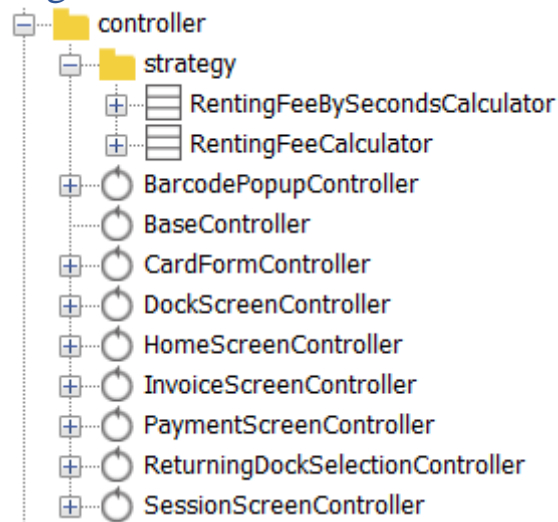
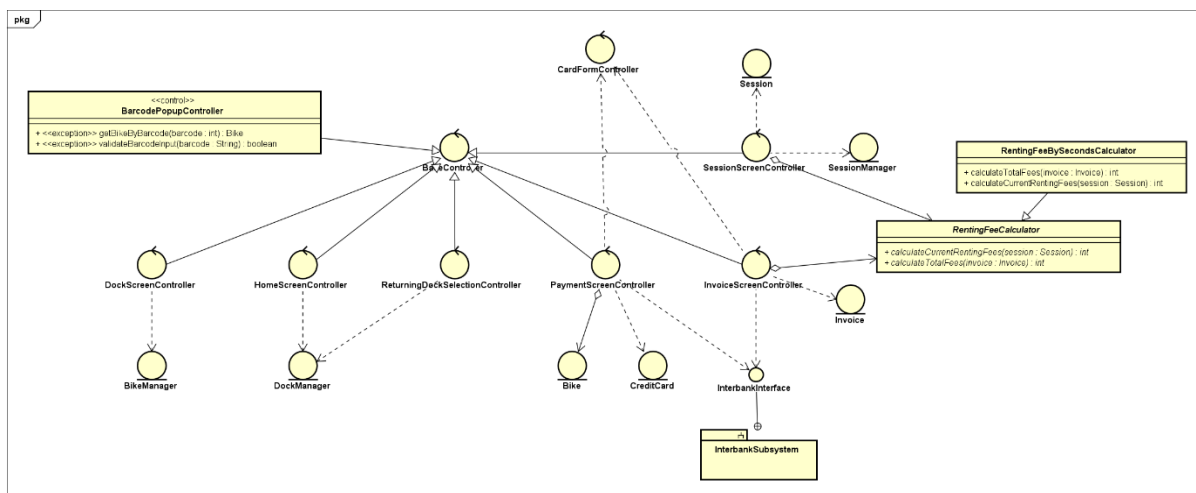


Class Design - Controller

1. Create Initial Design Classes

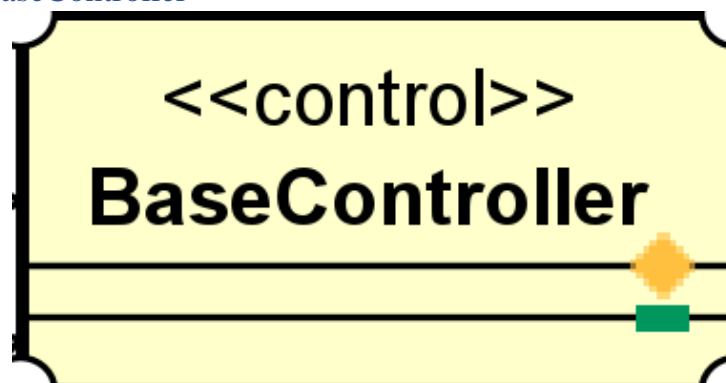


2. Define Relationships Between Classes



3. Class Design

3.1. Class “BaseController”



Attribute

None

Operation

None

Parameter:

None

Exception:

None

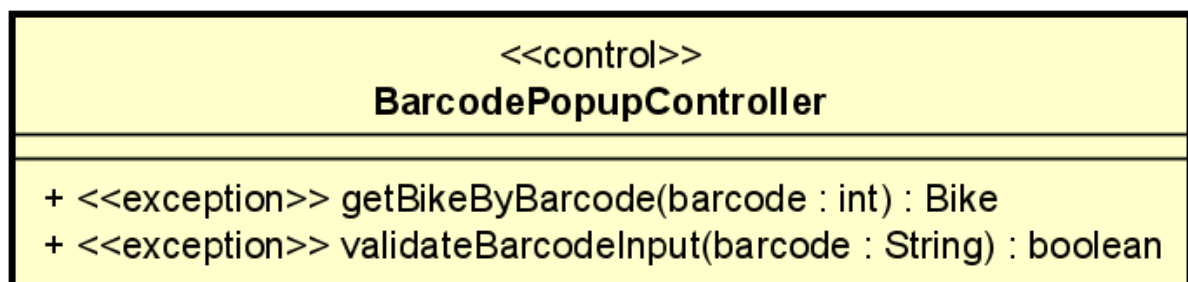
Method

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:

barcode– int

Barcode - string

Exception:

BarcodeNotFoundException

NullBarcodeException

InvalidBarcodeException

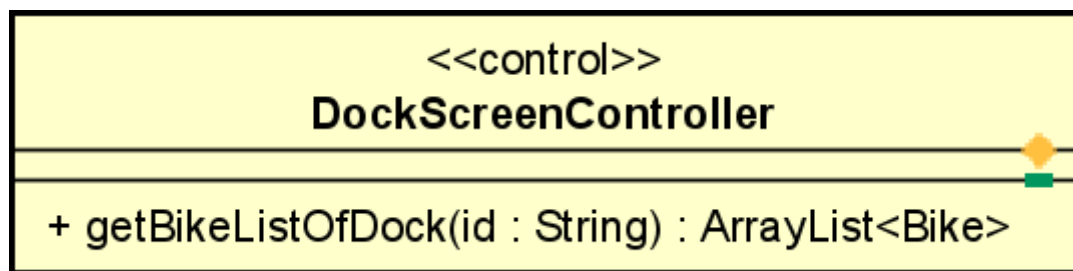
Method

None

State

None

3.3. Class “DockScreenController”



Attribute

None

Operation

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

Parameter:

Id – dock id

Exception:

None

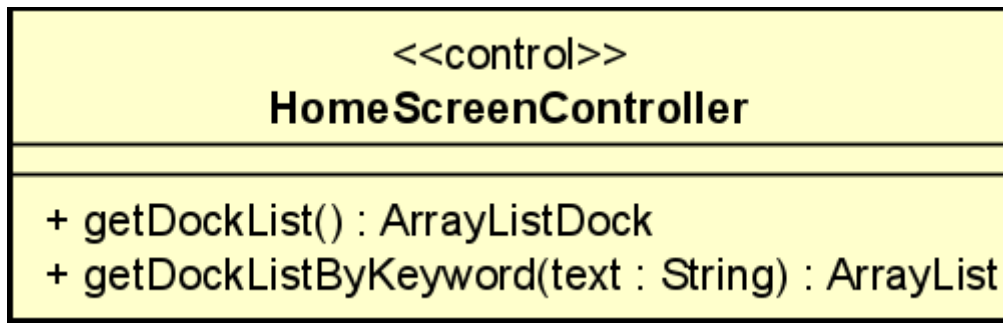
Method

None

State

None

3.4. Class “HomeScreenController”



Attribute

None

Operation

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

Parameter:

Text - keyword

Exception:

None

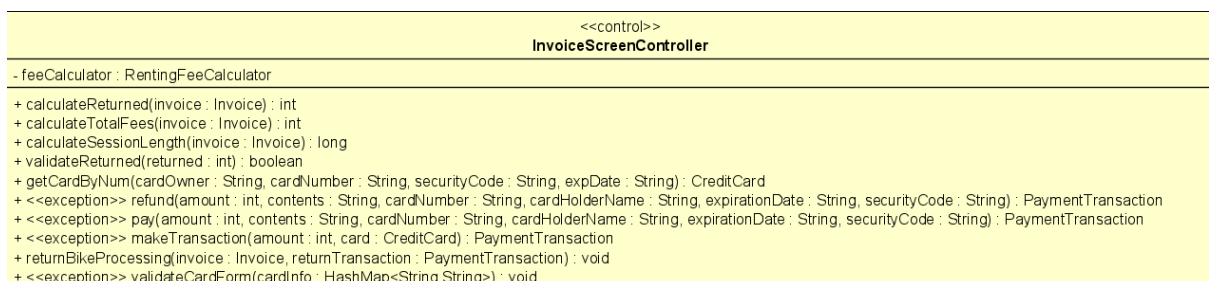
Method

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	validateCardForm	void	Check card form validity

Parameter:

Invoice
 Amount
 Contents
 CardNumber
 CardOwner
 ExpirationDate
 SecurityCode
 Returned

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:

Amount

Contents

CardNumber

CreditCardForm

CardInfo

Bike

Exception:

None

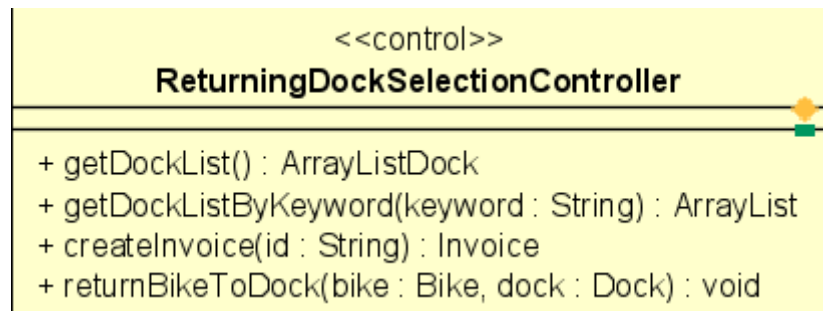
Method

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:

Keyword

Id

Bike

Dock

Exception:

None

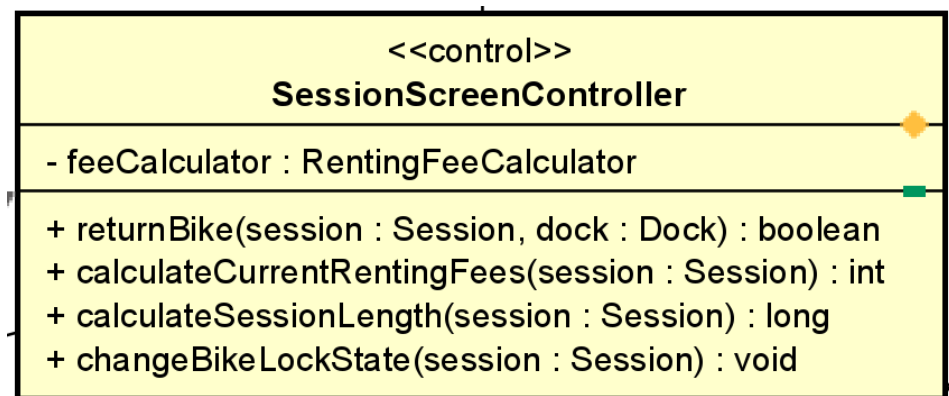
Method

None

State

None

3.8. Class “SessionScreenController”



Attribute

#	Name	Data type	Default value	Description
1	feeCalculator	RentingFeeCalculator	new RentingFeeBySecondsCalculator	calculate fee

Operation

#	Name	Return type	Description
1	returnBike	boolean	Check if bike is available
2	calculateCurrentRentingFees	int	Calculate current renting fees
3	calculateSessionLength	long	Calculate session length
4	changeBikeLockState	void	Change bike lock state

Parameter:

Session

Dock

Exception:

None

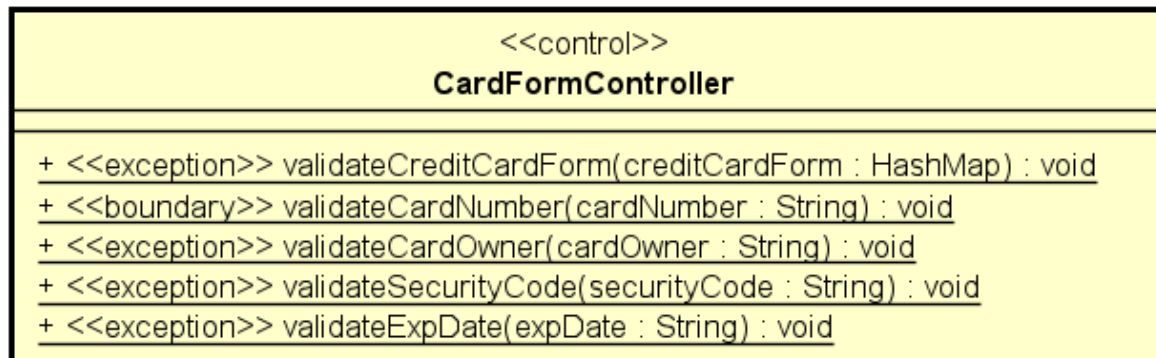
Method

None

State

None

3.9. Class “CardFormController”



Attribute

None

Operation

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

Parameter:

CreditCardForm

CardNumber

CardOwner

SecurityCode

expDate

Exception:

None

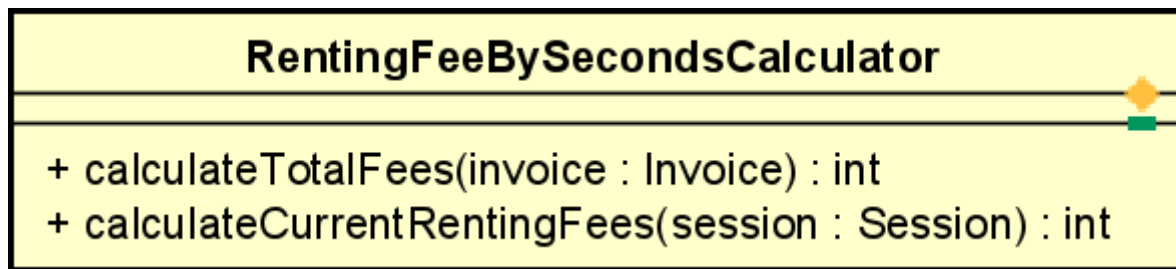
Method

None

State

None

3.10. Class “RentingFeeBySecondsCalculator”



Attribute

None

Operation

#	Name	Return type	Description
1	calculateTotalFees	int	Calculate total fees
2	calculateCurrentRentingFees	int	Calculate current renting fees

Parameter:

Invoice

session

Exception:

None

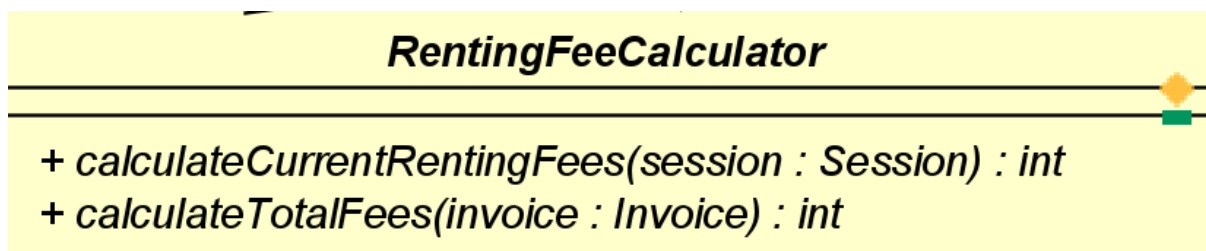
Method

None

State

None

3.11. Class “RentingFeeCalculator”



Attribute

None

Operation

#	Name	Return type	Description
1	calculateTotalFees	int	Calculate total fees
2	calculateCurrentRentingFees	int	Calculate current renting fees

Parameter:

Invoice

session

Exception:

None

Method

None

State

None

4. Class Diagram

