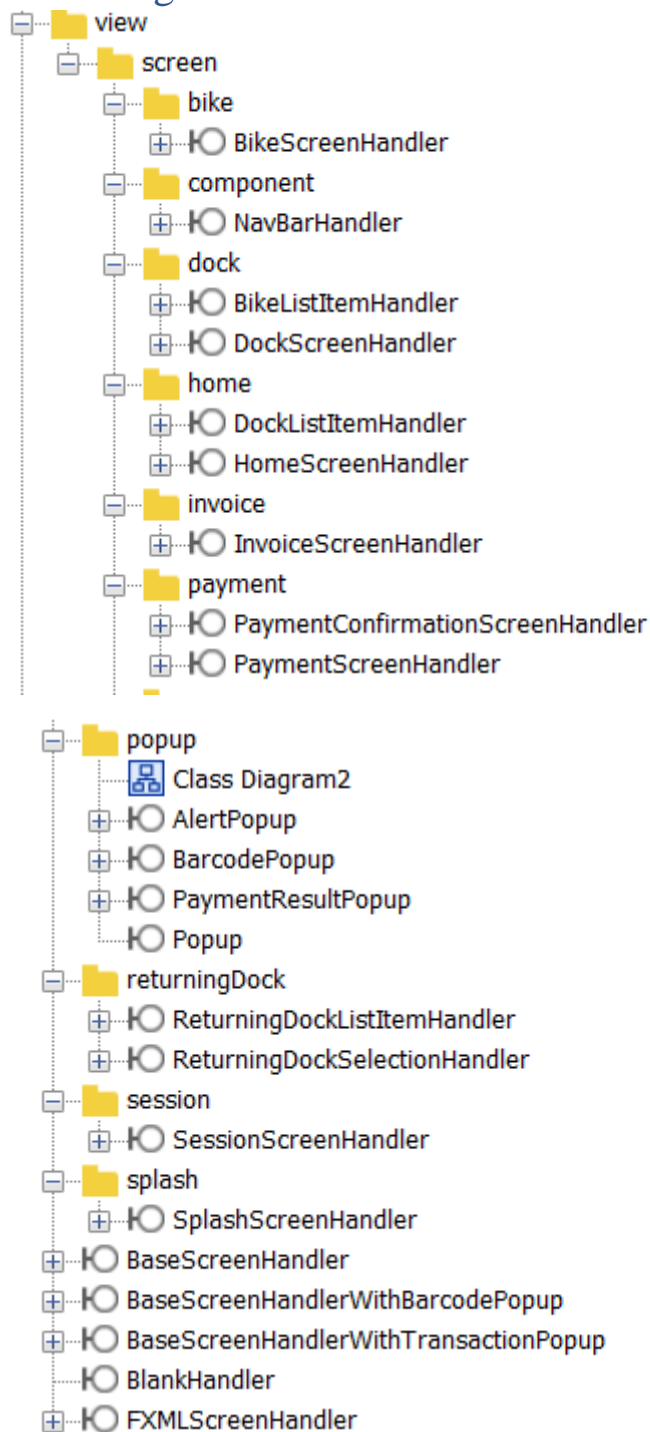
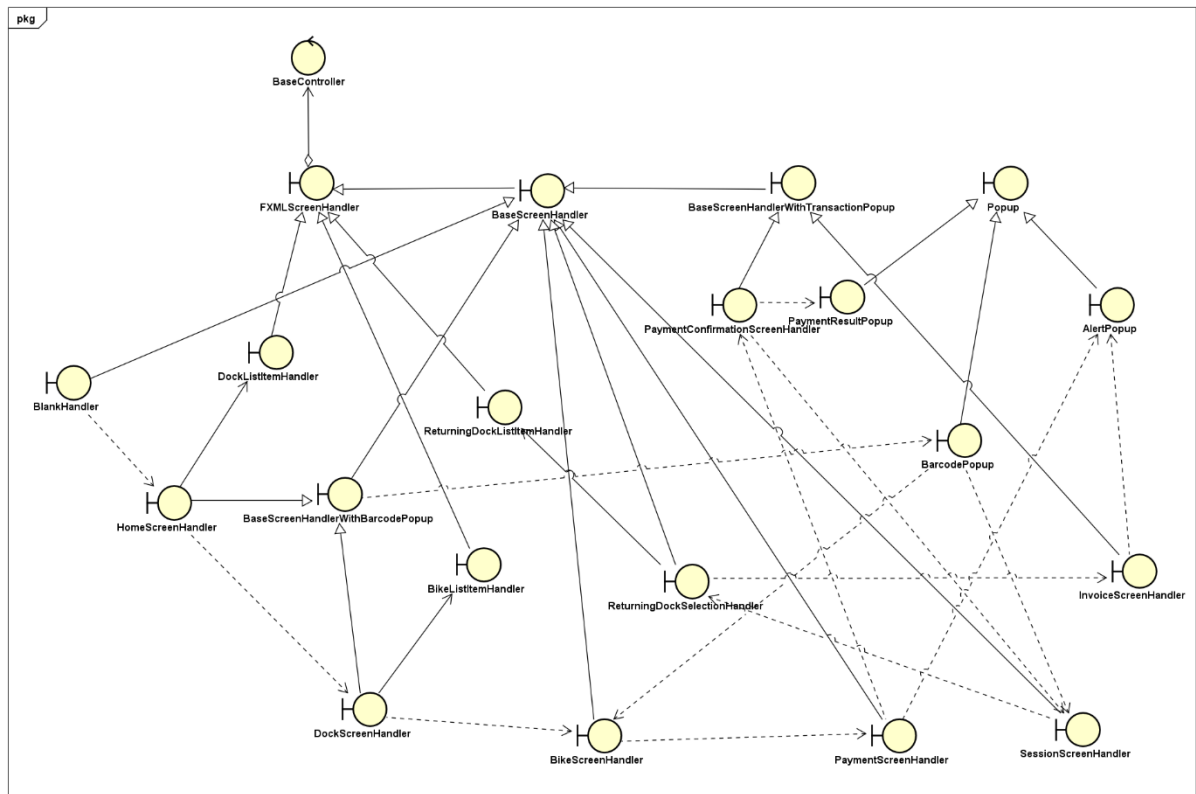


Class Design - View

1. Create Initial Design Classes

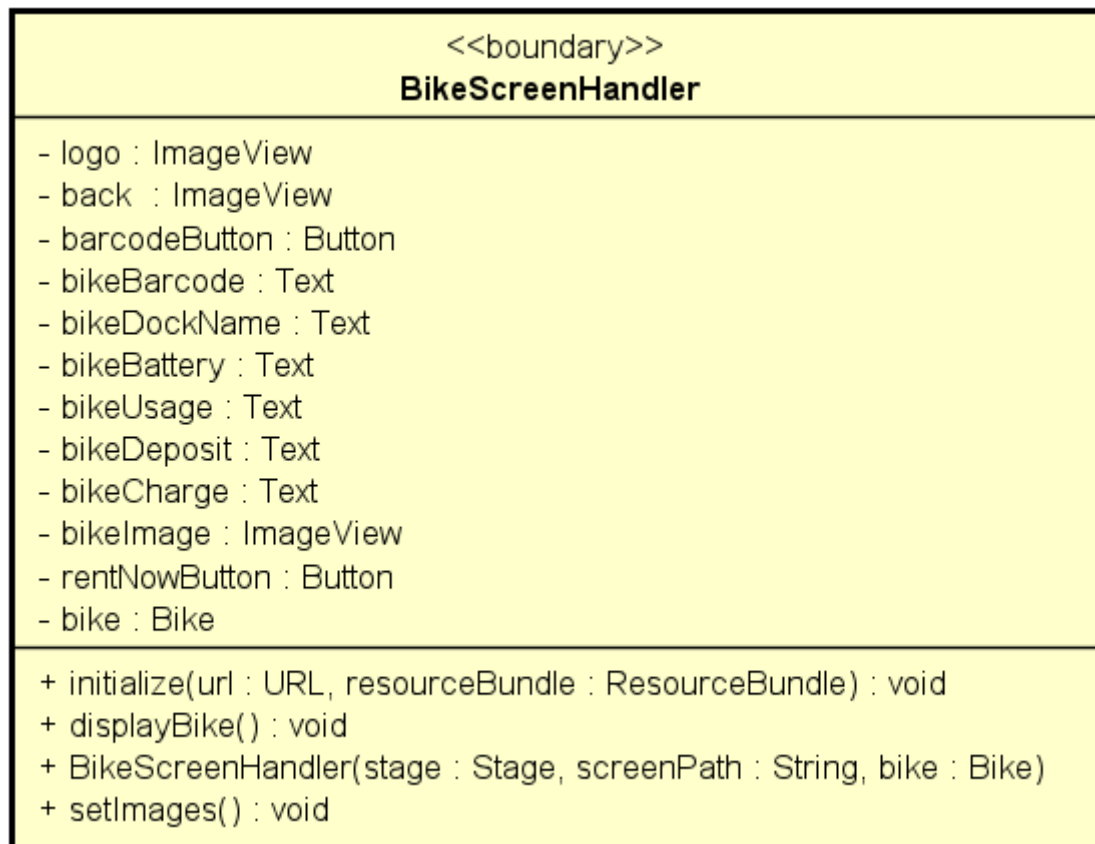


2. Define Relationships Between Classes



3. Class Design

3.1. Class “BikeScreenHandler”



Attribute

#	Name	Data type	Default value	Description
1	logo	ImageView	NULL	Logo image fxml
2	back	imageView	NULL	Back image fxml
3	barcodeButton	Button	NULL	BarcodeButton fxml
4	bikeBarcode	Text	NULL	Bike barcode fxml
5	bikeDockName	Text	NULL	Dock name fxml
6	bikeBattery	Text	NULL	Bike battery fxml
8	bikeUsage	Text	NULL	Bike usage fxml
9	bikeDeposit	Text	NULL	Bike deposit fxml
10	bikeCharge	Text	NULL	Bike charge fxml
11	bikeImage	ImageView	NULL	Bike image
12	rentNowButton	Button	NULL	Rent button fxml
13	bike	Bike	NULL	bike

Operation

#	Name	Return type	Description
1	initialize	void	initialize

2	displayBike	void	Display bike info screen
3	setImages	void	Set bike image

Parameter:

URL

ResourceBundle

Exception:

None

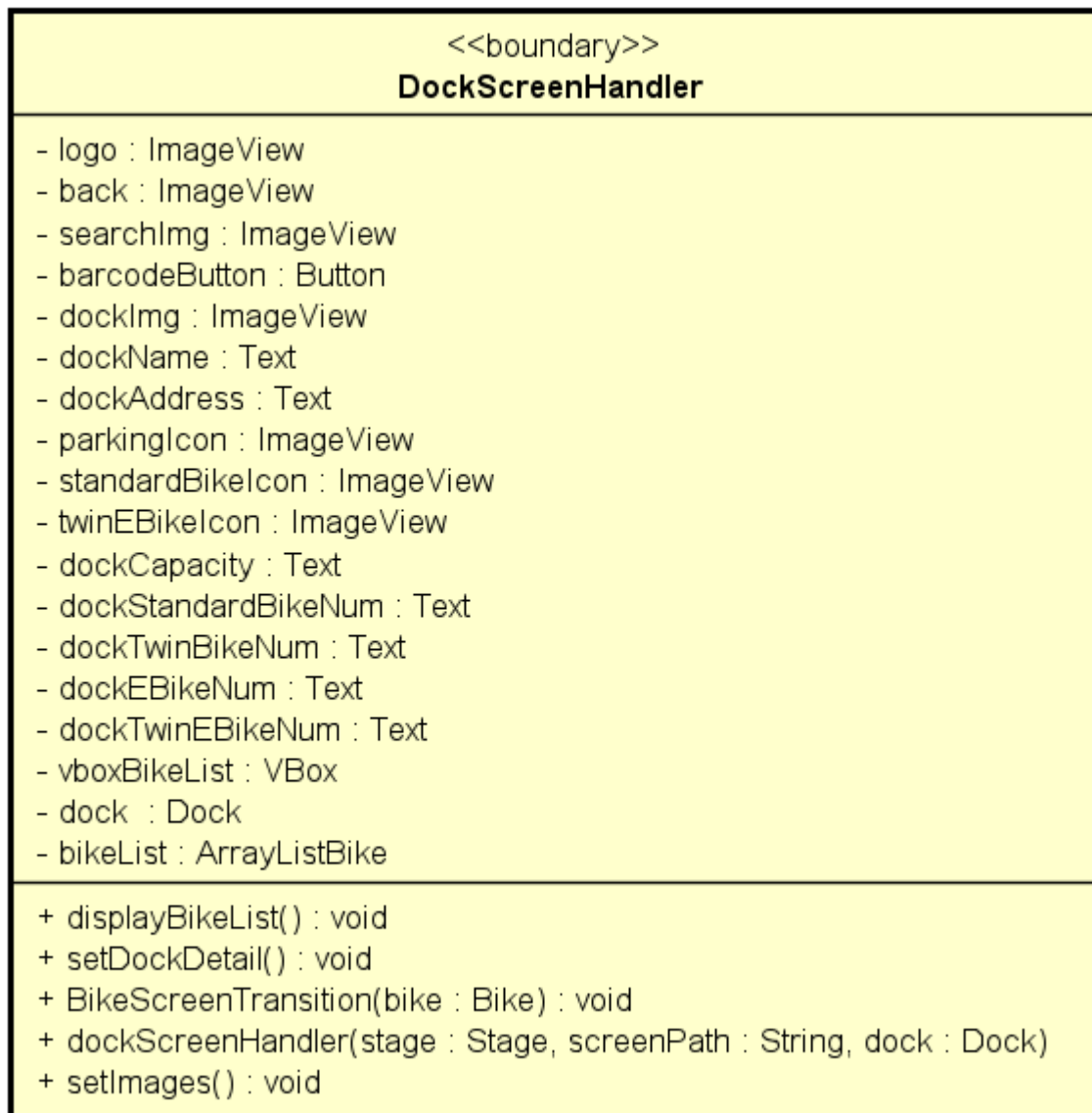
Method

None

State

None

3.2. Class “DockScreenHandler”



Attribute

#	Name	Data type	Default value	Description
1	logo	ImageView	NULL	Logo image fxml
2	back	imageView	NULL	Back image fxml
3	searchImg	imageView	NULL	Search image fxml
4	barcodeButton	Button	NULL	BarcodeButton fxml
5	dockName	Text	NULL	Dock name fxml
6	dockAddress	Text	NULL	Dock address fxml
7	dockImg	ImageView	NULL	Dock image fxml
	parkingIcon	ImageView	NULL	Parking icon fxml
	standardBikeIcon	ImageView	NULL	Standard bike icon fxml
	twinEBikeIcon	ImageView	NULL	Twin bike icon fxml
8	dockStandardBikeNum	Text	NULL	Standard bike number fxml

9	dockTwinBikeNum	Text	NULL	Twin bike number fxml
10	dockEBikeNum	Text	NULL	E-bike number fxml
11	dockTwinEBikeNum	Text	NULL	Twin e-bikenumber fxml
12	vboxBikeList	VBox	NULL	Bike list fxml
13	dock	Dock	NULL	dock
14	bikeList	ArrayList<Bike>	NULL	Bike list in dock

Operation

#	Name	Return type	Description
1	displayBikeList	void	Display bike list
2	setDockDetail	void	Set dock detail
3	BikeScreenTransition	void	Transition to bike screen
4	setImages	void	Set dock image

Parameter:

bike

Exception:

None

Method

None

State

None

3.3. Class “BikeListItemHandler”

<<boundary>> BikeListItemHandler	
- bikeName : Text - bikeSaddle : Text - bikePedal : Text - bikeRearSeat : Text - rentButton : Button - bikeBattery : Text - bikeUsageTime : Text - bikeElecInfo : GridPane - bikeImg : ImageView - bikeIcon : ImageView - batteryIcon : ImageView - clockIcon : ImageView - dockScreenHandler : DockScreenHandler - bike : Bike	
- setBikeInfo() : void + <<exception>> BikeListItemHandler(screenPath : String, dockScreenHandler : DockScreenHandler) - setImages() : void	

Attribute

#	Name	Data type	Default value	Description
1	bikeImg	ImageView	NULL	Bike image fxml
2	bikeIcon	imageView	NULL	Bike image fxml
3	rentButton	Button	NULL	Rent button fxml
4	bikeName	Text	NULL	Bike name fxml
5	bikeSaddle	Text	NULL	Bike saddle fxml
6	bikePedal	Text	NULL	Bike pedal fxml
7	bikeRearSeat	Text	NULL	Bike rear seat fxml
8	dockScreenHandler	DockScreenHandler	NULL	object
9	bike	Bike	NULL	object

Operation

#	Name	Return type	Description
1	setBikeInfo	void	Set bike info
2	setImages	void	Set bike imgae in list

Parameter:

none

Exception:

IOException

Method

None

State

None

3.4. Class “BarcodePopup”

<<boundary>> BarcodePopup	
- parentScreenHandler : BaseScreenHandlerWithBarcodePopup	
+ handleBarcodeEnter() : void - popup(parentScreenHandler : int) : BaseScreenHandlerWithBarcodePopup + display(parentScreenHandler : BaseScreenHandlerWithBarcodePopup) : void - handleBarcodeEnter() : void + BarcodePopup(stage : Stage, parentScreenHandler : BaseScreenHandlerWithBarcodePopup) : void + getBController() : BarcodePopupController	

Attribute

#	Name	Data type	Default value	Description
1	parentScreenHandler	BaseScreenHandlerWithBarcodePopup	NULL	Parent screen

Operation

#	Name	Return type	Description
1	handleBarcodeEnter	void	Handle action “enter barcode”
2	popup	BaseScreenHandlerWithBarcodePopup	Popup
3	display	void	Display popup
4	handleBarcodeEnter	void	Button enter handler
5	getBController	BarcodePopupController	Get barcode

Parameter:

parentScreenHandler
stage

Exception:

None

Method

None

State

None

3.5. Class “DockListItemHandler”

<<boundary>> DockListItemHandler	
- dockName : Text - dockAddress : Text - dockBikeNum : Text - dockImg : ImageView - dock : Dock	
- setDockInfo() : void + <<exception>> DockListItemHandler(screenPath : String, homeScreen : HomeScreenHandler, dock : Dock)	

Attribute

#	Name	Data type	Default value	Description
1	dockName	Text	NULL	Dock name
2	dockAddress	Text	NULL	Dock address
3	dockBikeNum	Text	NULL	Number of bikes in dock
4	dockImg	ImageView	NULL	Dock image
5	dock	Dock	NULL	dock

Operation

#	Name	Return type	Description
1	setDockInfo	void	Set dock info

Parameter:

none

Exception:

None

Method

None

State

None

3.6. Class “InvoiceScreenHandler”

<<boundary>> InvoiceScreenHandler	
- changeCardState : boolean - invoiceBikelImage : ImageView - invoiceCardNumber : Text - invoiceStartTime : Text - invoiceEndTime : Text - invoiceDeposit : Text - invoiceReturned : Text	
+ <<exception>> InvoiceScreenHandler(stage : Stage, screenPath : String, invoice : Invoice, controller : InvoiceScreenController) : void + handleCheckBox() : void + handleConfirmButton() : void - setTextFields() : void + continueAfterPopupClosed(paymentTransaction : PaymentTransaction) : void	

Attribute

#	Name	Data type	Default value	Description
1	changeCardState	boolean	NULL	Card state
2	invoiceBikelImage	ImageView	NULL	Bike image
3	invoiceCardNumber	Text	NULL	Invoice card number
4	invoiceStartTime	Text	NULL	Invoice start time
5	invoiceEndTime	Text	NULL	Invoice end time
6	invoiceDeposit	Text	NULL	Invoice deposit
7	invoiceReturned	Text	NULL	Invoice returned

Operation

#	Name	Return type	Description
1	handleCheckBox	void	Handle check box
2	handleConfirmButton	void	Handle confirm button
3	setTextFields	void	Set text fields
4	continueAfterPopupClosed	void	Continue program after popup close

Parameter:

paymentTransaction

Exception:

None

Method

None

State

None

3.7. Class “PaymentConfirmationScreenHandler”

<<boundary>> PaymentConfirmationScreenHandler	
<div>+ handleRentingConfirmation() : void</div> <div>+ transitionToSessionScreen(rentTransaction : PaymentTransaction) : void</div> <div>+ PaymentConfirmationScreenHandler(stage : Stage, screenPath : String, controller : PaymentScreenController) : void</div> <div>+ setBikeImage() : void</div> <div>+ setPaymentConfirmationConfirmButtonClickListener(e : MouseEvent) : void</div> <div>+ <<exception>> saveTransaction() : void</div> <div>- setTextLabels() : void</div> <div>+ continueAfterPopupClosed(paymentTransaction : PaymentTransaction) : void</div>	

Attribute

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

Operation

#	Name	Return type	Description
1	handleRentingConfirmation	void	Handle confirmation action
2	transitionToSessionScreen	void	Transition to session screen
3	setBikeImage	void	Set bike image
4	setPaymentConfirmationConfirmButtonClickListener	void	Payment Confirmation Confirm Button Click Listener
5	saveTransaction	void	Save transaction
6	setTextLabels	void	Set text
7	continueAfterPopupClosed	void	Continue program after popup close

Parameter:

rentTransaction

paymentTransaction

mouseEvent

Exception:

IOException

Method

None

State

None

3.8. Class “PaymentScreenHandler”

<<boundary>> PaymentScreenHandler	
<pre>+ handleCardInfoSubmit() : void + gotoConfirmationScreen() : void + PaymentScreenHandler(stage : Stage, screenPath : String, paymentScreenController : PaymentScreenController) - setImages() : void</pre>	

Attribute

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

Operation

#	Name	Return type	Description
1	handleCardInfoSubmit	void	Handle action submit card
2	gotoConfirmationScreen	void	Transition to confirmation screen
3	setImages	vodi	Set payment image

Parameter:

None

Exception:

None

Method

None

State

None

3.9. Class “ReturningDockListSelectionHandler”

<<boundary>> ReturningDockSelectionHandler	
<pre>- dockImg : ImageView - dockAddress : Text - dockEmptySlots : Text - returnBikeBtn : Button - dockInfo : Text - dock : Dock - session : Session - dockList : ArrayListDock</pre>	
<pre>+ displayDockList(dock : Dock) : void + onDockListitemClicked() : void + ReturningDockSelectionHandler(stage : Stage, screenPath : String, returningDockSelectionController : ReturningDockSelectionController, session : Session) + setImages() : void + onDockListitemClicked(dock : Dock) : void + displayDockList() : void + searchImgBtnListener(e : MouseEvent) : void + returnBikeBtnListener(e : MouseEvent) : void</pre>	

Attribute

#	Name	Data type	Default value	Description
1	dockName	Text	NULL	Dock name fxml
2	dockParkingSlot	Text	NULL	Dock parking slot fxml
3	dockImg	ImageView	NULL	Dock image fxml
4	dockInfo	Text	NULL	Dock information fxml
5	dockAdress	Text	NULL	Dock address fxml
6	dock	Dock	NULL	dock
7	session	Session	NULL	Session
8	dockList	ArrayListDock	NULL	Dock list

Operation

#	Name	Return type	Description
1	setDockInfo	void	Set dock info
2	onDockListItemClicked	void	Return dock clicked listener
3	setImages	void	Set return dock image
4	displayDockList	void	Display return dock list
5	searchImgBtnListener	void	Search image button listener
6	returnBikeBtnListener	void	Return bike button listener

Parameter:

None

Exception:

None

Method

None

State

None

3.10. Class “ReturningDockListItemHandler”

<<boundary>> ReturningDockListItemHandler	
- dockName : Text - dockParkingSlot : Text - dockImg : ImageView - dock : Dock - returningDockSelectionHandler : ReturningDockSelectionHandler	
+ ReturningDockListItemHandler(screenPath : String, returningDockSelectionHandler : ReturningDockSelectionHandler) : void + setDockInfo() : void + setDock(dock : Dock) : void + setDockInfo() : void	

Attribute

#	Name	Data type	Default value	Description
1	dockName	Text	NULL	Dock name fxml
2	dockParkingSlot	Text	NULL	Dock address fxml
3	dockImg	ImageView	NULL	Dock image fxml
4	dock	Dock	NULL	dock
5	returningDockSelectionHandler	ReturningDockSelectionHandler	NULL	Returning Dock Selection Handler

Operation

#	Name	Return type	Description
1	setDockInfo	void	Display dock information
2	setDock	void	Display dock

Parameter:

dock

Exception:

None

Method

None

State

None

3.11. Class “SessionScreenHandler”

<<boundary>> SessionScreenHandler	
<ul style="list-style-type: none"> - sessionBikeImg : ImageView - sessionBarcode : Text - sessionLength : Text - sessionRentingFees : Text - sessionCloseBtn : Button - sessionReturnBikeBtn : Button 	
<ul style="list-style-type: none"> + gotoDockSelection() : void - SessionScreenHandler(stage : Stage, screenPath : String, controller : SessionScreenController) : void - setSessionLengthText() : void - ToDayAndTime(seconds : long) : void - setImages() : void - setTextFields() : void - goToDockSelection() : void - setImageForLockBikeImg() : void 	

Attribute

#	Name	Data type	Default value	Description
1	sessionBikeImg	ImageView	NULL	Bike image fxml
2	sessionBarcode	Text	NULL	Session barcode fxml
3	sessionLength	Text	NULL	Session length fxml
4	sessionRentingFees	Text	NULL	Session renting fees fxml
5	sessionCloseBtn	Button	NULL	Button fxml
6	sessionReturnBikeBtn	Button	NULL	Button fxml

Operation

#	Name	Return type	Description
1	gotoDockSeletion	void	Transition to dock selection
2	setSessionLengthText	void	Set session's length text
3	ToDayAndTime	void	Day and time conversion
4	setImages	void	Set image for session
5	setTextFields	void	Set text fields for session
6	goToDockSelection	void	Back to the dock selection
7	setImageForLockBikeImg	void	Set image for bike

Parameter:

seconds

Exception:

None

Method

None

State

None

3.12. Class “HomeScreenHandler”

<<boundary>> HomeScreenHandler	
- vboxDockList : VBox - dockList : ArrayListDock - logo : ImageView - barcodeButton : Button - searchField : TextField	
+ displayDockList() : void + onDockListItemClicked(dock : Dock) : void + moveToSessionScreen(session : Session) : void + moveToBikeViewScreen(bike : Bike) : void + show() : void + <<exception>> HomeScreenHandler(stage : Stage, screenPath : String, homeScreenController : HomeScreenController) : void	

Attribute

#	Name	Data type	Default value	Description
1	vboxDockList	VBox	NULL	Vbox fxml
2	dockList	ArrayListDock	NULL	Dock list
3	logo	ImageView	NULL	Logo fxml
4	barcodeButton	Button	NULL	Barcode button fxml
5	searchField	TextField	NULL	Search field fxml

Operation

#	Name	Return type	Description
1	displayDockList	void	Display dock list
2	onDockListItemClicked	void	Dock list item clicked listener
3	moveToSessionScreen	void	Move to session screen
4	moveToBikeViewScreen	void	Move to bike view screen
5	show	void	Show screen

Parameter:

dock

session

bike

Exception:

None

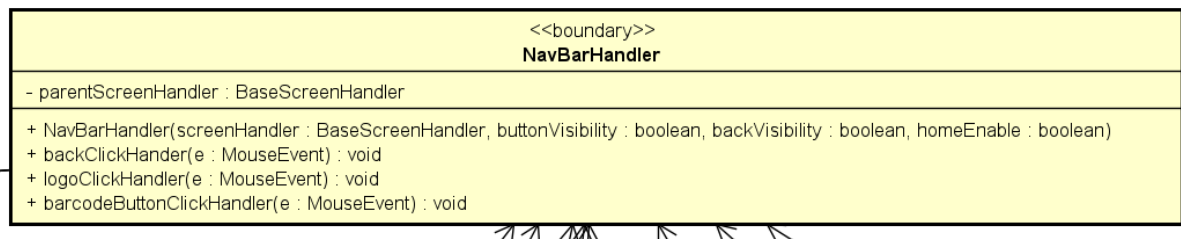
Method

None

State

None

3.13. Class “NavBarHandler”



Attribute

#	Name	Data type	Default value	Description
1	parentScreenHandler	BaseScreenHandler	NULL	handler

Operation

#	Name	Return type	Description
1	backClickHandler	void	go to previous screen
2	logoClickHandler	void	go to home screen
3	barcodeButtonClickHandler	void	bring up barcode popup

Parameter:

e – mouse event

Exception:

None

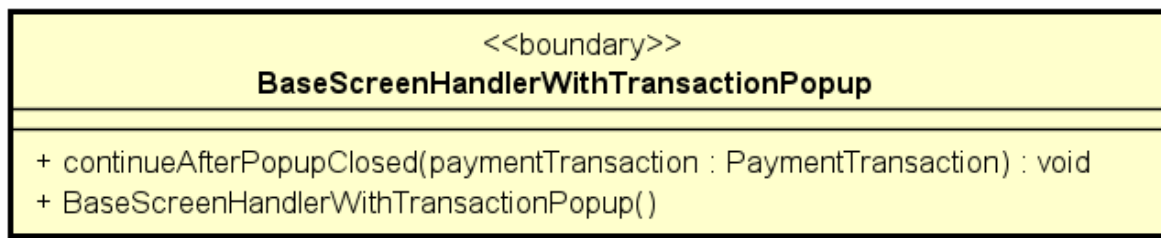
Method

None

State

None

3.14. Class “BaseScreenHandlerWithTransactionPopup”



Attribute

None

Operation

#	Name	Return type	Description
1	continueAfterPopupClosed	void	show then closed popup of payment result

Parameter:

paymentTransaction – transaction to be shown result

Exception:

None

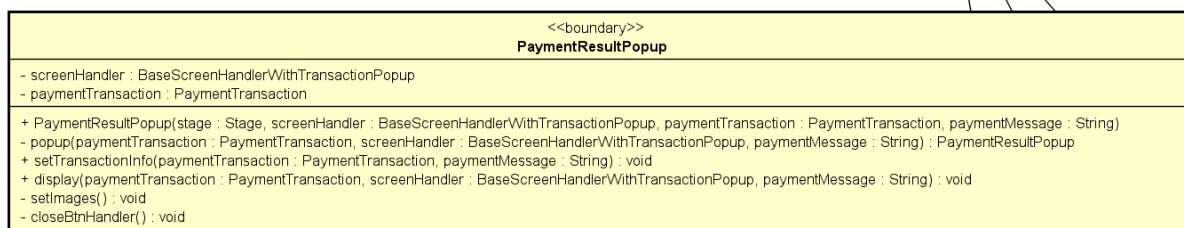
Method

None

State

None

3.15. Class “PaymentResultPopup”



Attribute

#	Name	Data type	Default value	Description
1	screenHandler	BaseScreenHandlerWithTransactionPopup	new	handler

2	paymentTransaction	PaymentTransaction	NULL	transaction to be shown
---	--------------------	--------------------	------	-------------------------

Operation

#	Name	Return type	Description
1	setTransactionInfo	void	Display result
2	display	void	Render result
3	setImage	void	Set image for screen
4	closeBtnHandler	void	Close the popup and move to next screen
5	popup	PaymentResultPopup	Create then return popup

Parameter:

screenHandler

paymentTransaction

paymentMessage

Exception:

None

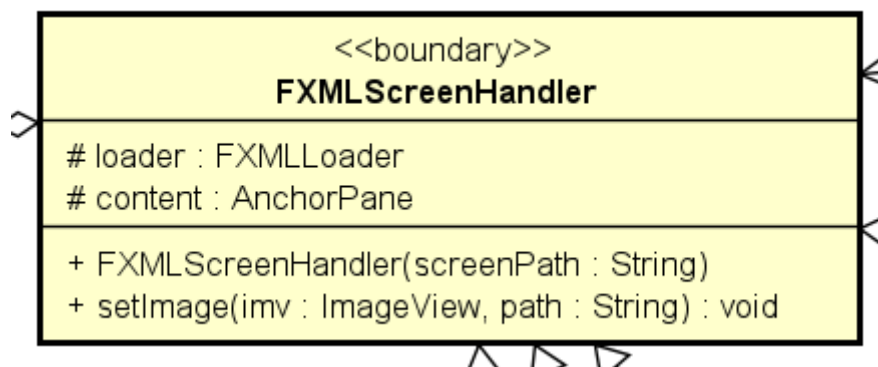
Method

None

State

None

3.16. Class “FXMLScreenHandler”



Attribute

#	Name	Data type	Default value	Description
1	loader	FXMLLoader	new	
2	content	AnchorPane	NULL	main app anchor

Operation

#	Name	Return type	Description
1	setImage	void	set image for screen

Parameter:

imv – image holder to be set

path – path to image

Exception:

None

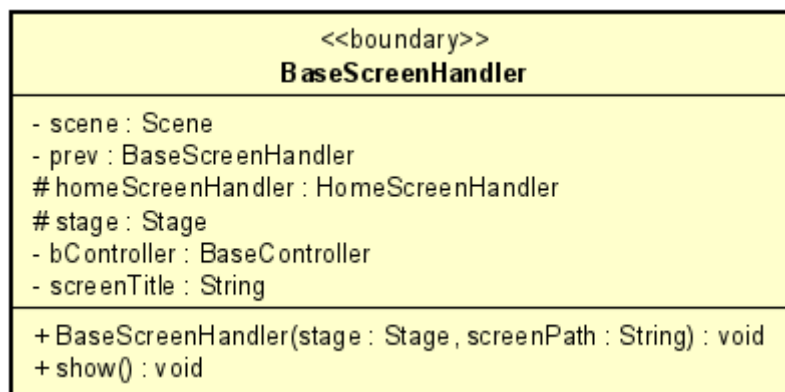
Method

None

State

None

3.17. Class “BaseScreenHandler”



Attribute

#	Name	Data type	Default value	Description
1	scene	Scene	NULL	Scene to render
2	prev	BaseScreenHandler	NULL	Previous screen
3	homeScreenHandler	HomeScreenHandler	NULL	Home screen
4	stage	Stage	NULL	JavaFx stage
5	bController	BaseController	NULL	Controller for this screen
6	screenTitle	String	NULL	Title of this screen

Operation

#	Name	Return type	Description
1	show	void	Render the screen

Parameter:

Exception:

None

Method

None

State

None

4. Class Diagram

