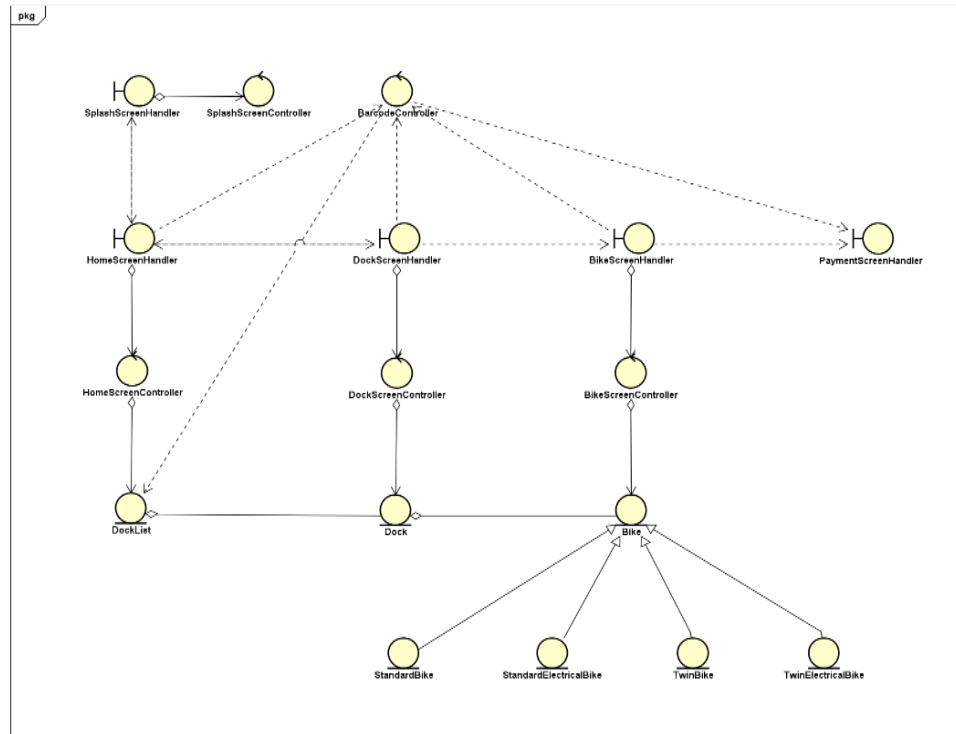


Class Design

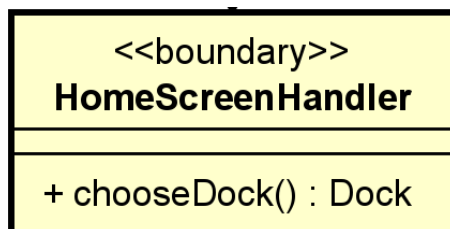
1. CREATE INITIAL DESIGN CLASSES

2. DEFINE RELATIONSHIPS BETWEEN CLASSES



3. CLASS DESIGN

3.1. Class "HomeScreenHandler"



Attribute: None

Operation:

#	Name	Return type	Description
1	chooseDock	Dock	handle dock tab that chosen by user

Parameter:

None

Exception:

None

Method:

None

State:

None

3.2. Class “HomeScreenController”

<<control>> HomeScreenController
- list : DockList - chosenDock : Dock
+ setDockList(list : DockList) : void + submitChosenDock() : void + searchDock(value : String) : void

Attribute:

#	Name	Data Type	Default Value	Description
1	list	DockList	empty ArrayList	List of docks
2	chosenDock	Dock	NULL	Particular dock that chosen by user

Operation

#	Name	Return type	Description
1	setDockList	void	Set the list of docks
2	submitChosenDock	void	Submit the dock chosen by user
3	searchDock	void	Search dock by name or address

Parameter:

list: list of docks

value: the string that user want to search

Exception:

None

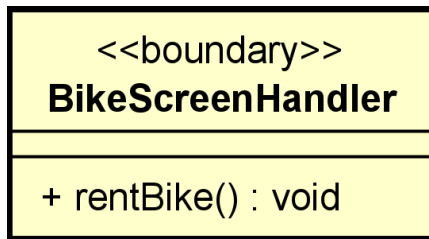
Method:

None

State:

None

3.3. Class “BikeScreenHandler”



Attribute: None

Operation:

#	Name	Return type	Description
1	rentBike	void	Rent bike

Parameter:

None

Exception:

None

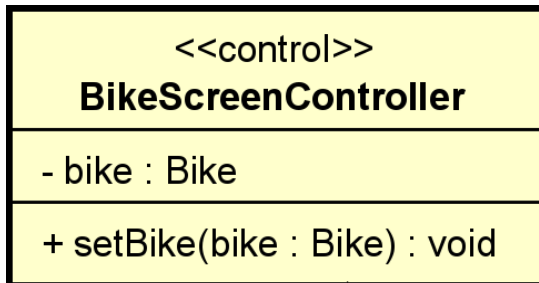
Method:

None

State:

None

3.4. Class “BikeScreenController”



Attribute:

#	Name	Data Type	Default Value	Description
1	bike	Bike	NULL	Bike chosen by user

Operation:

#	Name	Return type	Description
1	setBike	void	Set the chosen bike

Parameter:

bike: Particular Bike

Exception:

None

Method:

None

State:

None

3.5. Class “BikeScreenController”

<<entity>> Bike	
- barcode : String - saddle : int - rearSeat : int - pairOfPedals : int - dock : Dock - deposit : int - charge : int	
+ Bike(barcode : String, saddle : int, rearSeat : int, pairOfPedals : int, dock : Dock, deposit : int, charge : int) : void	

Attribute:

#	Name	Data Type	Default Value	Description
1	barcode	String	NULL	Barcode of the bike
2	saddle	int	1	Number of saddle
3	rearSeat	int	1	Number of rear seat
4	pairOfPedals	int	1	Number of pair of pedals
5	dock	Dock	NULL	Dock where bike is locating
6	deposit	int	NULL	Deposit money of bike
7	charge	int	NULL	Charge of renting

Operation:

#	Name	Return type	Description
1	bike	void	Initialize the info of the bike

Parameter:

bike: Barcode of the bike
saddle: Number of saddle
rearSeat: Number of rear seat
pairOfPedals: Number of pair of pedals
dock: Dock where bike is locating
deposit: Deposit money of bike
charge: Charge of renting

Exception:

None

Method:

None

State:

None

3.6. Class “BikeScreenController”

<<entity>> StandardElectricalBike	
- battery : int - time : String	
+ setBattery(battery : int) : void + setTimeLeft(time : String) : void	

Attribute:

#	Name	Data Type	Default Value	Description
1	battery	int	100%	The percentage of the battery
2	time	String	NULL	Time left before running out of battery

Operation:

#	Name	Return type	Description
1	setBattery	void	Set the percentage of the battery of E-Bike
2	setTimeLeft	void	Set the time left before running out of battery of E-Bike

Parameter:

battery: The percentage of the battery
time: Time left before running out of battery

Exception:

None

Method:

None

State:

None

4. Class Diagram

