Class Design - Controller

1. Create Initial Design Classes

- controller
  - strategy
    - RentingFeeBySecondsCalculator
    - RentingFeeCalculator
  - BarcodeController
  - BaseController
  - DockScreenController
  - HomeScreenController
  - InvoiceScreenController
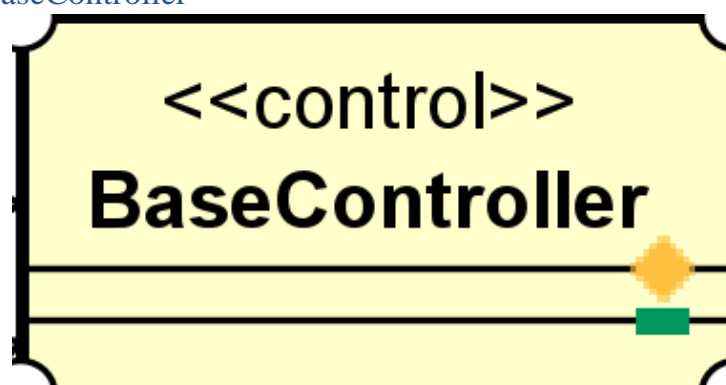 - PaymentScreenController
  - ReturningDockSelectionController
  - SessionScreenController
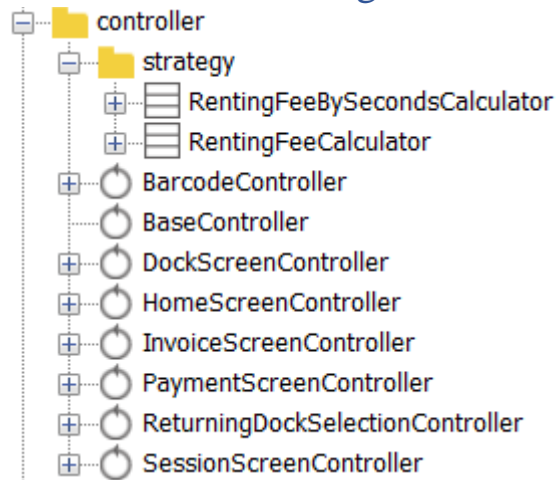
2. Define Relationships Between Classes
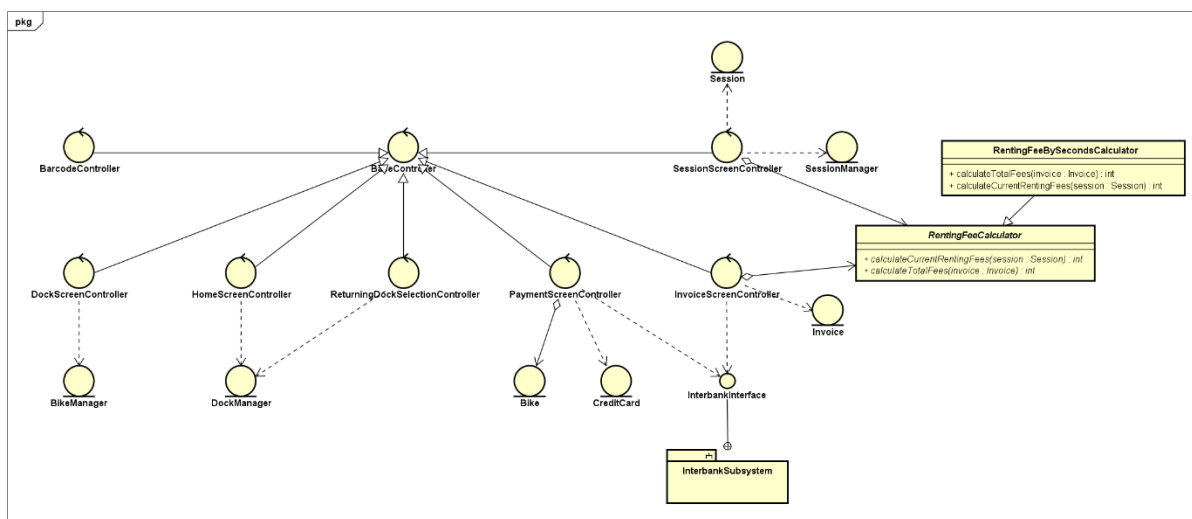
3. Class Design

3.1. Class "BaseController"

Attribute

      None

Operation

      None
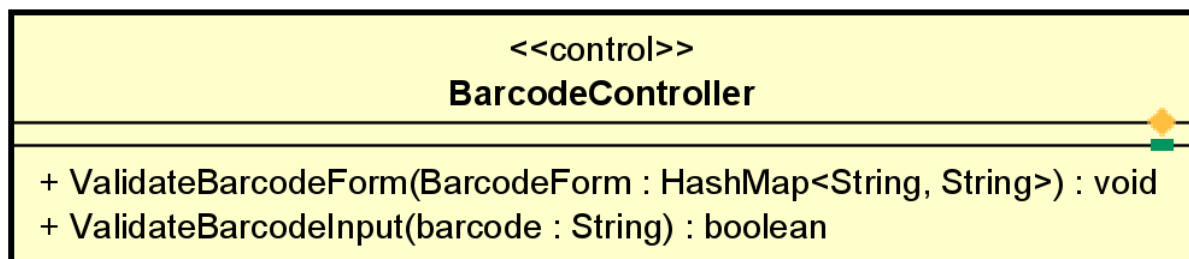
Parameter:

      None

Exception:

      None

Method

      None

State

      None

3.2. Class "BarcodeController"

```
                        <<control>>
                     BarcodeController

+ ValidateBarcodeForm(BarcodeForm : HashMap<String, String>) : void
+ ValidateBarcodeInput(barcode : String) : boolean
```

Attribute

      None

Operation

#	Name	Return type	Desciption
1	ValidateBarcodeForm	void	Validates barcode form
2	ValidateBarcodeInput	boolean	Checks if barcode is entered or not

Parameter:

      BarcodeForm – Barcode

      Barcode - barcode

Exception:

        BarcodeNotFoundException

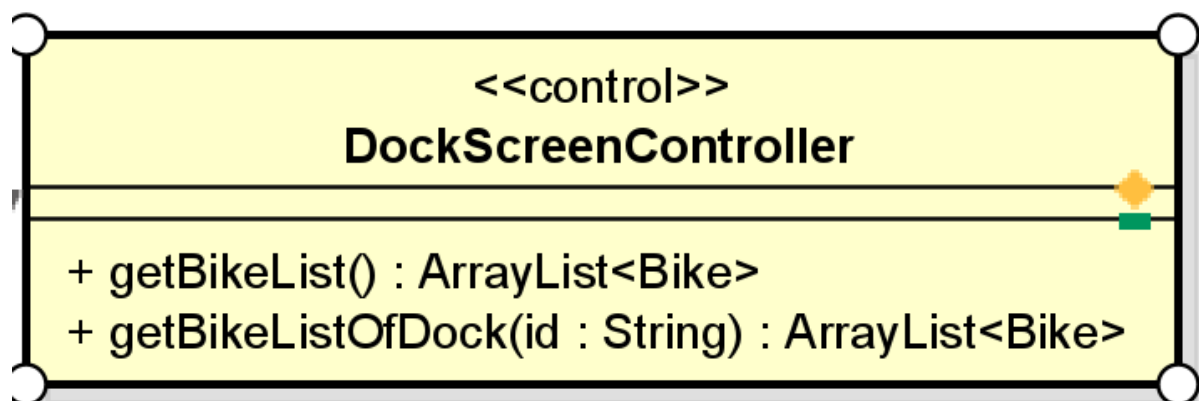   NullBarcodeException

        InvalidBarcodeException

Method

        None

State

        None

3.3. Class "DockScreenController"

Attribute

        None

Operation

#	Name	Return type	Desciption
1	getBikeList	ArrayList	Get bike list
2	getBikeListOfDock	ArrayList	Get bike list by using dock id

Parameter:

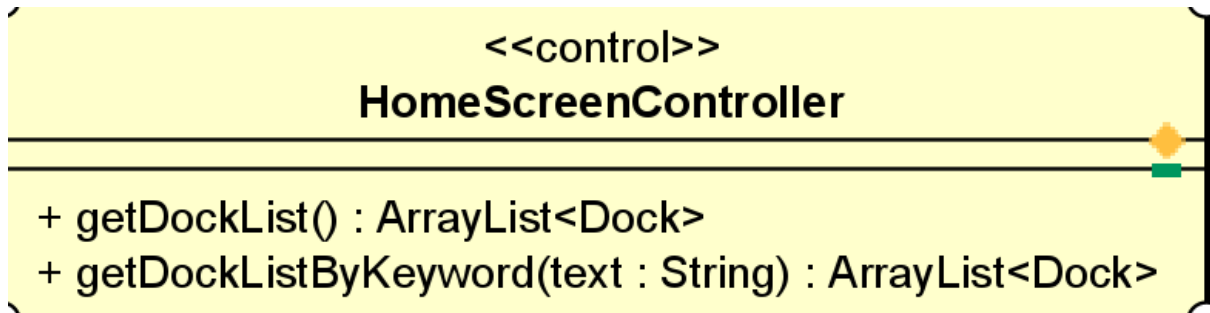   Id – dock id

Exception:

        None

Method

        None

State

None

3.4. Class "HomeScreenController"

Attribute

None

Operation

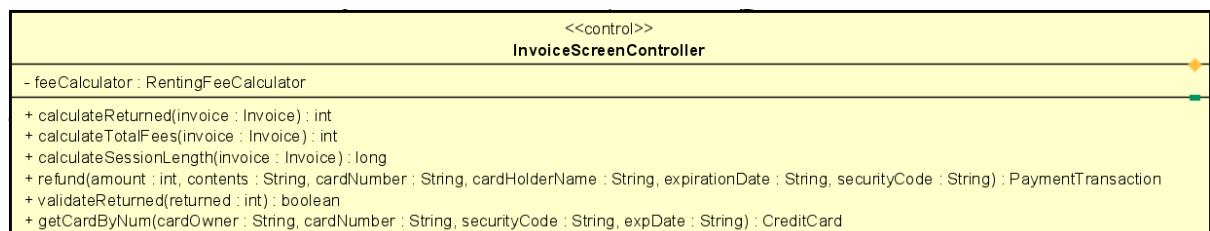| #	Name	Return type	Desciption
1	getDockList	ArrayList	Get dock list
2	getDockListByKeyword	ArrayList	Get dock list by keyword

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	Desciption
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 creditcard info

Parameter:

Invoice

Amount

Contents

CardNumber

CardOwner

ExpirationDate

SecurityCode

Returned

Exception:

PaymentException

NullCardNumberException

NullCardOwnerException

NullExpDateException

NullSecurityCodeException
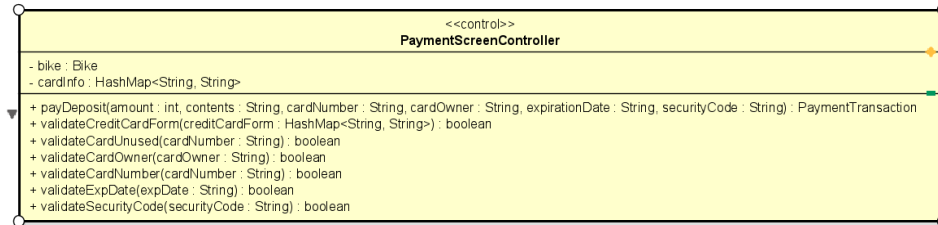
CardExpiredException

Method

None

State

None

3.6. Class "PaymentScreenController"

Attribute

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

Operation

#	Name	Return type	Desciption
1	payDeposit	PaymentTransaction	Set rent transaction
2	validateCreditCardForm	boolean	Check creditcard form
3	validateCardUnused	boolean	Check creditcard used or not
4	validateCardNumber	boolean	Check creditcard number
5	validateCardOwner	boolean	Check creditcard owner
6	validateExpDate	boolean	Check creditcard exp date
7	validateSecurituCode	boolean	Check creditcard security code

Parameter:

    Amount

    Contents

    CardNumber

    CardOwner

    ExpirationDate

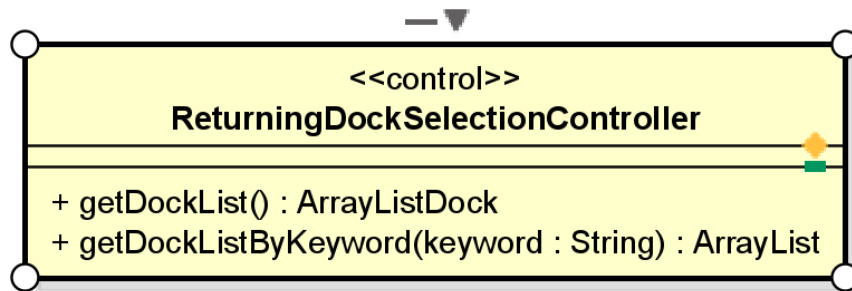   SecurityCode

    creditCardForm

Exception:

    None

Method

None

State

None

3.7. Class "ReturningDockSelectionController"

Attribute

None

Operation

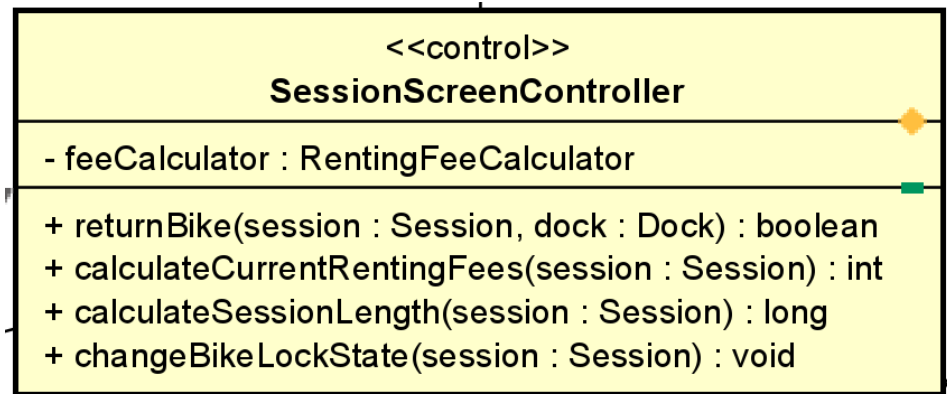| #	Name	Return type	Desciption
1	getDockList	ArrayList<Dock>	Get dock list
2	getDockListByKeyWord	ArrayList<Dock>	Get dock list by keyword

Parameter:

keyword

Exception:

None

Method

None

State

None

3.8. Class "SessionScreenController"

```
                <<control>>
          SessionScreenController
─────────────────────────────────────────
 - feeCalculator : RentingFeeCalculator
─────────────────────────────────────────
 + returnBike(session : Session, dock : Dock) : boolean
 + calculateCurrentRentingFees(session : Session) : int
 + calculateSessionLength(session : Session) : long
 + changeBikeLockState(session : Session) : void
```

Attribute

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

Operation

#	Name	Return type	Desciption
1	returnBike	boolean	Check if bike is available
2	calculateCurrenRentingFees	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

4. Class Diagram