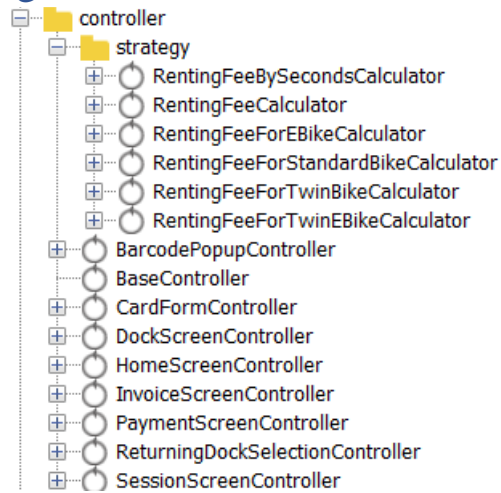
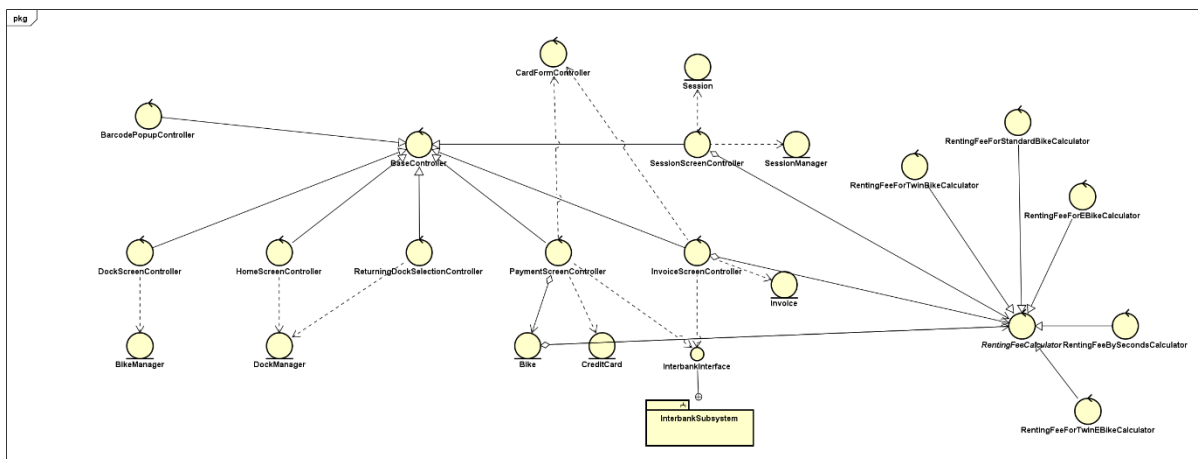


Class Design - Controller

1. Create Initial Design Classes

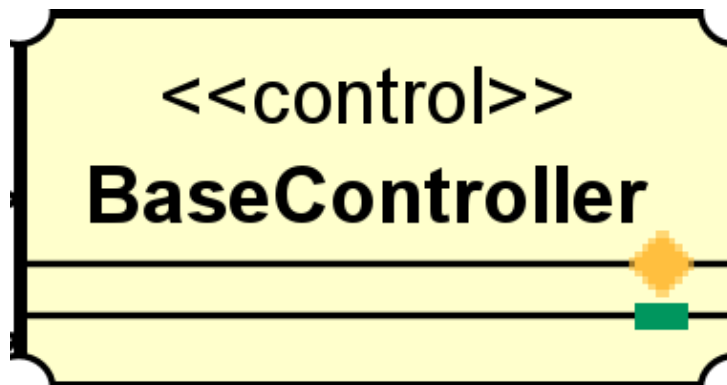


2. Define Relationships Between Classes



3. Class Design

3.1. Class “BaseController”



Attribute

None

Operation

None

Parameter:

None

Exception:

None

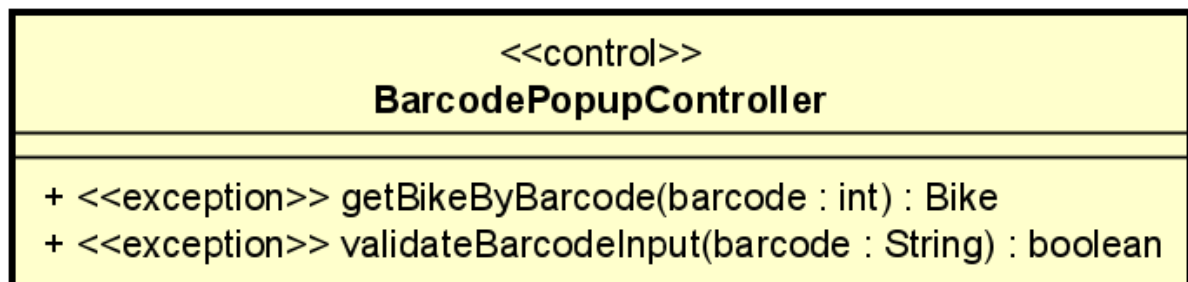
Method

None

State

None

3.2. Class “BarcodePopupController”



Attribute

None

Operation

#	Name	Return type	Description
1	getBikeByBarcode	Bike	Get bike by using barcode
2	ValidateBarcodeInput	boolean	Checks if barcode is entered or not

Parameter:

- Parameter of getBikeByBarcode()
 - barcode– int
- Parameter of ValidateBarcodeInput()
 - barcode - string

Exception:

- BarcodeNotFoundException
- NullBarcodeException
- InvalidBarcodeException

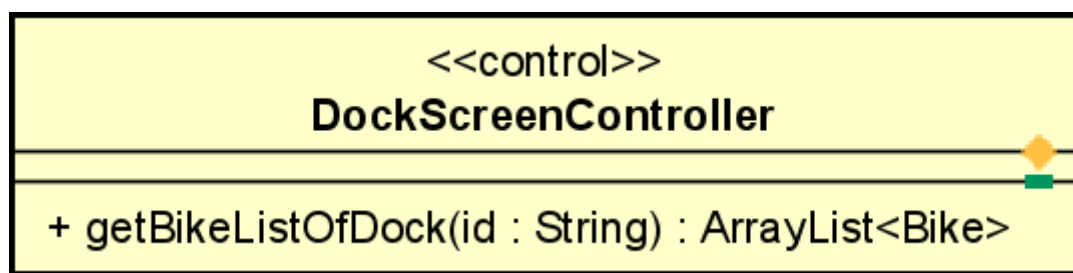
Method

None

State

None

3.3. Class “DockScreenController”



Attribute

None

Operation

#	Name	Return type	Description
1	getBikeListOfDock	ArrayList<Bike>	Get bike list by using dock id

Parameter:

- Parameter of getBikeListOfDock()
 - Id – dock id

Exception:

None

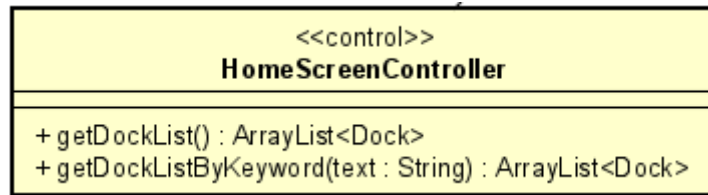
Method

None

State

None

3.4. Class “HomeScreenController”



Attribute

None

Operation

#	Name	Return type	Description
1	getDockList	ArrayList<Dock>	Get dock list
2	getDockListByKeyword	ArrayList<Dock>	Get dock list by keyword

Parameter:

- Parameter of getDockListByKeyword()
 - Text - keyword

Exception:

None

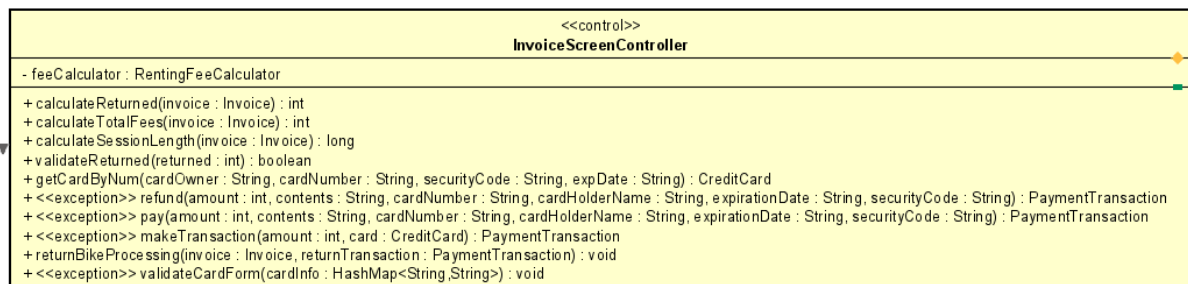
Method

None

State

None

3.5. Class “InvoiceScreenController”



Attribute

#	Name	Data type	Default value	Description
---	------	-----------	---------------	-------------

1	feeCalculator	RentingFeeCalculator	new RentingFeeBySecondsCalculator	calculate fee
---	---------------	----------------------	-----------------------------------	---------------

Operation

#	Name	Return type	Description
1	calculateReturned	Int	Calculate returned after deducting renting fees
2	calculateTotalFees	Int	Calculate total fees
3	calculateSessionLength	long	Session length
4	refund	PaymentTransaction	Create return transaction
5	validateReturned	boolean	Check returned validity
6	getCardByNum	CreditCard	Get credit card info
7	refund	PaymentTransaction	Setup Return transaction
8	pay	PaymentTransaction	Setup transaction
9	makeTransaction	PaymentTransaction	Proceed transaction
10	returnBikeProcessing	void	Process return bike
11	validateCardForm	void	Check card form validity

Parameter:

- Parameter of calculateReturned()
 - Invoice
- Parameter of calculateTotalFees()
 - Invoice
- Parameter of calculateSessionLength()
 - Invoice
- Amount Parameter of getCardByNum()
 - cardOwner
 - cardNumber
 - securityCode
 - expDate
- Parameter of refund()
 - amount
 - contents
 - cardNumber
 - cardHolderName
 - expirationDate
 - securityCode
- Parameter of pay()
 - amount
 - contents
 - cardNumber
 - cardHolderName
 - expirationDate
 - securityCode
- ExpirationDate Parameter of makeTransaction()
 - amount
 - card

- Parameter of validateReturned()
 - returned
- Parameter of returnBikeProcessing()
 - Invoice
 - returnTransaction
- Parameter of validateCardForm()
 - cardInfo

Exception:

PaymentException

NullCardNumberException

NullCardOwnerException

NullExpDateException

NullSecurityCodeException

CardExpiredException

Method

None

State

None

3.6. Class “PaymentScreenController”

<<control>> PaymentScreenController	
- bike : Bike	
- cardInfo : HashMap<String, String>	
+ payDeposit(amount : int, contents : String, cardNumber : String, cardOwner : String, expirationDate : String, securityCode : String) : PaymentTransaction + <<exception>> validateCreditCardForm(creditCardForm : HashMap<String, String>) : boolean + <<exception>> validateCardUnused(cardNumber : String) : boolean + PaymentScreenController(bike : Bike) + PaymentScreenController(bike : Bike, cardInfo : HashMap<String, String>)	

Attribute

#	Name	Data type	Default value	Description
1	bike	Bike	NULL	bike
2	cardInfo	HashMap<String, String>	NULL	Card info

Operation

#	Name	Return type	Description
1	payDeposit	PaymentTransaction	Set rent transaction
2	validateCreditCardForm	boolean	Check creditcard form
3	validateCardUnused	boolean	Check creditcard used or not

Parameter:

- Parameter of payDeposit()
 - amount
 - contents
 - cardNumber
 - cardOwner
 - expirationDate
 - securityCode
- Parameter of validateCreditCardForm()
 - creditCardForm
- Parameter of validateCardUnused()
 - cardNumber

Exception:

None

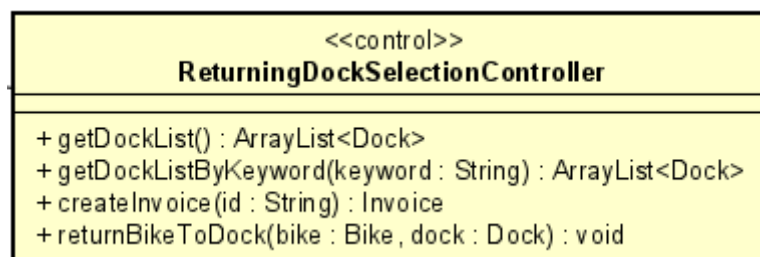
Method

None

State

None

3.7. Class “ReturningDockSelectionController”



Attribute

None

Operation

#	Name	Return type	Description
1	getDockList	ArrayList<Dock>	Get dock list
2	getDockListByKeyWord	ArrayList<Dock>	Get dock list by keyword

3	createInvoice	Invoice	Create new invoice
4	returnBikeToDock	void	Return bike to the chosen dock

Parameter:

- Parameter of getDockListByKeyWord()
 - keyword
- Parameter of createInvoice()
 - id
- Parameter of returnBikeToDock()
 - bike
 - dock

Exception:

None

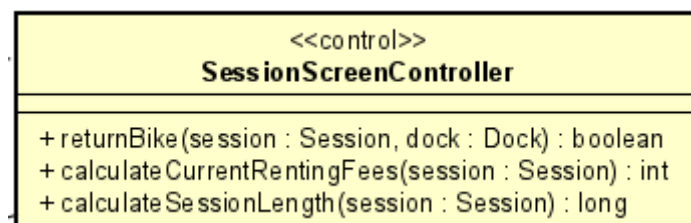
Method

None

State

None

3.8. Class “SessionScreenController”



Attribute

None

Operation

#	Name	Return type	Description
1	returnBike	boolean	Check if bike is available
2	calculateCurrenRentingFees	int	Calculate current renting fees
3	calculateSessionLength	long	Calculate session length

Parameter:

- Parameter of returnBikeToDock()
 - session
 - dock
- Parameter of returnBikeToDock()
 - session
- Parameter of returnBikeToDock()
 - session

Exception:

None

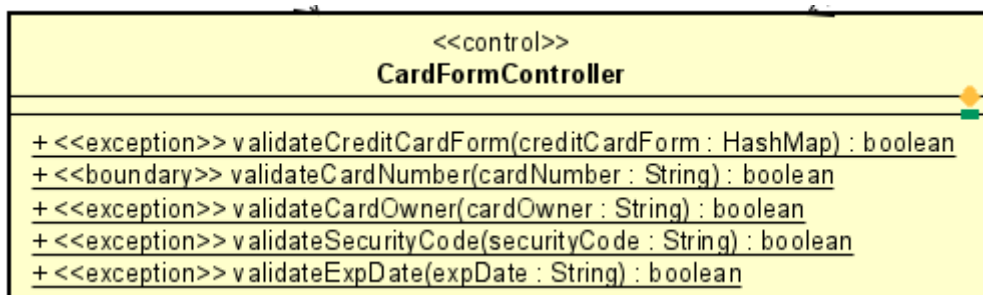
Method

None

State

None

3.9. Class “CardFormController”



Attribute

None

Operation

#	Name	Return type	Description
1	validateCreditCardForm	boolean	Check if card is valid
2	validateCardNumber	boolean	Validate card number
3	validateCardOwner	boolean	Validate card owner
4	validateSecurityCode	boolean	Validate security code
5	validateExpDate	boolean	Validate exp date

Parameter:

- Parameter of validateCreditCardForm()
 - creditCardForm
- Parameter of validateCardNumber()
 - CardNumber
- Parameter of validateCardOwner()
 - cardOwner
- Parameter of validateSecurityCode()
 - securityCode
- Parameter of validateExpDate()
 - expDate

Exception:

None

Method

None

State

None