Class Design - Model

1. Create Initial Design Classes

- 📁 model
  - 📁 bike
  - 📁 db
    - ◯ EBRDB
  - 📁 dock
    - ◯ Dock
    - ◯ DockManager
  - 📁 invoice
    - ◯ Invoice
    - ◯ InvoiceManager
  - 📁 payment
    - 📁 creditCard
      - ◯ CreditCard
      - ◯ CreditCardManager
    - 📁 transaction
      - ◯ PaymentTransaction
      - ◯ PaymentTransactionManager
  - 📁 session
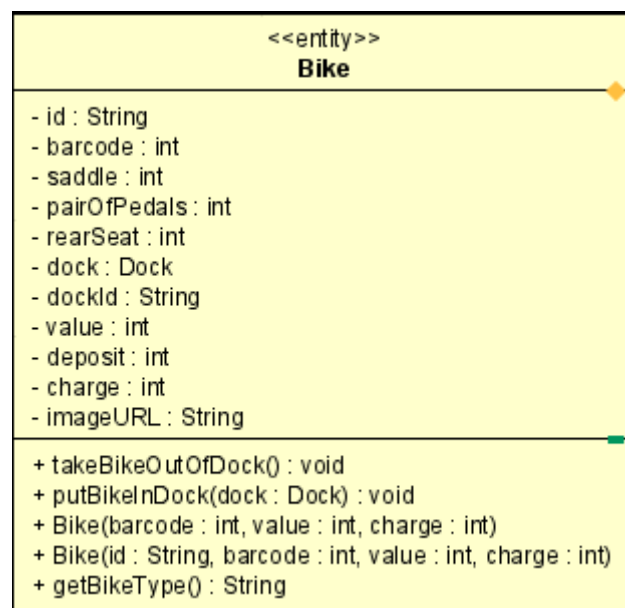   - ◯ Session
    - ◯ SessionManager
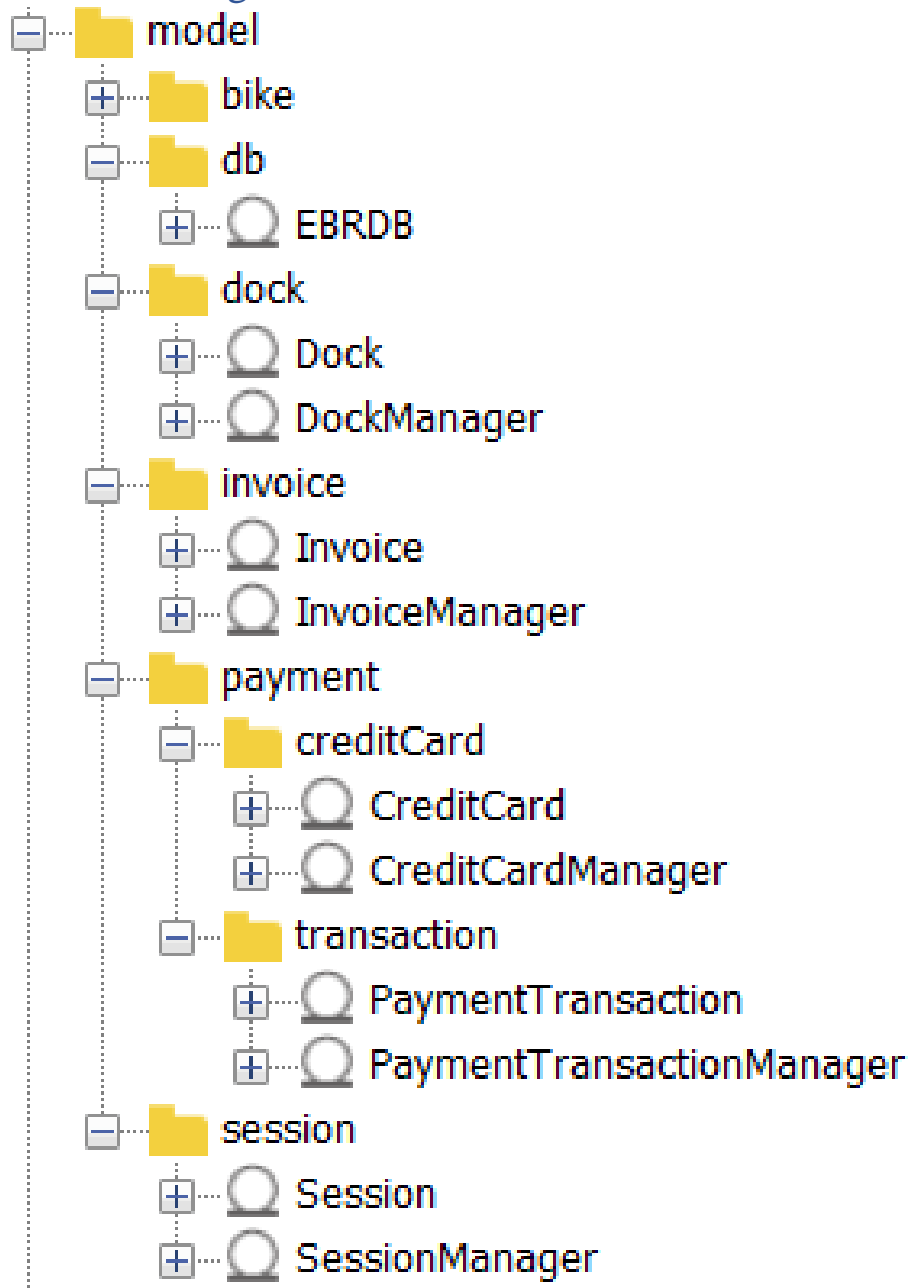
2. Define Relationships Between Classes
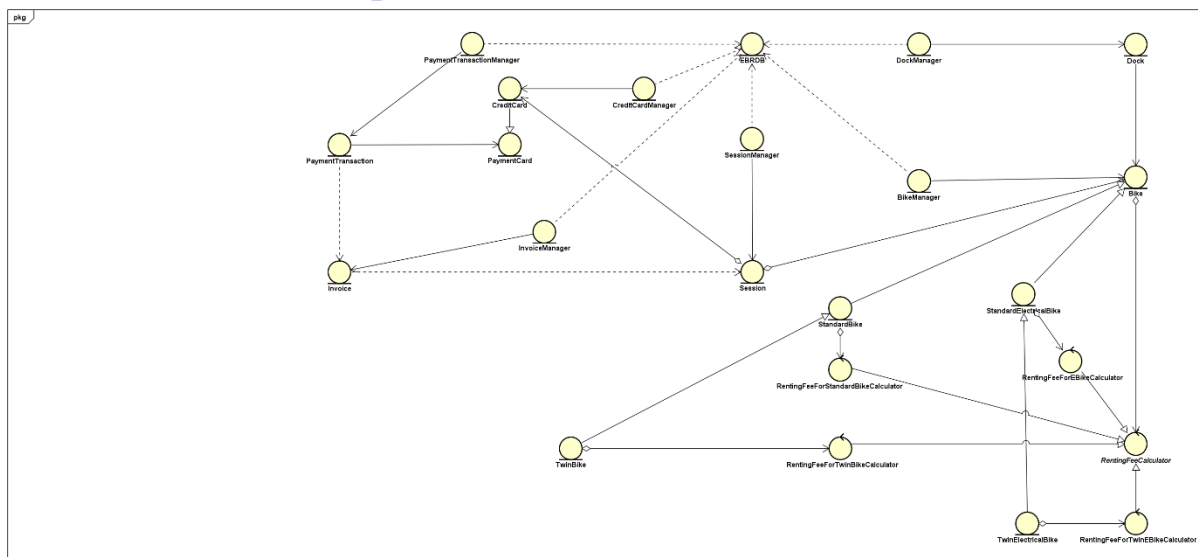
3. Class Design

3.1. Class "Bike"

Attribute

#	Name	Data type	Default value	Description
1	id	String	NULL	Bike id
2	barcode	int	NULL	barcode
3	saddle	int	NULL	saddle
4	pairOfPedals	int	NULL	Pair of pedals
5	rearSeat	int	NULL	Number of rear seats
6	dock	Dock	NULL	dock
7	dockId	String	NULL	Dock id
8	value	int	NULL	value
9	doposit	int	NULL	deposit
10	charge	int	NULL	charge

11	imageURL	String	NULL	Image URL

Operation

#	Name	Return type	Desciption
1	takeBikeOutOfDock	void	Take bike out of dock
2	putBikeInDock	void	Put bike to a dock
3	getBikeType	String	Get bike type

Parameter:
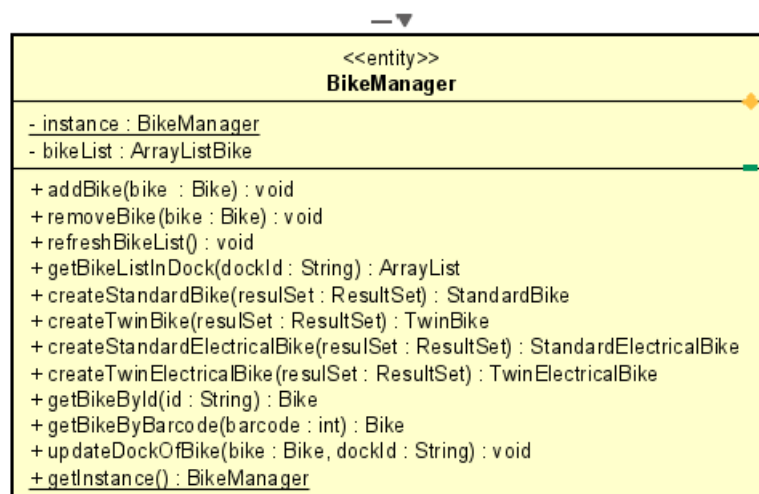
- putBikeInDock
  - dock - Dock

Exception:

None

Method

None

State

None

3.2. Class "BikeManager"

Attribute

#	Name	Data type	Default value	Description
1	instance	BikeManager	NULL	Bike instance
2	bikeList	ArrayList	NULL	Bike list

Operation

#	Name	Return type	Desciption
1	addBike	void	Add bike
2	removeBike	void	Remove bike
3	refreshBikeList	void	Refresh bike list
4	getBikeListInDock	ArrayList	Get bike list in dock
5	createStandardBike	StandardBike	Create standard bike
6	createTwinBike	TwinBike	Create twin bike
7	createStandardEBike	StandardEBike	Create standard e bike
8	createTwinEBike	TwinEBike	Create twin e bike
9	getBikeById	Bike	Get bike by id
10	getBikeByBarcode	Bike	Get bike by barcode
11	updateDockOfBike	void	Update dock
12	getInstance	BikeManager	Get bike instance

Parameter:

- addBike
  - bike - Bike
- getBikeListInDock
  - dockId - String
- createStandardBike
  - resultSet - ResultSet
- createTwinBike
  - resultSet - ResultSet
- createTwinEBike
  - resultSet - ResultSet
- getBikeById
  - id – String
- getBikeByBarcode
  - barcode - int
- updateDockOfBike
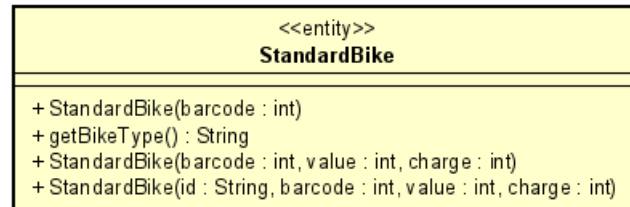 - dockId – String
  - bike - Bike

Exception:

None

Method

None

State

> None

3.3. Class "StandardBike"

Attribute

> None

Operation

#	Name	Return type	Desciption
1	getBikeType	String	Get bike type

Parameter:

> None

Exception:

> None

Method

> None

State

> None

3.4. Class "StandardElectricalBike"
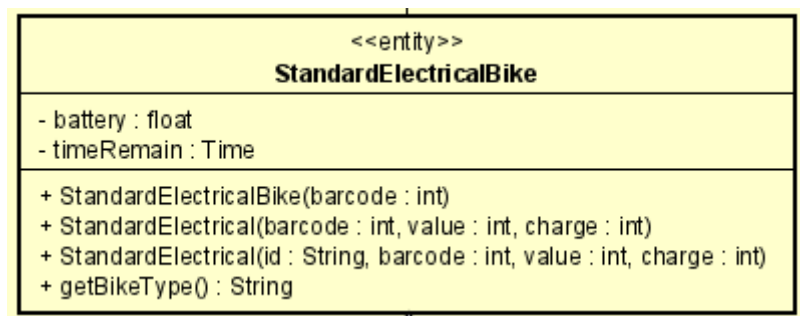
Attribute

#	Name	Data type	Default value	Description

1	battery	float	100.0	battery remained
2	timeRemain	Time	NULL	usage time base on battery and bike

Operation

#	Name	Return type	Desciption
1	getBikeType	String	Get bike type

Parameter:

None

Exception:

None

Method

None

State

None

3.5. Class "TwinBike"

Attribute

None

Operation

#	Name	Return type	Desciption
1	getBikeType	String	Get bike type

Parameter:

None

Exception:

None

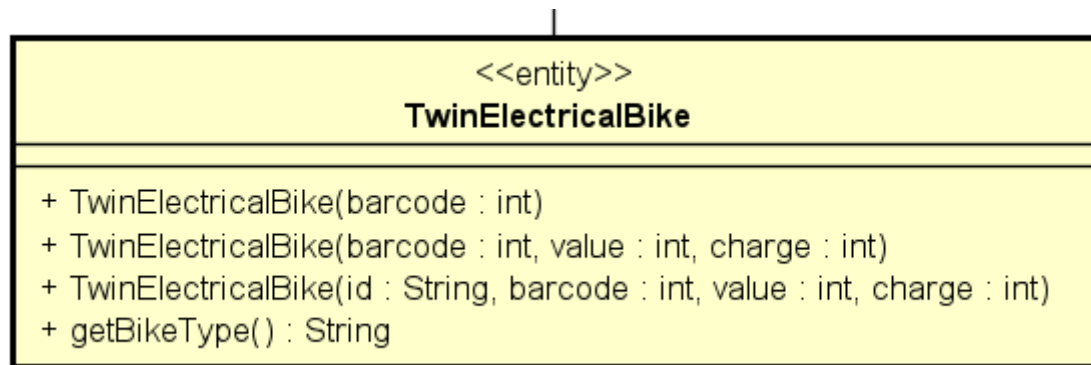**Method**

None

State

    None

3.6. Class "TwinElectricalBike"

Attribute

    None

Operation

#	Name	Return type	Desciption
1	getBikeType	String	Get bike type

Parameter:

    None

Exception:

    None

Method

    None

State

    None

3.7. Class "EBRDB"

Attribute

　　None

Operation

#	Name	Return type	Desciption
1	getConnection	Connect	get a connection to PostgreSQL database

Parameter:

　　None

Exception:

　　None

Method

　　None

State

　　None

3.8. Class "Dock"
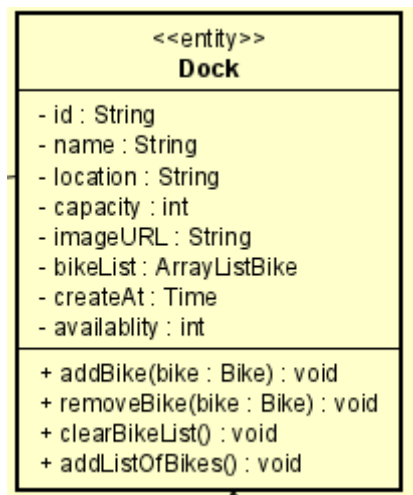
Attribute

#	Name	Data type	Default value	Description
1	id	String	NULL	Dock id
2	name	String	NULL	Name of dock
3	location	String	NULL	Dock location
4	capacity	int	NULL	Dock size
5	imageURL	String	NULL	Dock image
6	bikeList	ArrayList<Bike>	NULL	Bike list in dock
7	createAt	Time	NULL	Time Create Dock
8	Availalbity	Int	NULL	Number of availability bike

Operation

#	Name	Return type	Description
1	addBike	void	Add bike to dock
2	removeBike	void	Remove bike out of dock
3	clearBikeList	void	Clear bike list in dock
4	addListOfBike	void	Add bike list to dock

Parameter:

- addBike
  - bike – Bike
- removeBike
  - bike – Bike

Exception:

None

Method

None

State

None

3.9. Class "DockManager"
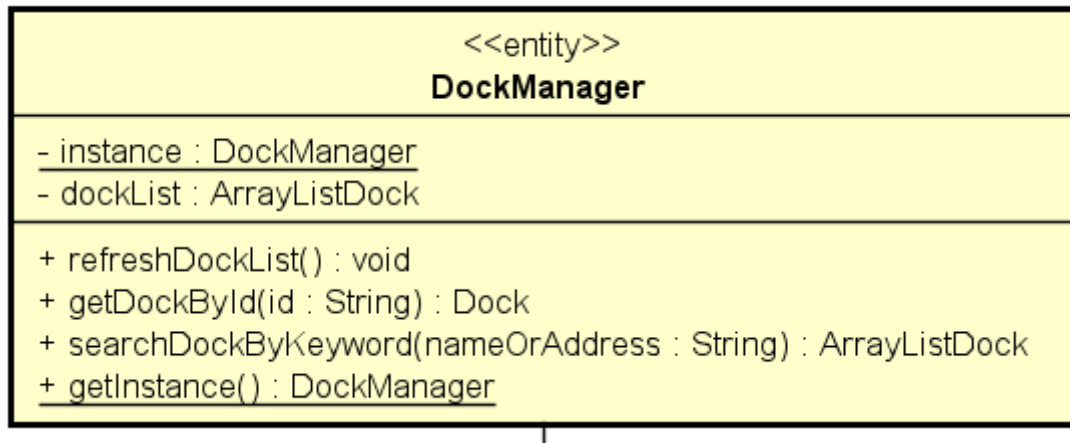
Attribute

#	Name	Data type	Default value	Description
1	instance	DockManager	NULL	Instance of a dock
2	dockList	ArrayList<Dock>	NULL	Dock list

Operation

#	Name	Return type	Desciption
1	refreshDockList	void	Refresh dock list
2	getDockById	Dock	Get dock by id
3	searchDockByKeyword	ArrayList<Dock>	Get docks by keyword
4	getInstance	DockManager	Get dock instance

Parameter:

- getDockById
  - id - String
- searchDockByKeyWord
  - nameOrAddress - String

Exception:

None

Method

None

State

>   None

3.10.   Class "Invoice"
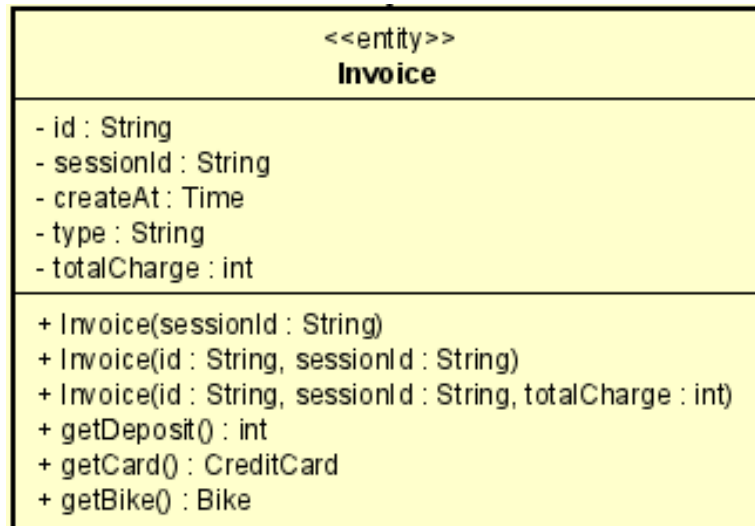
Attribute

#	Name	Data type	Default value	Description
1	id	String	NULL	Invoice id
2	sessionId	String	NULL	Session id
3	createAt	Time	NULL	Create time
4	type	String	NULL	Session corresponds to invoice
5	totalCharge	int	NULL	Total Charge

Operation

#	Name	Return type	Desciption
1	getDeposit	int	Get deposit
2	getCard	CreditCard	Get credit card
3	getBike	Bike	Get bike

Parameter:

>   None

Exception:

>   None

Method

>   None

State

None

3.11. Class "InvoiceManager"
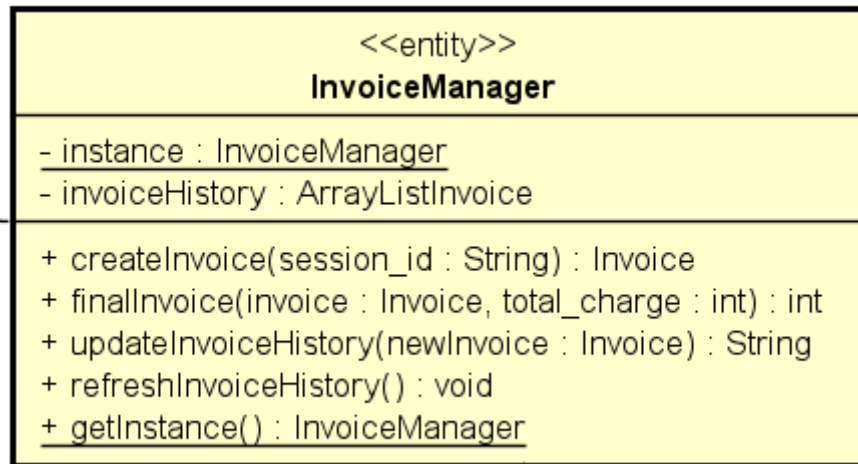
Attribute

#	Name	Data type	Default value	Description
1	instance	DockManager	NULL	Instance of a dock
2	invoiceHistory	ArrayList<Invoice>	NULL	Invoice History

Operation

#	Name	Return type	Desciption
1	createInvoice	Invoice	Create new invoice
2	finalInvoice	int	Complete invoice
3	updateInvoiceHistory	String	Update invoice history
4	refreshInvoiceHistory	void	Refresh invoice history
5	getInstance	InvoiceManager	Get invoice instance

Parameter:

- createInvoice
    - session_id - String
- finallInvoice
    - invoice – Invoice
    - total_charge: int
- updateInvoiceHistory
    - newInvoice - Invoice

Exception:

None

Method

      None

State

      None

3.12. Class "CreditCard"
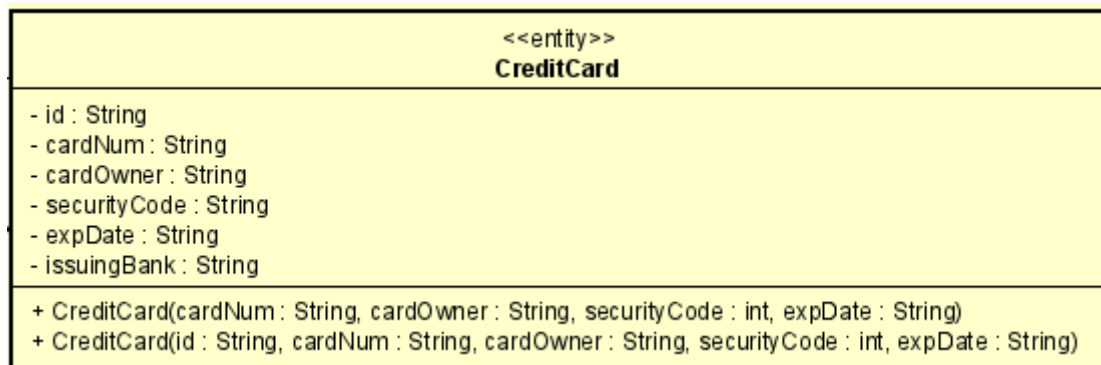
Attribute

#	Name	Data type	Default value	Description
1	id	String	NULL	Card id
2	cardNum	String	NULL	Card number
3	cardOwner	String	NULL	Card owner's name
4	securityCode	String	NULL	Card security code
5	expDate	String	NULL	Card exp date
6	issuingBank	String	NULL	IssuingBank

Operation

      None

Parameter:

      None

Exception:

      None

Method

      None

State

      None

3.13. Class "CreditCardManager"
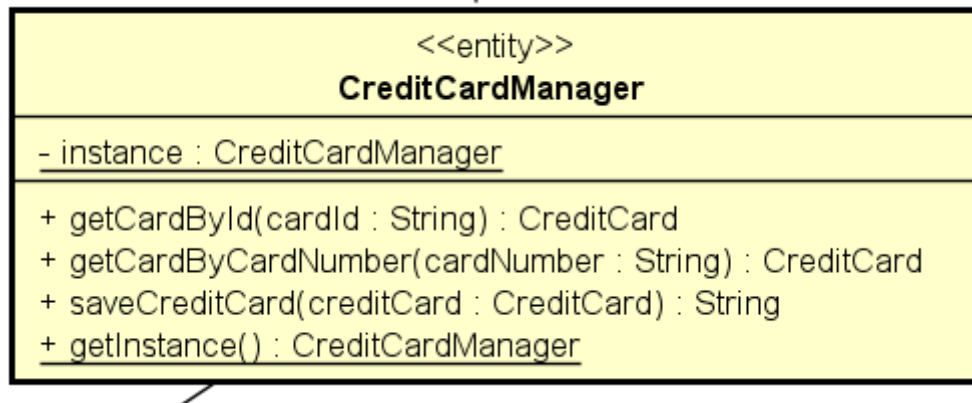
Attribute

#	Name	Data type	Default value	Description
1	instance	CreditCardManager	NULL	Instance of a card

Operation

#	Name	Return type	Desciption
1	getCardById	CreditCard	Get card by id
2	getCardByCardNumber	CreditCard	Get card by card number
3	saveCreditCard	String	Save credit card
4	getInstance	CreditCardManager	Get card instance

Parameter:

- getCardById
    - id - String
- getCardByCardNumber
    - carNumber – String
- saveCreditCard
    - credidCard - CreditCard

Exception:

None

Method

None

State

None

3.14.　Class "PaymentTransaction"

```
                    <<entity>>
                 PaymentTransaction
- id : String
- errorCode : String
- card : PaymentCard
- amount : int
- createAt : String
- type : String
- method : String
- contents : String
+ PaymentTransaction(errCode : String, card : PaymentCard, transactionId : String, transactionContent : String, amount : int, createdAt : String)
+ PaymentTransaction(id : String, transactionId : String, type : String, amount : int, method : String)
```

Attribute

#	Name	Data type	Default value	Description
1	errorCode	String	NULL	Error code
2	card	PaymentCard	NULL	Credit card
3	transactionId	String	NULL	Transaction id
4	transactionContent	String	NULL	Contents of transaction
5	amount	int	NULL	Transaction amount
6	createAt	String	NULL	Transaction create time
7	id	String	NULL	PaymentTransaction id
8	type	String	NULL	Transaction type
9	method	String	NULL	Transaction method

Operation

None

Parameter:

None

Exception:

None

Method

None

State

None

3.15.   Class "PaymentTransactionManager"

```
                          <<entity>>
                  PaymentTransactionManager

 - instance : PaymentTransactionManager

 + getTransactionById(transactionId : String) : PaymentTransaction
 + savePaymentTransaction(paymentTransaction : PaymentTransaction) : String
```
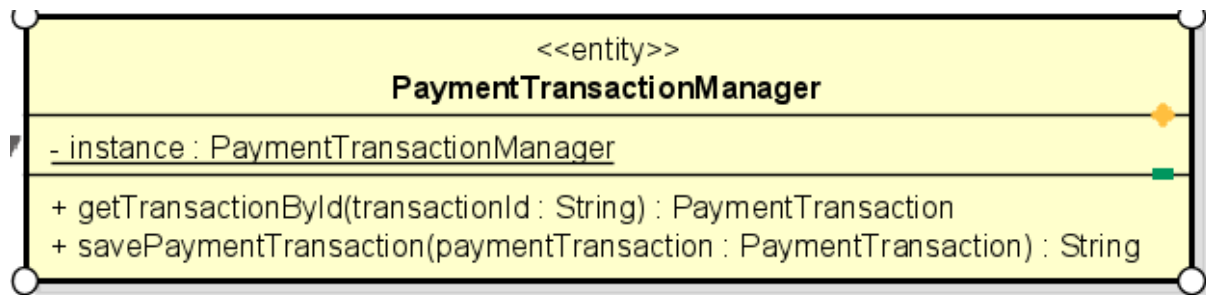
Attribute

#	Name	Data type	Default value	Description
1	instance	PaymentTransaction	NULL	Instance of transaction

Operation

#	Name	Return type	Desciption
1	getTransactionById	PaymentTransaction	Get transaction by Id
2	savePaymentTransaction	String	Save payment transaction

Parameter:

- getTransactionById
    - transactionId - String
- savePaymentTransaction
    - paymentTransaction - Paymentransaction

Exception:

None

Method

None

State

None

3.16.   Class "Session"

```
                              <<entity>>
                               Session

- id : String
- bikeId : String
- rentDockId : String
- returnDockId : String
- startTime : LocalDateTime
- endTime : LocalDateTime

+ isActive() : boolean
+ setActive() : void
+ getSessionLength() : long
+ Session(id : String, bike : Bike, card : CreditCard, rentTransaction : PaymentTransaction)
+ Session(id : String, bike : Bike, card : CreditCard, startTime : LocalDateTime, rentTransaction : PaymentTransaction)
+ Session(id : String, bike : Bike, card : CreditCard, startTime : LocalDateTime, endTime : LocalDateTime, rentTransaction : PaymentTransaction, returnTransaction : PaymentTransaction)
```

Attribute

#	Name	Data type	Default value	Description
1	id	String	NULL	Session id
2	bikeId	String	NULL	Bike Id
3	rentDockId	String	NULL	Dock rented Id
4	returnDockId	String	NULL	Dock Return Id
5	endTime	LocalDateTime	NULL	End time
6	startTime	LocalDateTime	NULL	Start time

Operation

#	Name	Return type	Desciption
1	isActive	boolean	Session is active or not
2	setActive	void	Set session active
3	getSessionLength	long	Get session length

Parameter:

None

Exception:

None

Method

None

State

None

3.17.　Class "SessionManager"

```
                         ─ ▾
┌─────────────────────────────────────────────────────────────────────┐
│                            <<entity>>                               ◆│
│                          SessionManager                              │
├─────────────────────────────────────────────────────────────────────┤
│ - instance : SessionManager                                          │
│ - sessions : ArrayListSession                                       ─│
├─────────────────────────────────────────────────────────────────────┤
│ + createSession(bike : Bike, card : CreditCard, rentTransaction : PaymentTransaction) : Session │
│ + endSession(session : Session, returnTransaction : PaymentTransaction) : int │
│ + getSessionById(id : String) : Session                             │
│ + insertNewSession(newSession : Session) : String                   │
│ + refreshSessionList() : void                                       │
│ + pauseSession(session : Session) : int                             │
│ + resumeSession(session : Session) : int                            │
│ + switchSessionState(session : Session) : void                      │
└─────────────────────────────────────────────────────────────────────┘
```

Attribute

#	Name	Data type	Default value	Description
1	instance	SessionManager	NULL	Session instance
2	sessions	ArrayList<Session>	NULL	Session list

Operation

#	Name	Return type	Desciption
1	createSession	Session	Create a session
2	endSession	int	Complete session
3	getSessionById	Session	Get session by id
4	insertNewSession	String	Insert new session to db
5	refreshSessionList	void	Refresh session list
6	pauseSession	int	Pause session
7	resumeSession	int	Resume session
8	switchSessionState	void	Switch session state

Parameter:

- createSession
  - bike - Bike
  - card – CreditCard
  - rentTransaction - PaymentTransaction
- endSession
  - session – Session
  - returnTransaction - PaymentTransaction
- getSessionById

- o id - String
- insertNewSession
  - o newSession – Session
- pauseSession
  - o session – Session
- resumeSession
  - o session – Session
- switchSessionState
  - o session – Session

Exception:

None

Method

None

State

None