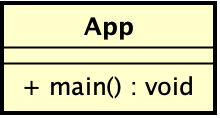
**CLASS DESIGN SPECIFICATION**

1. **App**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | main | void | The main entry point for the application |

* + 1. *Parameter*

None

* + 1. *Exception*

None

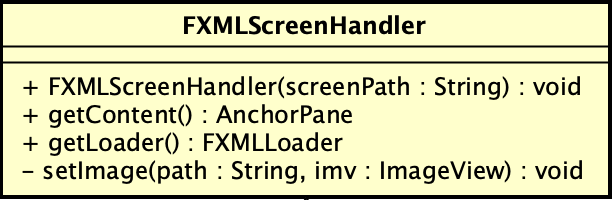
* 1. *Method*

None

* 1. *State*

None

1. **FXMLScreenHandler**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | FXMLScreenHandler | void | Constructs a FXMLScreenHandler from a screenPath |
| 2 | getContent | AnchorPane | Gets the root of the screen |
| 3 | getLoader | FXMLLoader | Gets the loader |

* + 1. *Parameter*
* screenPath: the path to the fxml file of the screen
  + 1. *Exception*

None

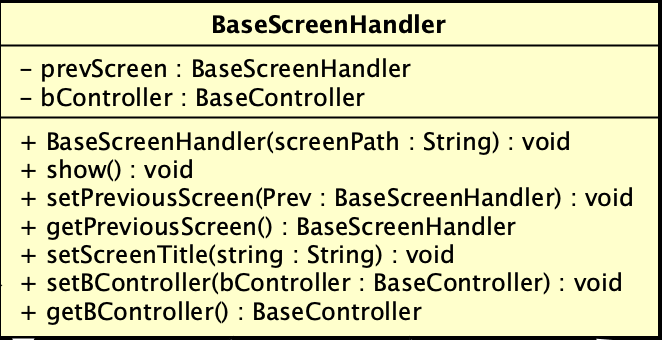
* 1. *Method*

- setImage: Sets image file in *path* to the ImageView *imv*

* 1. *State*

None

1. **BaseScreenHandler**

****

* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description** |
| 1 | prevScreen | BaseScreenHandler |  | Represents the previous screen (in case the user go back) |
| 2 | bController | BaseController |  | Represents the BaseController for the screen |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | BaseScreenHandler | void | Constructs a FXMLScreenHandler from a screenPath |
| 2 | show |  | Shows the screen to the user |

* + 1. *Parameter*
* screenPath: the path to the fxml file of the screen
  + 1. *Exception*

None

* 1. *Method*

- setPreviousScreen: set the screen in BaseScreenHandler *prev* as the previous screen of the current screen

- setPreviousScreen: get the previous screen as a BaseScreenHandler object

- setScreenTitle: set *string* as the title of the screen

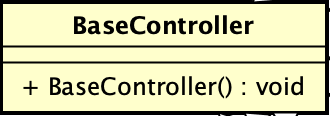
- setBController: set the BaseController for the screen

- getBController: get the BaseController of the screen

* 1. *State*

None

1. **BaseController**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | BaseController | void | Constructs a BaseController |

* + 1. *Parameter*

None

* + 1. *Exception*

None

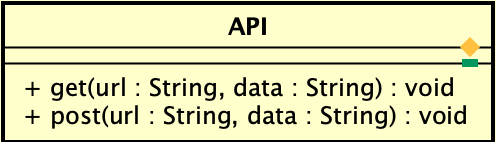
* 1. *Method*

None

* 1. *State*

None

1. **API**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | get | String | Make a HTTP GET request to the given url with the given data |
| 2 | post | String | Make a HTTP POST request to the given url with the given data |

* + 1. *Parameter*
* url: the destination url to send the request
* data: the data of the request
  + 1. *Exception*

None

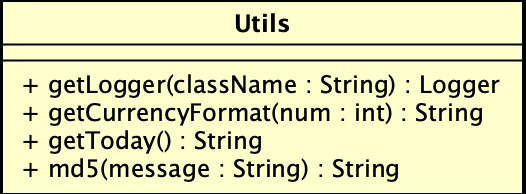
* 1. *Method*

None

* 1. *State*

None

1. **Utils**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | getLogger | Logger | Gets the logger of the class |
| 2 | getCurrencyFormat | String | Return the formatted string according to the currency |
| 3 | getToday | String | Return a string representing the current time in the format yyyy-MM-dd HH:mm:ss |
| 4 | md5 | String | Return the hash value of the message |

* + 1. *Parameter*
* className: the name of the class in string format
* num: the amount of currency
* message: the message in string format
  + 1. *Exception*

None

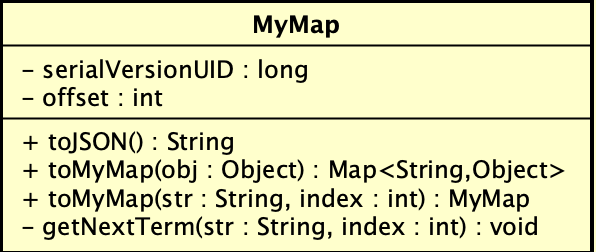
* 1. *Method*

None

* 1. *State*

None

1. **MyMap**

****

* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description** |
| 1 | serialVersionUID | long | 1 | The serial version UID |
| 2 | offset | int | 0 | Keep track of the current index when calling a function |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | toJSON | String | Return a String that represents a JSON object of the MyMap object |
| 2 | toMyMap | Map<String, Object> | Return a Map that represents the mapping among attribute names and their values of the object obj |
| 3 | toMyMap | MyMap | Return a MyMap that represents the interested substring in a String |

* + 1. *Parameter*
* obj: the object of interest
* string: the atring of interest
* idx: the index of the character in the original string where the substring begins
  + 1. *Exception*

None

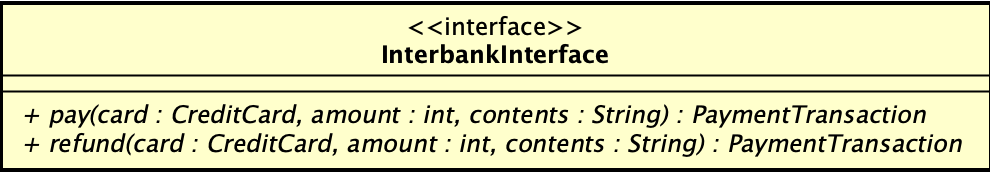
* 1. *Method*

- getNextTerm: Return a string that represents the term in between the double quotes inside a string

* 1. *State*

None

1. **InterbankInterface**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | pay | PaymentTransaction | Pay rent, then return the payment transaction |
| 2 | refund | PaymentTransaction | Refund, then return the payment transaction |

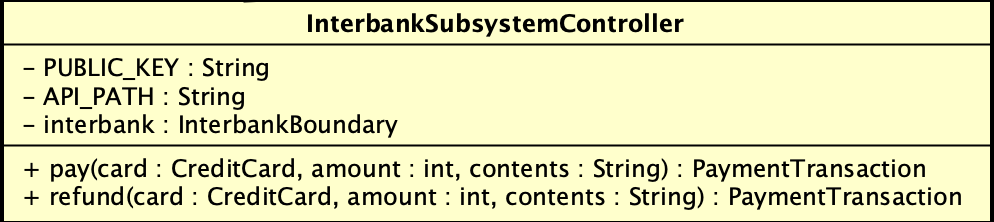
* + 1. *Parameter*
* card: the credit card used for payment/refund
* amount: the amount to pay/refund
* contents: the transaction contents
  + 1. *Exception*
* PaymentException: if responded with a pre-defined error code
* UnrecognizedException: if responded with an unknown error code or something goes wrong
  1. *Method*

None

* 1. *State*

None

1. **InterbankSubsystemController**

****

* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description** |
| 1 | PUBLIC\_KEY | String | supply by subsystem | Being used to encode the data sent to interbank api gateway |
| 2 | API\_PATH | String | supply by subsystem | Path to interbank api gateway |
| 3 | interbank | InterbankBoundary | Injected by IoC | Communicate to the interbank api gateway |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | pay | PaymentTransaction | Pay rent, then return the payment transaction |
| 2 | refund | PaymentTransaction | Refund, then return the payment transaction |

* + 1. *Parameter*
* card: the credit card used for payment/refund
* amount: the amount to pay/refund
* contents: the transaction contents
  + 1. *Exception*
* PaymentException: if responded with a pre-defined error code
* UnrecognizedException: if responded with an unknown error code or something goes wrong
  1. *Method*

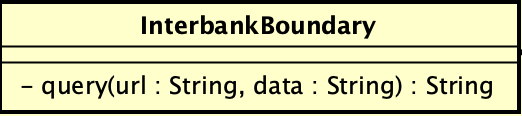
- generateURL: Generate correctly-form URL satisfied the standard interbank system design

- handleResponse: Read the response returned from the interbank by converting JSON-formatted string into separated list of information

* 1. *State*

None

1. **InterbankBoundary**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | query | String | Connect to the Interbank API, make query and receive response |

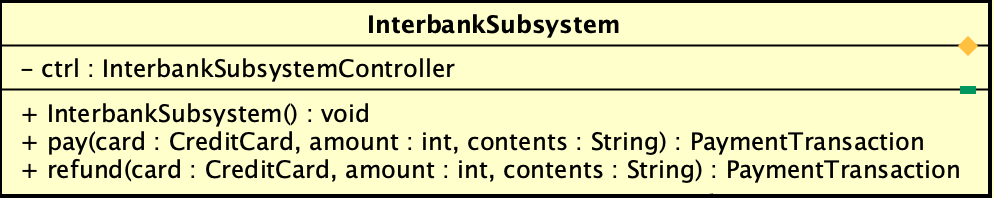
* + 1. *Parameter*
* url: the destination url to send the request
* data: the data of the request
  + 1. *Exception*
* UnrecognizedException: if responded with an unknown error code or something goes wrong
  1. *Method*

None

* 1. *State*

None

1. **InterbankSubsystem**

****

* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description** |
| 1 | ctrl | InterbankSubsystemController |  | Represent the controller of the system |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | InterbankSubsystem |  | Create a new InterbankSubsystem with its InterbankSubsystemController |
| 2 | pay | PaymentTransaction | Pay rent, then return the payment transaction |
| 3 | refund | PaymentTransaction | Refund, then return the payment transaction |

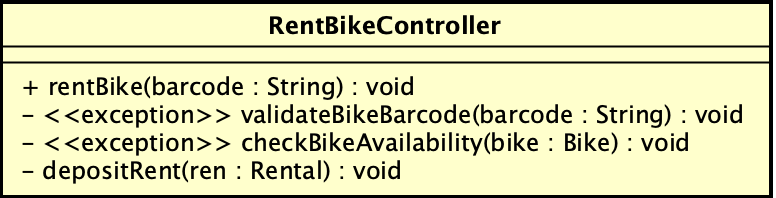
* + 1. *Parameter*
* card: the credit card used for payment/refund
* amount: the amount to pay/refund
* contents: the transaction contents
  + 1. *Exception*
* PaymentException: if responded with a pre-defined error code
* UnrecognizedException: if responded with an unknown error code or something goes wrong
  1. *Method*

None

* 1. *State*

None

1. **RentBikeController**



* 1. *Attribute*

None

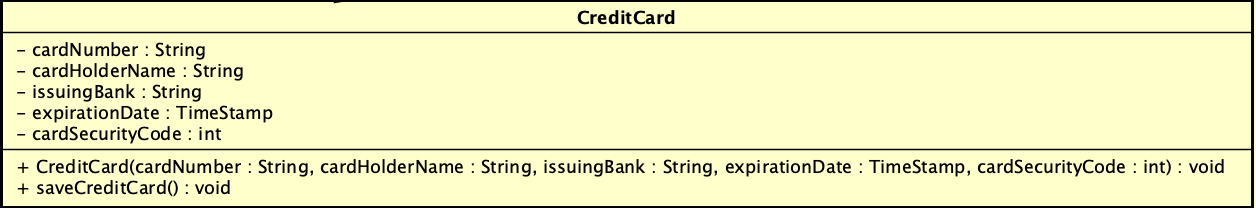
* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | rentBike | Rental | Rent bike, and then return the corresponding rental |

* + 1. *Parameter:*
* barcode: the barcode of the bike
* rental: the rental information
  + 1. *Exception:*
* InvalidBarcodeException: if the input barcode is not valid
* UnavailableBikeException: if the bike is not available for renting
  1. *Method*
* validateBikeBarcode: validate the bike barcode from input
* checkBikeAvailability: check whether the bike is available for renting
* depositRent: deposit the initial renting fee
  1. *State*

None

1. **CreditCard**

****

* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description (purpose)** |
| 1 | cardNumber | String |  | Represents the credit card number |
| 2 | cardHolderName | String |  | Represents the credit card holder’s name |
| 3 | issuingBank | String |  | Represent the code of the issuing bank |
| 4 | expirationDate | Timestamp |  | Represents the date that the credit card is expired |
| 5 | cardSecurityCode | int |  | Represent the security code of the card |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | CreditCard | Invoice | Create credit card instance |
| 2 | saveCreditCard |  | Save the card information |

* + 1. *Parameter*
* cardNumber
* cardHolderName
* issuingBank
* expirationDate
* cardSecurityCode
  + 1. *Exception*

None

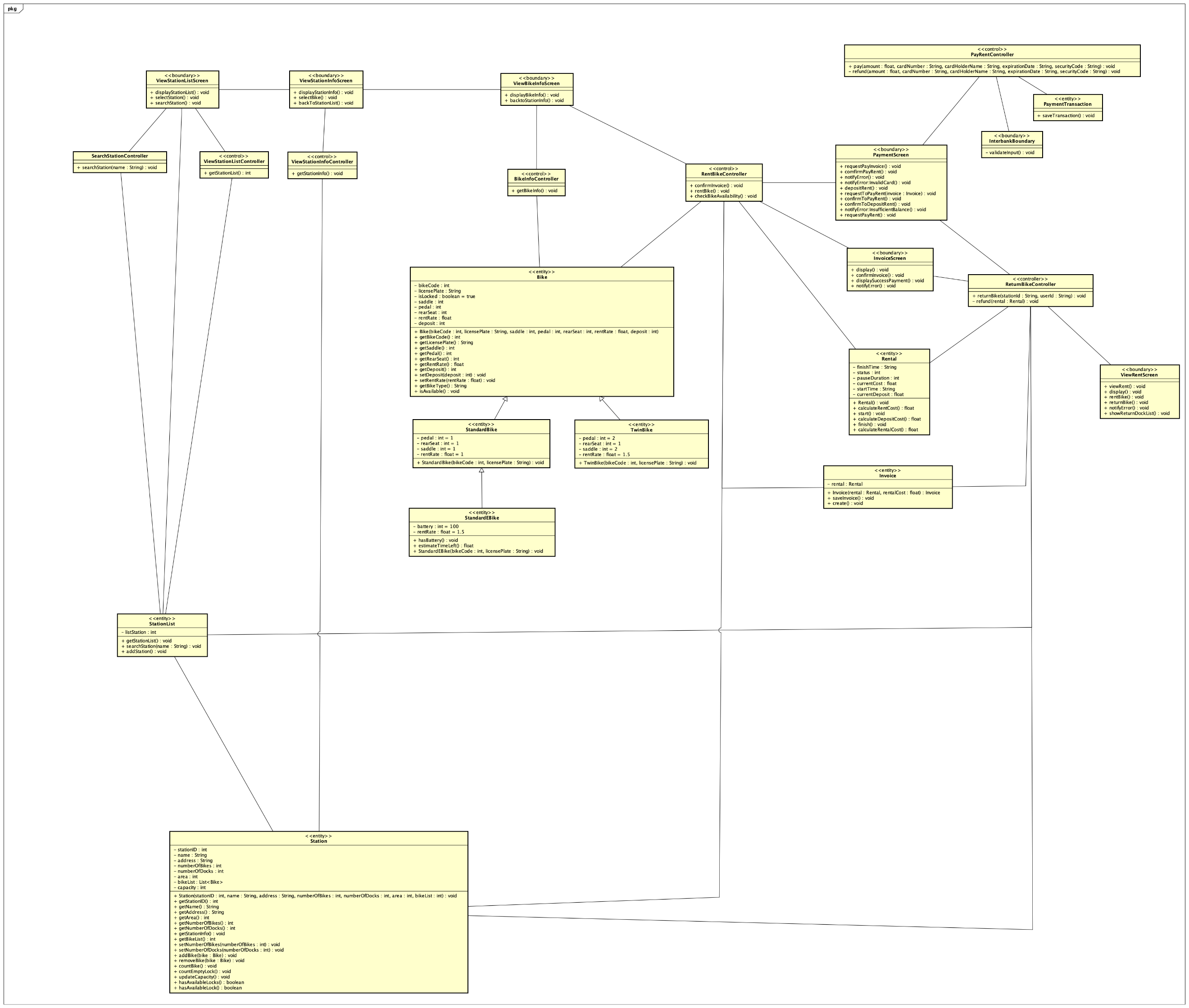
* 1. *Method*

None

* 1. *State*

None

1. **Invoice**

****

* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description (purpose)** |
| 1 | rental | Rental | null | Rental information linked to this invoice |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | Invoice | Invoice | Create invoice instance |
| 2 | saveInvoice | void | Save the invoice information |
| 3 | create | void | Create invoice instance with default value |

* + 1. *Parameter*
* rental: rental linked to invoice instance
* rentalCost: cost of rental to add to invoice
  + 1. *Exception*

None

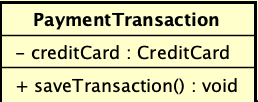
* 1. *Method*

None

* 1. *State*

None

1. **PaymentTransaction**

****

* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description (purpose)** |
| 1 | creditCard | CreditCard | null | Represent the credit card used in this PaymentTransation |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | saveTransaction | void | Save the transaction |

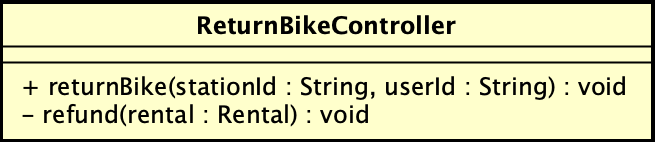
* 1. *Method’*

None

* 1. *State*

None

1. **Return bike controller**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | refund | void | Confirm payment |
| 2 | renturnBike | void | Proceed rent bike |

* + 1. *Parameter*

None

* + 1. *Exception*

None

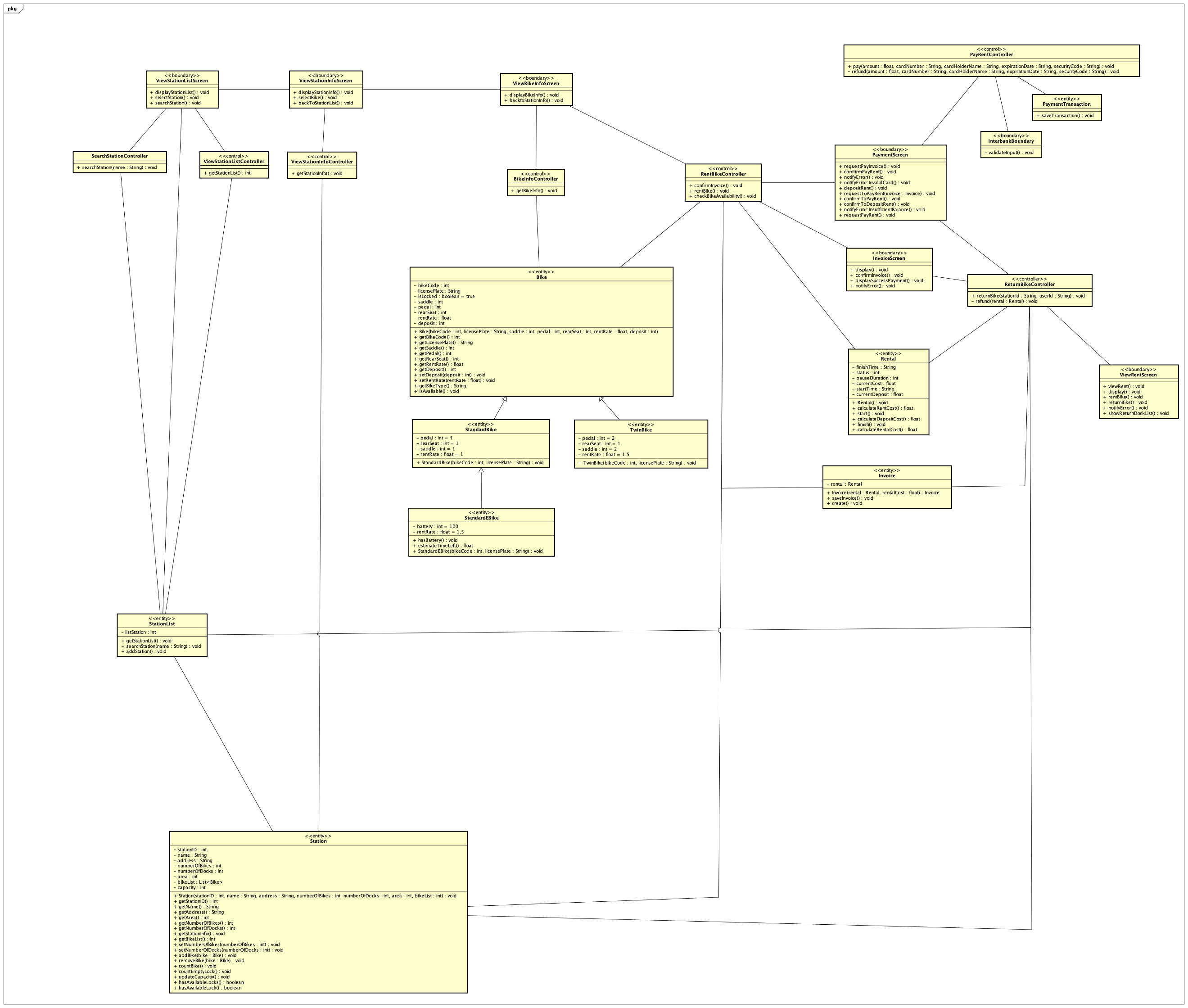
* 1. *Method*

None

* 1. *State*

None

1. **Invoice Screen**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | display | void | Display invoice page UI |
| 2 | confirmInvoice | void | Confirm selected invoice |
| 3 | displaySuccessPayment | void | Display payment information after invoice is proceeded |
| 4 | notifyError | void | Notify error when proceeding |

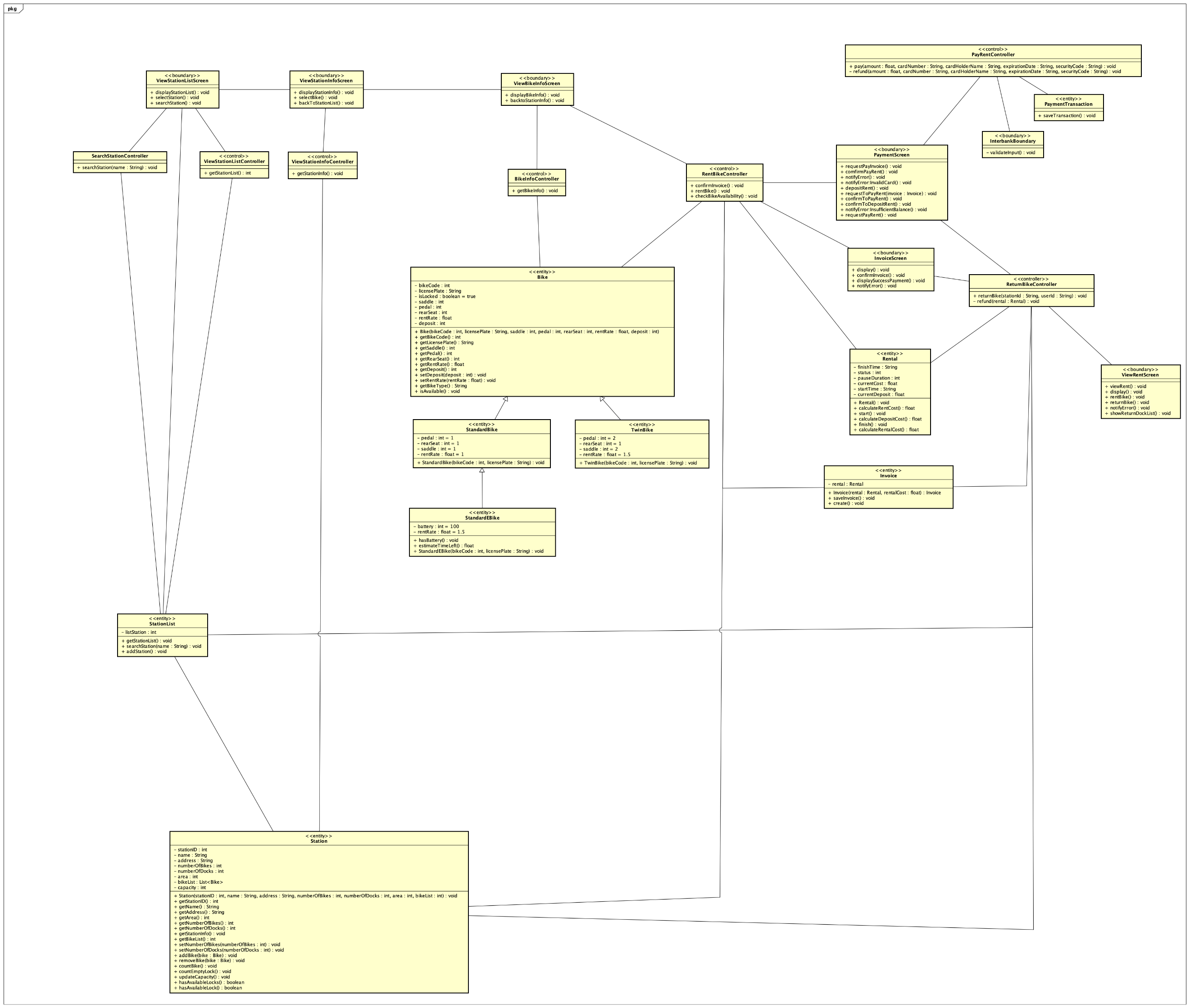
* 1. *Method*

None

* 1. *State*

None

1. **View Rent Screen**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | viewRent | void | Display rent information |
| 2 | display | void | Display view rent screen UI |
| 3 | returnBike | void | Return currently renting bike |
| 4 | rentBike | void | Rent selected bike |
| 5 | notifyError | void | Notify error when processing rent or return bike |
| 6 | showReturnDockList | void | Display available dock to return bike |

* + 1. *Parameter*
    2. *Exception*

None

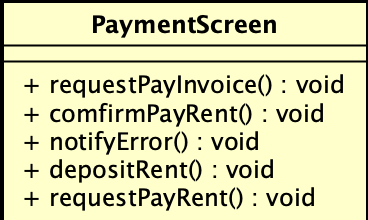
* 1. *Method*

None

* 1. *State*

None

1. **Payment Screen**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | requestPayInvoice | void | Start paying invoice processing |
| 2 | confirmPayRent | void | Process pay rent with current rent information |
| 3 | notifyError | void | Display error from processing |
| 4 | depositRent | void | Deposit an amount from user account |
| 5 | requestPayRent | void | Start processing current info to display rent |

* + 1. *Parameter*

None

* + 1. *Exception*

None

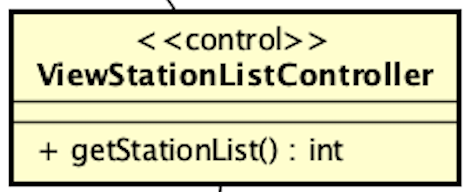
* 1. *Method*

None

* 1. *State*

None

1. **ViewStationListController**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | getStationList | void | Get list of stations |

* + 1. *Parameter*

None

* + 1. *Exception*

None

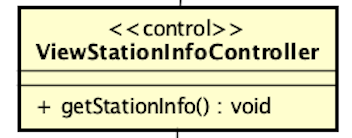
* 1. *Method*

None

* 1. *State*

None

1. **ViewStationInfoController**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | getStationInfo | void | Get info of the current station |

* + 1. *Parameter*

None

* + 1. *Exception*

None

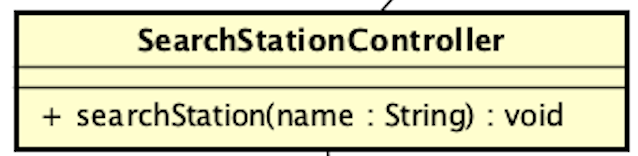
* 1. *Method*

None

* 1. *State*

None

1. **SearchStationController**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | searchStation | void | Search for a station by its name |

* + 1. *Parameter*

name - String: name of the station you want to search for

* + 1. *Exception*

None

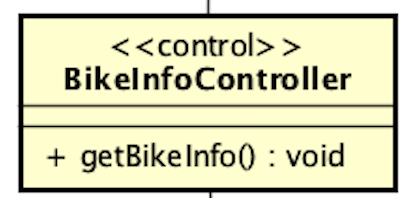
* 1. *Method*

None

* 1. *State*

None

1. **ViewBikeInfoController**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | getBikeInfo | void | Get info of the current bike |

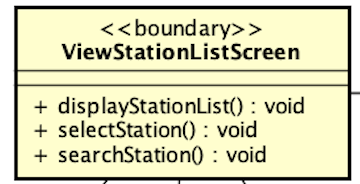
* 1. *Method*

None

* 1. *State*

None

1. **ViewStationListScreen**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | selectStation | void | Select a station from the list |
| 2 | searchStation | void | Call SearchStationController to search station by name or address |
| 3 | displayStationList | void | Display passed list of stations |

* + 1. *Parameter*

None

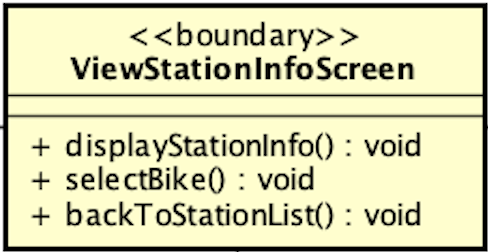
* + 1. *Exception*

None

* 1. *Method*
     + - selectStation: Get station from user selection and call getStationInfo() in ViewStationInfoController to display station info in ViewStationInfoScreen
       - searchStation: Input a name and search for a station using the searchStation() in SearchStationController
  2. *State*

None

1. **ViewStationInfoScreen**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | displayStationInfo | void | Display the info of the current station |
| 2 | selectBike | void | Select a bike from the list |
| 3 | backToStationList | void | Back to the previous station list screen |

* + 1. *Parameter*

None

* + 1. *Exception*

None

* 1. *Method*

selectBike: Get bike from user selection and call getBikeInfo() in ViewBikeInfoController to display bike info in ViewBikeInfoScreen

* 1. *State*

None

1. **ViewBikeInfoScreen**

****

* 1. *Attribute*

None

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | displayBikeInfo | void | Display the information of the current bike |
| 2 | backToStationInfo | void | Return to the previous station info screen |

* + 1. *Parameter*

None

* + 1. *Exception*

None

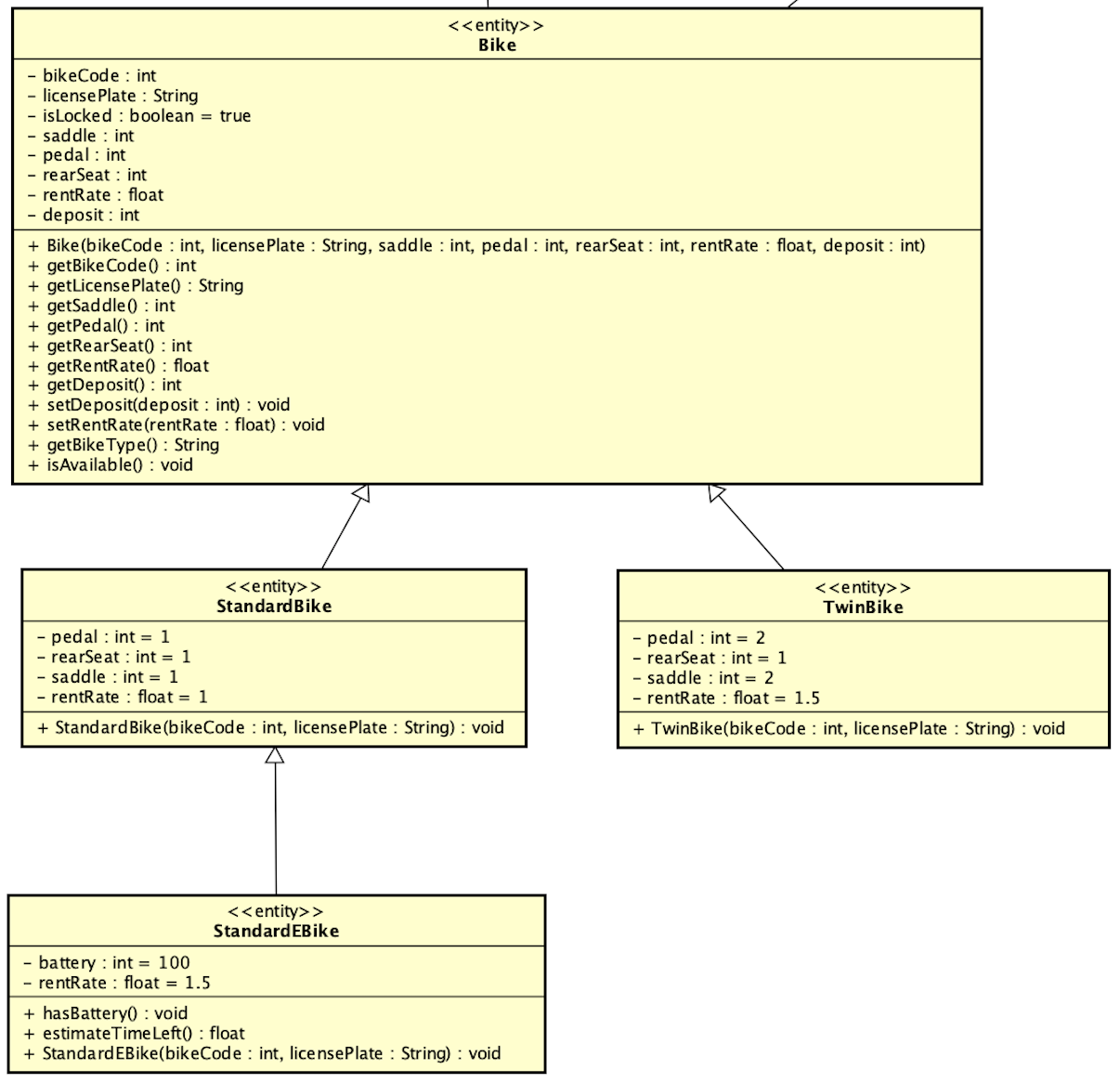
* 1. *Method*

None

* 1. *State*

None

1. **Bike**



* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description (purpose)** |
| 1 | bikeCode | int | NULL | Code of the bike in the system |
| 2 | licensePlate | String | NULL | License plate of the bike |
| 3 | isLocked | boolean | True | Lock condition of the bike |
| 4 | saddle | int | NULL | Number of saddles |
| 5 | pedal | int | NULL | Number of pedals |
| 6 | rearSeat | int | NULL | Number of rear seats |
| 7 | rentRate | float | NULL | Rate of the rent price |
| 8 | deposit | int | NULL | The deposit amount needed to rent the bike |
| 9 | battery | int | 100 | Percentage of battery left |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | getBikeCode | int | Get the code of the bike |
| 2 | getLicensePlate | String | Get the license plate of the bike |
| 3 | getSaddle | int | Get number of saddles of the bike |
| 4 | getPedal | int | Get number of pedals of the bike |
| 5 | getRearSeat | int | Get number of rear seats of the bike |
| 6 | getRentRate | float | Get rentRate of the bike |
| 7 | getDeposit | int | Get deposit of the bike |
| 8 | setDeposit | void | Set the deposit of the bike |
| 9 | setRentRate | void | Set the rentRate of the bike |
| 10 | getBikeType | String | Get the type of the bike |
| 11 | isAvailable | boolean | Check the available condition of the bike |
| 12 | getBattery | int | Get the percentage of battery left |
| 13 | estimateTimeLeft | int | Get the number of minutes left the electric bike can run |

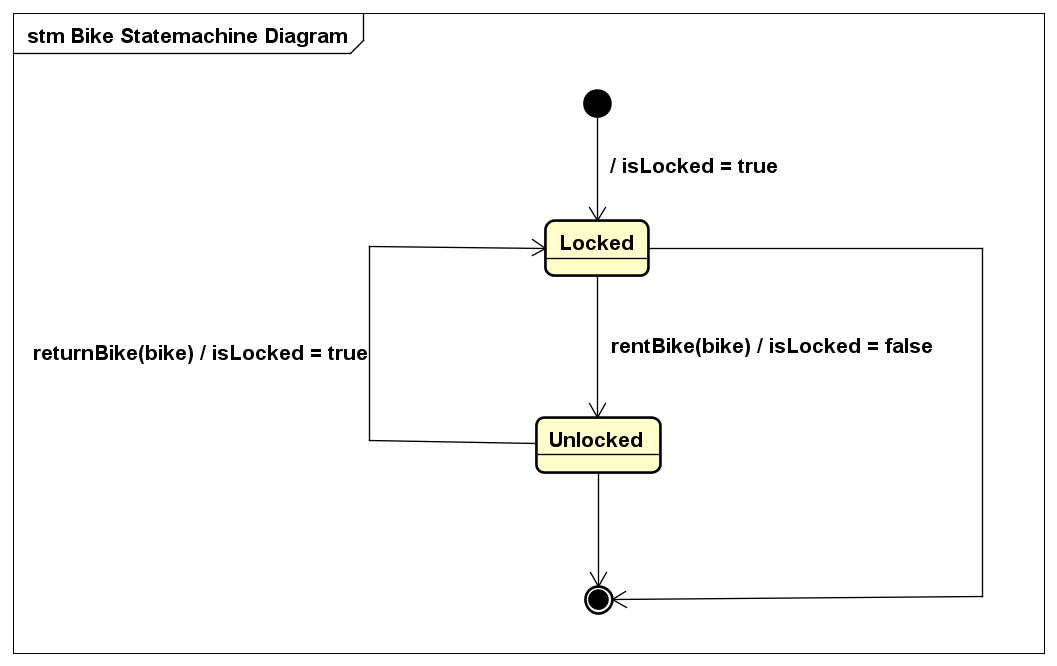
* + 1. *Parameter*
* None
  + 1. *Exception*

None

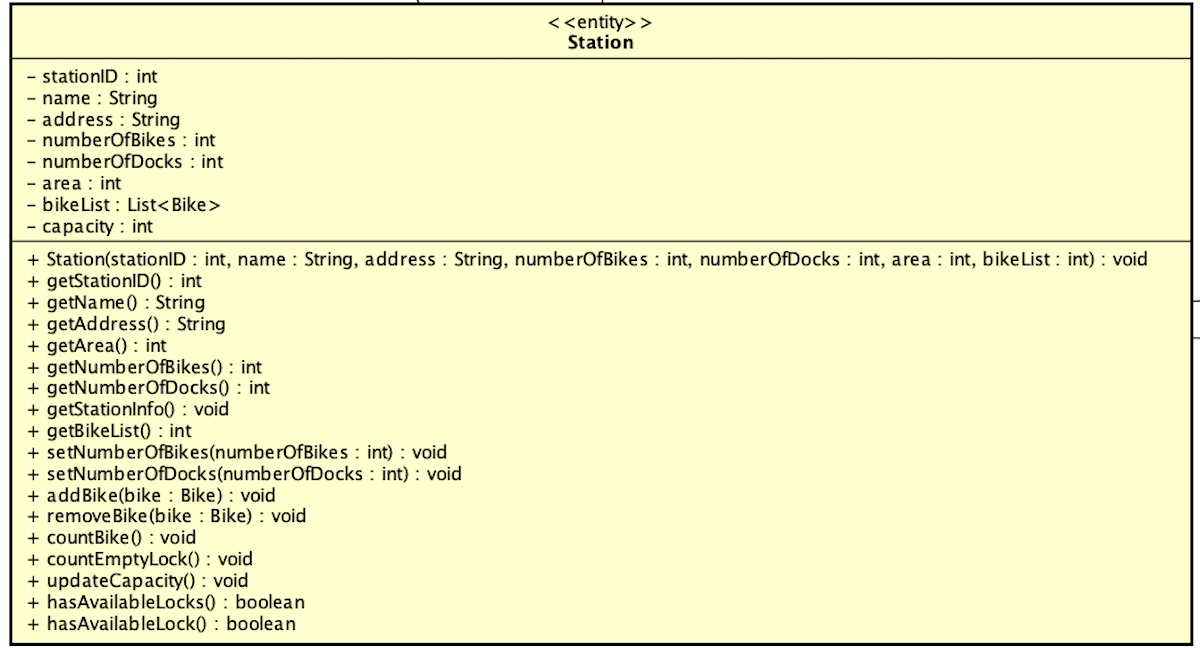
* 1. *Method*

None

* 1. *State*



1. **Station**



* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description (purpose)** |
| 1 | stationID | int | NULL | ID of the station in the system |
| 2 | name | String | NULL | Name of the station |
| 3 | address | String | NULL | Address of the station |
| 4 | numberOfBikes | int | NULL | Number of bikes currently in the station |
| 5 | numberOfDocks | int | NULL | Number of docks in the station |
| 6 | area | int | NULL | Area size of the station |
| 7 | bikeList | List<Bike> | NULL | List of the bikes currently in the station |
| 8 | capacity | int | NULL | The number of available docks left in the station |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | getStationID | int | Get the stationID |
| 2 | getName | String | Get the name of the station |
| 3 | getAddress | String | Get the address of the station |
| 4 | getNumberOfBikes | int | Get the number of bikes currently in the station |
| 5 | getNumberOfDocks | int | Get the number of docks in the station |
| 6 | getBikeList | List <Bike> | Get the list of bikes currently in the station |
| 7 | addBike | Bike | Add a bike to the list of bikes currently in the station |
| 8 | removeBike | Bike | Remove of bike from the list of bikes currently in the station |
| 9 | updateCapacity | void | Update the capacity of the station according to numberOfBikes and numberOfDocks |

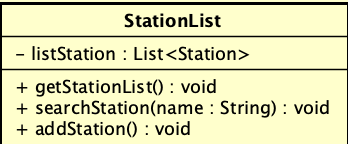
* + 1. *Parameter*
* bike – Bike: A particular bike object
* rentalCode – String: code given by the system when user successfully rents a bike and can be used for other purposes
* barcode – String: Unique code of each bike in String format
  + 1. *Exception*
* None
  1. *Method*

None

* 1. *State*

None

1. **StationList**



* 1. *Attribute*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Data type** | **Default value** | **Description** |
| 1 | listStation | List<Station> | NULL | List of the stations |

* 1. *Operation*

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Return type** | **Description (purpose)** |
| 1 | getStationList | List<Station> | Get list of the stations |
| 2 | searchStation | Station | Search for a station by its name |
| 3 | addStation | void | Add a new station to the list |

* + 1. *Parameter*
* name – String: Name of station
* station – Station: The station you want to add
  + 1. *Exception*
* None
  1. *Method*
* None
  1. *State*

None