a pre-defined error code

UnrecognizedException – If responded with unknown error code or something goes wrong

### Method

None

### State

None

## 2.6 Class “PaymentController”

<<control>> PaymentController	
- card : CreditCard	
- interbank : InterbankInterface	
+ deductMoney(amount : int, card : CreditCard) : Map<String,String>	
+ returnDepositMoney(card : CreditCard, amount : int) : Map<String,String>	
+ checkAccountInfo(card : CreditCard) : Map<String,String>	

### Attributes

Card: CreditCard - Represent the card used for payment

Interbank:InterbankInterface - Represent the Interbank system

### Operation

#	Name	Return type	Description
1	DeductMoney	Map<String,String>	Deduct money for renting and return the result with a message
2	ReturnDepositMoney	Map<String,String>	Return deposit money when user return bike
3	CheckAccountInfo	Map<String,String>	Check credit card information and return result with a message

### Parameters

card – credit card used for payment/refund

amount – the amount to pay

**Exception**

None

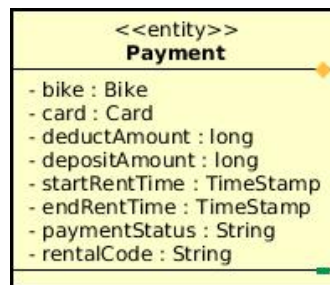
**Method**

None

**State**

None

**2.7 Class “Payment”**



**Attributes**

#	Name	Data type	Default value	Description
1	Bike	Bike	NULL	Represent the bike which user rented
2	card	Card	NULL	Represent the card used for payment
3	DeductAmount	Long	NULL	Amount of deduct money
4	DepositAmount	Long	NULL	Amount of deposit money
5	StartRentTime	TimeStamp	NULL	Time when user start renting
6	EndRentTime	TimeStamp	NULL	Time when user end renting
7	paymentStatus	String	NULL	Status of the payment
8	RentalCode	String	NULL	Unique code of every rental

**Operation**

None

## Parameters

card – credit card used for payment/refund

amount – the amount to pay

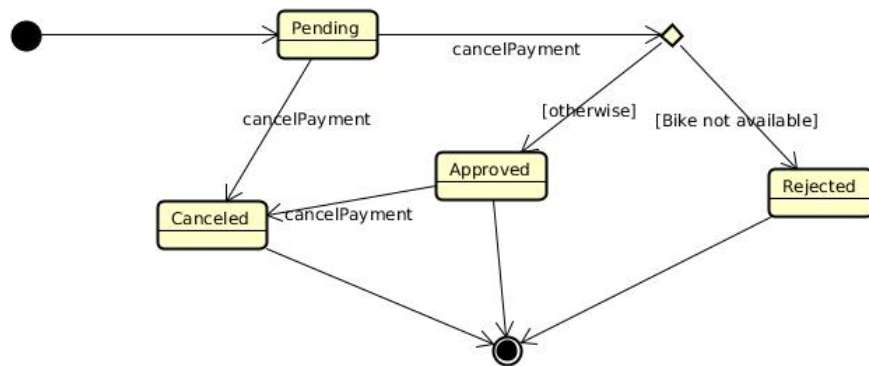
## Exception

None

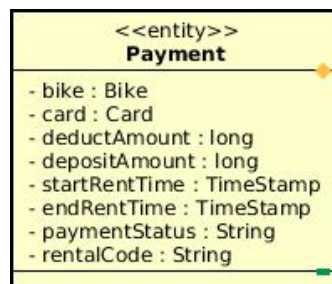
## Method

None

## State



## 2.8 Class “Payment”



## Attributes

None

## Operation

#	Name	Return type	Description
1	showDetailedBikeInfo	Void	Display the detailed bike information view

### Parameters

bike – the bike selected

### Exception

None

### Method

None

### State

None

## 2.9 Class “RentedBikeView”

RentedBikeView
+ showRentedBikeInfo(bike : Bike) : void

### Attributes

None

### Operation

#	Name	Return type	Description
1	showRentedBikeInfo	Void	Display the rented bike information view

### Parameters

bike – the bike selected

### Exception

None

### Method

None

### State

None

## 2.10 Class “BikeController”

BikeController
+ requestBikeInfo(bike : Bike) : void + updateLockStatus(bike : Bike) : void

### Attributes

None

### Operation

#	Name	Return type	Description
1	requestBikeInfo	Void	Request all related information for selected bike from database
2	updateLockStatus	Void	Change rented bike's lock status

### Parameters

bike – the bike selected

### Exception

None

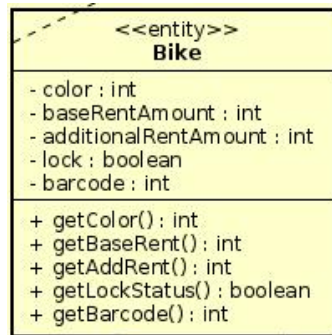
### Method

None

### State

None

## 2.11 Class “Bike”



### Attributes

#	Name	Data type	Default value	Description
1	color	int	NULL	Color of the bike
2	baseRentAmount	int	NULL	Base amount of rent for the bike
3	additionalRentAmount	int	NULL	Additional amount of rent for the bike
4	lock	boolean	NULL	Lock status for rented bike



5	barcode	int	NULL	Barcode of the bike
---	---------	-----	------	---------------------

### Operation

#	Name	Return type	Description
1	getColor	int	Request selected bike's color
2	getBaseRent	int	Request selected bike's base rent amount
3	getAddRent	int	Request selected bike's additional rent amount
4	getLockStatus	boolean	Request selected bike's lock status
5	getBarcode	int	Request selected bike's barcode

### Parameters

bike – the bike selected

### Exception

None

### Method

None

### State

None

## 2.12 Class “RentingController”

RentingController
<pre> + lockBike(rentalCode : string) : boolean + requestRentBike(barcode : string, dock : DockStation) : boolean + unlockBike(rentalCode : string) : boolean + calculateRentingAmount(rentalCode : string) : long + getRentedBikeInformation(rentalCode : string) : Bike + requestReturnBike(dock : DockStation, rentalCode : string) : boolean + checkPaymentStatus(rentalCode : string) : boolean + getAvailableDock() : List&lt;DockStation&gt; </pre>

### Attributes

None

### Operation

#	Name	Return type	Description
1	lockBike	boolean	Lock the rented bike
2	unlockBike	boolean	Unlock the rented bike
3	requestRentBike	boolean	Request to rent a bike in dock
4	calculateRentingAmount	Long	Calculate the renting amount of the rented bike
5	getRentedBikeInformation	Bike	Get the information of the rented bike
6	requestReturnBike	Boolean	Request to return a bike to a dock

7	checkPaymentStatus	Boolean	Check whether the transaction has been paid or not
8	getAvailableDock	List<DockStation>	Get available dock for returning bike

### Parameters

rentalCode:string – the rental code of the rented bike

Barcode:string - barcode of the bike

Dock:DockStation - dock station model

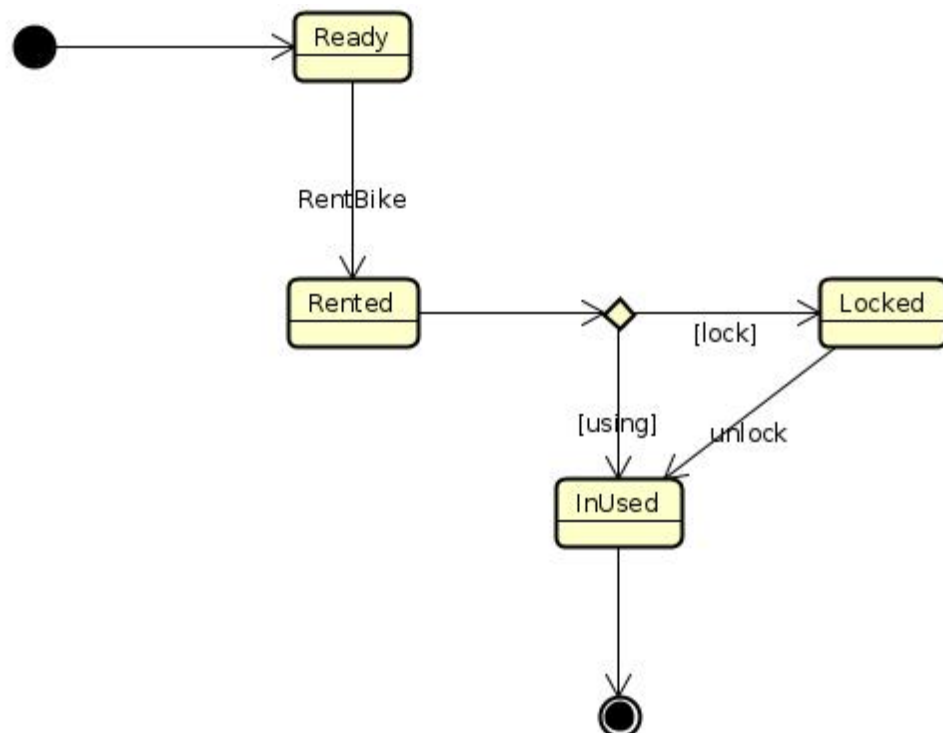
### Exception

None

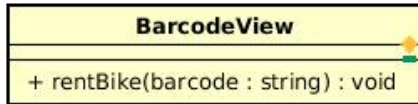
### Method

None

### State



## 2.13 Class “BarcodeView”



### Attributes

None

### Operation

#	Name	Return type	Description
1	rentBike	void	Request to rent bike from user

### Parameters

Barcode:string - barcode of the bike

### Exception

None

### Method

None

### State

None

## 2.14 Class “ReturnBikeView”



### Attributes

None

### Operation

#	Name	Return type	Description
1	showAvailableDock	void	Show available dock to user to return bike
2	returnBike	void	Button to request to return bike

### Parameters

listDock: List<DockStation> - list of DockStation model

bike:Bike – the bike selected

### Exception

dockException - show exception if retrieve dock information failed

**Method**

None

**State**

None

**2.15 Class “RentedBikeView”**

RentedBikeView
+ <<exception>> showBikeInfo(bike : Bike) : void + <<exception>> showPaymentInfo(payment : Payment) : Payment + unlockBike(rentalCode : string) : boolean + lockBike(rentalCode : string) : boolean

**Attributes**

None

**Operation**

#	Name	Return type	Description
1	showBikeInfo	void	Show Information of rented bike
2	showPaymentInfo	Void	Show information about rented bike fee
3	unlockBike	Void	Lock the rented bike from user
4	lockBike	Void	Unlock the rented bike from user

**Parameters**

listDock: List<DockStation> - list of DockStation model

bike:Bike – the bike selected

**Exception**

rentedBikeException - exception raise if the rented bike is not eligible

paymentException - exception raise if there is no payment with the bike

**Method**

None

**State**

None

**3. Class diagram**

