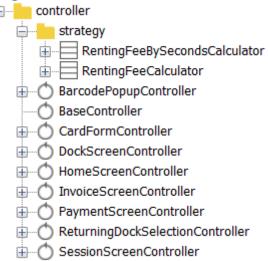
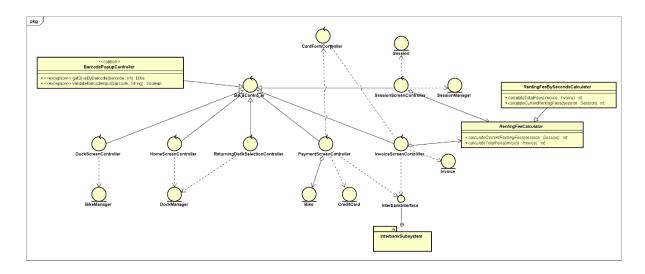
Class Design - Controller

1. Create Initial Design Classes

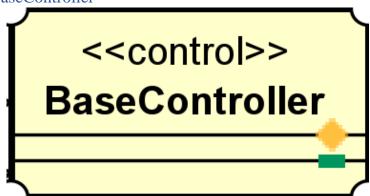


2. Define Relationships Between Classes



3. Class Design

3.1. Class "BaseController"



Attribute

None

Operation

None

Parameter:

None

Exception:

None

Method

None

State

None

3.2. Class "BarcodePopupController"

<<control>> BarcodePopupController

- + <<exception>> getBikeByBarcode(barcode : int) : Bike
- + <<exception>> validateBarcodeInput(barcode : String) : boolean

Attribute

None

Operation

#	# Name i		Desciption
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

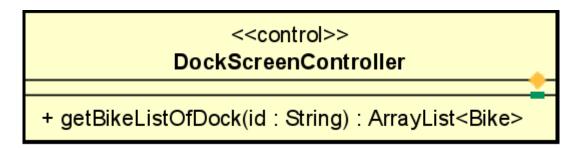
Method

None

State

None

3.3. Class "DockScreenController"



Attribute

None

Operation

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

Parameter:

Id - dock id

Exception:

None

Method

None

State

None

3.4. Class "HomeScreenController"

<<control>> **HomeScreenController**

- + getDockList(): ArrayListDock
- + getDockListByKeyword(text : String) : ArrayList

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"

	< <control>> InvoiceScreenController</control>
- feeCalculator : Re	entingFeeCalculator

- + calculateReturned(invoice : Invoice) : int
- + calculateTotalFees(invoice : Invoice) : int + calculateSessionLength(invoice : Invoice) : long

- + calculateSessionLength(invoice : Invoice) : long
 + validateReturned(returned : int) : boolean
 + getCardByNum(cardOwner : String, cardNumber : String, securityCode : String, expDate : String) : CreditCard
 + <<exception>> refund(amount : int, contents : String, cardNumber : String, cardHolderName : String, expirationDate : String, securityCode : String) : PaymentTransaction
 + <<exception>> pay(amount : int, contents : String, cardNumber : String, cardHolderName : String, expirationDate : String, securityCode : String) : PaymentTransaction
 + <<exception>> make Transaction (amount : int, card : CreditCard) : PaymentTransaction
 + returnBikeProcessing(invoice : Invoice, returnTransaction : PaymentTransaction) : void
 + <<exception>> validateCardForm(cardInfo : HashMap<String>) : void

Attribute

#	Name	Data type	Default value	Descriptio
				n

1	feeCalculato	RentingFeeCalculato	new	calculate
	r	r	RentingFeeBySecondsCalculato	fee
			r	

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 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

3.6. Class "PaymentScreenController"

< <control>> PaymentScreenController</control>
- bike : Bike - cardInfo : HashMap <string, 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>)</string,string></string,>

Attribute

#	Name	Data type	Default value	Description
1	bike	Bike	NULL	bike
2	cardInfo	HashMap <string, 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

Parameter:

Amount

Contents

CardNumber

CreditCardForm

CardInfo

Bike

Exception:

None

Method

None

State

3.7. Class "ReturningDockSelectionController"

<<control>> ReturningDockSelectionController

- + getDockList() : ArrayListDock
- + getDockListByKeyword(keyword: String): ArrayList
- + createInvoice(id : String) : Invoice
- + returnBikeToDock(bike : Bike, dock : Dock) : void

Attribute

None

Operation

#	Name	Return type	Desciption
1	getDockList	ArrayList <dock></dock>	Get dock list
2	getDockListByKeyWord	ArrayList <dock></dock>	Get dock list by keyword
3	createInvoice	Invoice	Create new invoice
4	returnBikeToDock	void	Return bike to the chosen dock

Parameter:

Keyword

ld

Bike

Dock

Exception:

None

Method

None

State

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	Descriptio
				n
1	feeCalculato	RentingFeeCalculato	new	calculate
	r	r	RentingFeeBySecondsCalculato	fee
			r	

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

Pa	rai	m	۵t	۵	r.
ıa	ıaı	111	υL	v	ι.

Session

Dock

Exception:

None

Method

State

None

3.9. Class "CardFormController"

<<control>>

CardFormController

- + <<exception>> validateCreditCardForm(creditCardForm : HashMap) : void
- + <<boundary>> validateCardNumber(cardNumber : String) : void
- + <<exception>> validateCardOwner(cardOwner : String) : void
- + <<exception>> validateSecurityCode(securityCode : String) : void
- + <<exception>> validateExpDate(expDate : String) : void

Attribute

None

Operation

#	Name	Return type	Desciption
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

3.10. Class "RentingFeeBySecondsCalculator"

RentingFeeBySecondsCalculator



- + calculateTotalFees(invoice : Invoice) : int
- + calculateCurrentRentingFees(session: Session): int

Attribute

None

Operation

#	Name	Return type	Desciption
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"

RentingFeeCalculator



- + calculateCurrentRentingFees(session : Session) : int
- + calculateTotalFees(invoice : Invoice) : int

Attribute

Operation

#	Name	Return type	Desciption
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

