### Class Design

#### Table Description automatically generatedClass “RentbikeController”

**Hình 4-13: Biểu đồ lớp RentbikeController Đặc tả các thuộc tính**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | rentalCode | int | null | Mã thuê xe |
| 2 | rentalCounter | int | 0 | Bộ đếm mã thuê xe phục vụ cho rentalCode |

**Bảng 4-5: Đặc tả các thuộc tính của RentbikeController Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | processRentBike | String | Xử lý thuê xe |
| 2 | updateRentBikeStatus | void | Cập nhật trạng thái thuê xe |
| 3 | convertBikeCodeToRentalCode | String | Chuyển đổi mã xe sang mã thuê xe |
| 4 | getRentBikeInvoice | RentBikeInvoice | Hóa đơn thuê xe |

**Bảng 4-6: Đặc tả các phương thức của RentbikeController**

*Parameter*:

* card – thẻ được sử dụng
* bike – xe muốn thuê
* amount – tổng số tiền thuê xe
* rentStamp – timestamp lúc thuê
* returnStamp – timestamp lúc trả
* bikeID – mã xe
* rentalCode – mã thuê xe

*Exception*:

* Các exception sẽ xảy ra trong quá trình thanh toán: InvalidCardException, …

**Method** Không **State**

Không

#### Class “BarcodeController”

Table

Description automatically generated with medium confidence

**Hình 4-14: Biểu đồ lớp BarcodeController Đặc tả các thuộc tính**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | bikeCode | int | null | Mã code xe |

**Bảng 4-7: Đặc tả các thuộc tính của BarcodeController Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | getBikeFromBarcode | Bike | Chuyển đổi từ barcode sang xe |

**Bảng 4-8: Đặc tả các phương thức của BarcodeController**

*Parameter*:

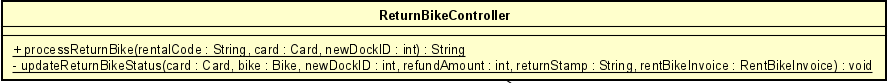
* Barcode: Mã barcode được nhập từ người dùng

*Exception*:

* InvalidBarcodeException

**Method** Không **State** Không

#### Class “ReturnbikeController”



**Hình 4-15: Biểu đồ lớp ReturnBikeController Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | processReturnBike | String | Xử lý trả xe |

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | updateReturnBikeStatus | void | Cập nhật trạng thái trả xe |

**Bảng 4-9: Đặc tả các phương thức của ReturnBikeController**

*Parameter*:

* rentalCode – mã thuê xe
* card – thẻ được sử dụng
* newDockID – ID bãi xe mới
* bike – xe đang được thuê và sẽ tiến hành trả xe
* refundAmount – tổng tiền trả lại
* returnStamp – timestamp lúc trả xe
* rentBikeInvoice – hóa đơn thuê xe

*Exception*:

* Các exception sẽ xảy ra trong quá trình thanh toán: InvalidCardException, …

**Method** Không **State** Không

#### Table Description automatically generatedClass “CardController”

**Hình 4-16: Biểu đồ lớp CardController**

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | checkCardInUse | boolean | Kiểm tra thẻ có đang hoạt động |
| 2 | validateCardInfo | boolean | Kiểm tra thông tin thẻ |
| 3 | validateCardCode | boolean | Kiểm tra mã thẻ |
| 4 | validateOwner | boolean | Kiểm tra người sở hữu |
| 5 | validateCcvCode | boolean | Kiểm tra mã ccv |
| 6 | validateExpiredDate | boolean | Kiểm tra ngày hết hạn |

**Bảng 4-10: Đặc tả các phương thức của CardController**

*Parameter*:

* cardCode – mã thẻ
* owner – người sở hữu
* cvvCode – mã cvv
* ExpiredDate – ngày hết hạn

*Exception*:

* Không

**Method** Không **State** Không

#### Class “DockController”

Diagram

Description automatically generated with medium confidence

**Hình 4-17: Biểu đồ lớp DockController**

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | getDocks | ArrayList<Dock> | Lấy thông tin bãi xe |
| 2 | getBikes | ArrayList<Bike> | Lấy thông tin xe |
| 3 | getDockFromString | Dock | Lấy thông tin bãi xe |
| 4 | getBikeFromID | Bike | Lấy thông tin xe từ ID |
| 5 | checkBikeIsInUse | boolean | Kiểm tra xe đang dùng |

**Bảng 4-11: Đặc tả các phương thức của DockController**

*Parameter*:

* dockID – mã bãi xe
* dockInfo – thông tin bãi xe
* id – mã xe/ mã bãi xe tùy theo phương thức

*Exception*:

* Không

**Method** Không **State** Không

#### Class “BarcodeConverter”

Diagram

Description automatically generated with medium confidence

**Hình 4-18: Biểu đồ lớp BarcodeConverter**

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | convertBarcodeToBikeCode | int | Chuyển barcode sang mã xe |

**Bảng 4-12: Đặc tả các phương thức của BarcodeConverter**

*Parameter*:

* barcode – barcode nhập vào từ người dùng

*Exception*:

* InvalidBarcodeException

**Method** Không **State** Không

#### Class “IBarcodeConverter”

Diagram

Description automatically generated

**Hình 4-19: Biểu đồ lớp IBarcodeConverter Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | convertBarcodeToBikeCode | int | Chuyển barcode sang mã xe |

**Bảng 4-13: Đặc tả các phương thức của IBarcodeConverter**

*Parameter*:

* Barcode: Mã barcode nhập từ người dùng

*Exception*:

* InvalidBarcode

**Method** Không **State** Không

#### Class “BarcodeConvertion”

Table

Description automatically generated

**Hình 4-20: Biểu đồ lớp BarcodeConvertion Đặc tả các thuộc tính**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | barcode | String |  | Barcode nhập vào từ người dùng |
| 2 | URL | String | https://... | URL kết nối đến API chuyển barcode |

**Bảng 4-14: Đặc tả các thuộc tính của BarcodeConverter Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | BarcodeConvertion | BarcodeConvertion | Chuyển đổi barcode |
| 2 | pack | String | Phương thức đóng gói một gói tin |
| 3 | post | String | Phương thức POST lên 1 URL |

**Bảng 4-15: Đặc tả các phương thức của BarcodeConvertion**

*Parameter*:

* barcode – barcode nhập vào từ người dùng

*Exception*:

* Không

**Method** Không **State** Không

#### Class “Message” (của gói subsystem.barcodeconverter)

Diagram

Description automatically generated

**Hình 4-21: Biểu đồ lớp Message (của gói subsystem.barcodeconverter) Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | pack | String | Phương thức đóng gói một gói tin |
| 2 | post | String | Phương thức POST lên 1 URL |

**Bảng 4-16: Đặc tả các phương thức của Message (của gói subsystem.barcodeconverter)**

*Parameter*:

* Không

*Exception*:

* Không

**Method** Không **State** Không

#### Class “IInterBank”

Table

Description automatically generated

**Hình 4-22:Biểu đồ lớp IInterbank**

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | processTransaction | String | Xử lý giao dịch, trả về mã lỗi giao dịch |
| 2 | reset | String | Reset thẻ, trả về mã lỗi giao dịch |

**Bảng 4-17: Đặc tả các phương thức của IInterbank**

*Parameter*:

* card – thẻ của người dùng
* amount – số tiền cần phải trả
* content – nội dung giao dịch
* createAt – thời gian giao dịch
* command – lệnh của giao dịch theo quy ước của API: pay/refund.

*Exception*:

* Các exception xảy ra trong quá trình giao dịch như: InvalidCardException, …

**Method** Không **State** Không

#### Class “Interbank”

Diagram

Description automatically generated with low confidence

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

**Hình 4-23: Biểu đồ lớp Interbank**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | processTransaction | String | Xử lý giao dịch |
| 2 | reset | String | Reset thẻ |

**Bảng 4-18: Đặc tả phương thức của Interbank**

*Parameter*:

* card – thẻ được sử dụng
* amount – số tiền được giao dịch
* content – nội dung của giao dịch
* createAt – thời gian giao dịch
* command – lệnh của giao dịch: pay/ refund

*Exception*:

* Các exception xảy ra trong quá trình giao dịch như: InvalidCardException, …

**Method** Không **State** Không

#### Class “Message” (của gói subsystem.interbank)

Diagram

Description automatically generated

**Hình 4-24: Biểu đồ lớp Message (của gói subsystem.interbank) Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | pack | String | Phương thức đóng gói một gói tin |
| 2 | patch | String | Phương thức PATCH lên 1 URL |

**Bảng 4-19: Đặc tả các phương thức của Message (của gói subsystem.interbank)**

*Parameter*:

* Không

*Exception*:

* Không

**Method** Không **State** Không

#### Class “ResetTransaction”

Table

Description automatically generated

**Hình 4-25: Biểu đồ lớp ResetTransaction**

**Đặc tả các thuộc tính**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | URL | int | "https://ecopark-system- api.herokuapp.com/api/card/reset- balance" | URL đến API để thực hiện reset thẻ |

**Bảng 4-20: Đặc tả các thuộc tính của ResetTransaction Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | ResetTransaction | ResetTransaction | Phương thức khởi tạo |
| 2 | pack | String | Đóng gói gói tin |
| 3 | patch | String | Phương thức PATCH lên URL |

**Bảng 4-21: Đặc tả các phương thức của ResetTransaction**

*Parameter*:

* card – đối tượng thẻ cần thao tác

*Exception*:

* Không

**Method** Không **State** Không

#### Class “ProcessTransaction”

Table

Description automatically generated with medium confidence

**Hình 4-26: Biểu đồ lớp ProcessTransaction Đặc tả các thuộc tính**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | amount | int |  | Số tiền giao dịch |
| 2 | content | String |  | Nội dung giao dịch |
| 3 | createAt | String |  | Thời gian giao dịch |
| 4 | command | String |  | Lệnh của API: pay/ refund |
| 5 | URL | String | https://ecopark-system- api.herokuapp.com/api/card/processTransaction | Link tới server của API |
| 6 | VERSION | String | 1.0.1 | Phiên bản, cần thiết để gọi API |

**Bảng 4-22: Đặc tả các thuộc tính của ProcessTransaction Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | ProcessTransaction | ProcessTransaction | Phương thức khởi tạo |
| 2 | pack | String | Phương thức đóng gói gói tin |
| 3 | patch | String | Phương thức PATCH lên 1 URL |

**Bảng 4-23: Đặc tả các phương thức của ProcessTransaction**

*Parameter*:

* card – thẻ được sử dụng
* amount – số tiền cần giao dịch
* content – nội dung cần giao dịch
* createAt – thời gian giao dịch
* command – lệnh của giao dịch: pay/ refund

*Exception*:

* Không

**Method** Không **State** Không

#### Class “HashFunction”

A picture containing diagram

Description automatically generated

**Hình 4-27: Biểu đồ lớp HashFunction**

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | hashTransaction | String | Xử lý thuê xe |
| 2 | hashRentalCode | String | Xử lý để hash mã thuê xe |

**Bảng 4-24: Đặc tả các phương thức của HashFunction**

*Parameter*:

* key – được sử dụng làm nguyên liệu cho hàm băm
* transaction – Thông tin giao dịch, được đóng gói theo đặc tả API
* rentalCode – mã thuê xe

*Exception*:

* Không

**Method**

hashTransaction: chuyển đổi từ một đối tượng được trừu tượng hóa InterbankTransaction thành đối tượng Json phù hợp với API:

Diagram

Description automatically generated

**Hình 4-28: Chuyển đổi InterbankTransaction thành đối tượng Json**

**State**

Không

#### Class “HTTPBinder”

Text

Description automatically generated with low confidence

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

**Hình 4-29: Biểu đồ lớp HTTPBinder**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | patch | String | Phương thức PATCH lên 1 URL |
| 2 | post | String | Phương thức POST lên 1 URL |

**Bảng 4-25: Đặc tả các phương thức của HTTPBinder**

*Parameter*:

* url – liên kết tới server cần thực hiện
* body – Nội dung của gói tin

*Exception*:

* Không

**Method** Không **State** Không

#### Class “Bike”

A picture containing text

Description automatically generated

**Hình 4-30: Biểu đồ lớp Bike**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | bikeID | int | Mã xe |
| 2 | inUse | boolean | Có đang dùng không |
| 3 | value | int | Giá trị xe |
| 4 | numPedal | int | Số lượng bàn đạp |
| 5 | numSaddle | int | Số lượng yên xe |
| 6 | numSeat | int | Số chỗ ngồi |
| 7 | dockID | int | Mã bãi xe |
| 8 | licensePlate | String | Biển số xe |
| 9 | type | String | Loại xe |

**Bảng 4-26: Đặc tả các thuộc tính của Bike Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | Bike | Bike | Phương thức khởi tạo |
| 2 | updateStatus | void | Cập nhật trạng thái xe |
| 3 | calculateDeposit | int | Tính tiền cọc |
| 4 | caculateRentCost | int | Tính tiền thuê xe |
| 5 | getBike | Bike | Lấy thông tin xe |

**Bảng 4-27: Đặc tả các phương thức của Bike**

*Parameter*:

* bike – xe muốn thuê
* inUse – xe có đang được sử dụng
* dockID – mã bãi xe
* duration – thời gian thuê xe

*Exception*:

* Không

**Method** Không **State** Không

#### Class “Card”

Table

Description automatically generated

**Hình 4-31: Biểu đồ lớp Card**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | cardCode | String | Mã thẻ |
| 2 | owner | String | Người sở hữu |
| 3 | CVV | String | Cvv |
| 4 | expiredDate | String | Ngày hết hạn |

**Bảng 4-28: Đặc tả các thuộc tính của Card**

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | Card | Card | Phương thức khởi tạo |
| 2 | getCard | Card | Lấy thông tin thẻ |

**Bảng 4-29: Đặc tả các phương thức của Card**

*Parameter*:

* cardCode – mã thẻ
* owner – người sở hữu
* CVV – CVV của thẻ
* expiredDate – ngày hết hạn

*Exception*:

* Không

**Method** Không **State** Không

#### Class “Dock”

Graphical user interface, text, email

Description automatically generated

**Hình 4-32: Biểu đồ lớp Dock**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | dockID | int | Mã bãi xe |
| 2 | name | String | Tên bãi xe |
| 3 | address | String | Địa chỉ |
| 4 | area | String | Khu vực |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | numberOfDockingPoints | int | Số lượng bãi đỗ xe |

**Bảng 4-30: Đặc tả các thuộc tính của Dock Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | Dock | Dock | Phương thức khởi tạo |
| 2 | getDistance | float | Lấy khoảng cách |
| 3 | getBikeFromString | Bike | Lấy thông tin xe |
| 4 | toString | String | Phương thức chuyển thành dạng  String (ghi đề phương thức của Object) |
| 5 | getDockTable | String | Lấy danh sách bãi xe |
| 6 | checkSparePoints | boolean | Kiểm tra chỗ trống còn lại trong bãi |
| 7 | getDockInfo | String | Lấy thông tin bãi xe |
| 8 | updateRemainCapacity | void | Cập nhật chỗ trống bãi xe |

**Bảng 4-31: Đặc tả các phương thức của Dock**

*Parameter*:

* dockID – mã bãi xe
* name – tên bãi xe
* address – địa chỉ của bãi xe
* area – khu vực của bãi xe
* numberOfDockingPoints – số lượng chỗ trống gửi xe trong bãi (ban đầu)
* bikes – danh sách xe
* bikeInfo – thông tin xe

*Exception*:

* Không

**Method** Không **State** Không

#### Class “DoubleNormalBike”

Text

Description automatically generated

**Hình 4-33: Biểu đồ lớp DoubleNormalBike Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | DoubleNormalBike | DoubleNormalBike | Phương thức khởi tạo |
| 2 | caculateRentCost | Bike | Tính tiền thuê xe |

**Bảng 4-32: Đặc tả các phương thức của DoubleNormalBike**

*Parameter*:

* bike – Thông tin xe muốn truyền vào, dưới dạng String
* duration – thời gian thuê

*Exception*:

* Không

**Method** Không **State** Không

#### Class “InterbankTransaction”

Shape

Description automatically generated with low confidence

**Hình 4-34: Biểu đồ lớp InterbankTransaction Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | cardCode | String | Mã thẻ |
| 2 | owner | String | Người sở hữu |
| 3 | cvvCode | String | Mã cvv của thẻ |
| 4 | dateExpired | String | Ngày hết hạn |
| 5 | command | String | Lệnh của giao dịch: pay/ refund |
| 6 | transactionContent | String | Nội dung giao dịch |
| 7 | amount | double | Số tiền giao dịch |
| 8 | createAt | String | Thời điểm tạo |

**Bảng 4-33: Đặc tả các thuộc tính của InterbankTransaction Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | InterbankTransaction | InterbankTransaction | Phương thức khởi tạo |

**Bảng 4-34: Đặc tả các phương thức của InterbankTransaction**

*Parameter*:

* card – thẻ được sử dụng
* command – lệnh của giao dịch: pay/ refund
* content – nội dung của giao dịch
* amount – số tiền được giao dịch
* createAt – thời gian giao dịch

*Exception*:

* Không

**Method** Không **State** Không

#### Class “PaymentTransaction”

Graphical user interface

Description automatically generated with medium confidence

**Hình 4-35: Biểu đồ lớp PaymentTransaction**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | rentalCode | int | Mã thuê xe |
| 2 | cardCode | String | Mã thẻ |
| 3 | owner | String | Người sở hữu |
| 4 | transactionContent | String | Nội dung giao dịch |
| 5 | amount | int | Tổng số tiền |
| 6 | daytime | String | Thời gian giao dịch |

**Bảng 4-35: Đặc tả các thuộc tính của PaymentTransaction Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | PaymentTransaction | PaymentTransaction | Phương thức khởi tạo |
| 2 | savePaymentTransaction | void | Lưu thông tin thanh toán |

**Bảng 4-36: Đặc tả các phương thức của PaymentTransaction**

*Parameter*:

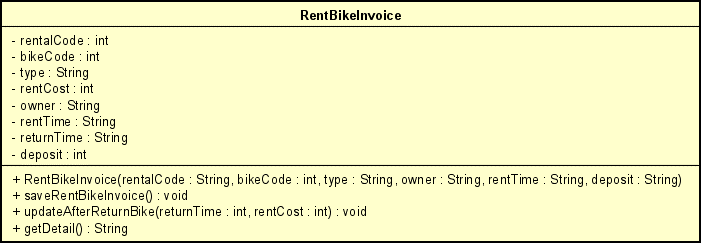
* rentalCode – mã thuê xe
* cardCode – mã thẻ
* owner – người sở hữu
* transactionContent – nội dung giao dịch
* amount – tổng số tiền
* daytime – thời gian giao dịch

*Exception*:

* Không

**Method** Không **State** Không

#### Class “RentBikeInvoice”



**Hình 4-36: Biểu đồ lớp RentBikeInvoice**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | rentalCode | int | Mã thuê xe |
| 2 | bikeCode | int | Mã xe |
| 3 | type | String | Loại xe |
| 4 | rentCost | int | Giá thuê xe |
| 5 | owner | String | Người sở hữu |
| 6 | rentTime | String | Thời gian thuê |
| 7 | returnTime | String | Thời gian trả |
| 8 | deposit | int | Tiền cọc |

**Bảng 4-37: Đặc tả các thuộc tính của RentBikeInvoice Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | RentBikeInvoice | RentBikeInvoice | Phương thức khởi tạo |
| 2 | saveRentBikeInvoice | void | Lưu hóa đơn thuê xe |
| 3 | updateAfterReturnBike | void | Cập nhật thông tin xe sau khi trả |
| 4 | getDetail | String | Lấy thông tin chi tiết |

**Bảng 4-38: Đặc tả các phương thức của RentBikeInvoice**

*Parameter*:

* rentalCode – mã thuê xe
* bikeCode – mã xe
* type – loại xe
* owner – người sở hữu
* rentTime – thời gian bắt đầu thuê xe
* deposit – tiền cọc
* returnTime – thời gian trả xe
* rentCost – giá thuê xe

*Exception*:

* Không

**Method** Không **State** Không

#### Class “SingleElectricBike”

Text

Description automatically generated

**Hình 4-37: Biểu đồ lớp SingleElectricBike**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | remainBattery | int | Phần trăm pin còn lại |
| 2 | maxTime | int | Thời gian sử dụng tối đa |

**Bảng 4-39: Đặc tả các thuộc tính của SingleElectricBike Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | SingleElectricBike | SingleElectricBike | Phương thức khởi tạo |
| 2 | SingleElectricBike | SingleElectricBike | Phương thức khởi tạo |
| 3 | caculateRentCost | int | Tính giá thuê xe |

**Bảng 4-40: Đặc tả các phương thức của SingleElectricBike**

*Parameter*:

* bike – Thông tin xe dưới dạng String

*Exception*:

* Không

**Method**

Không **State** Không

#### Class “SingleNormalBike”

Text

Description automatically generated

**Hình 4-38: Biểu đồ lớp SingleNormalBike**

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | SingleNormalBike | SingleNormalBike | Phương thức khởi tạo |
| 2 | SingleNormalBike | SingleNormalBike | Phương thức khởi tạo |
| 3 | caculateRentCost | int | Tính tiền thuê xe |

**Bảng 4-41: Đặc tả các phương thức của SingleNormalBike**

*Parameter*:

* bike – Thông tin xe dưới dạng String

*Exception*:

* Không

**Method** Không **State** Không

#### Class “BikeDAO”

Text, table

Description automatically generated

**Hình 4-39: Biểu đồ lớp BikeDAO**

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | updateStatus | void | Cập nhật tình trạng xe (thuê hoặc chưa thuê) |
| 2 | queryWithDockID | String | Truy vấn Dock dựa vào DockID |
| 3 | queryWithBikeCode | String | Truy vấn Bike dựa vào BikeID |
| 4 | getBikes | ArrayList<Bike> | Lấy toàn bộ thông tin Bike |

**Bảng 4-42: Đặc tả các phương thức của BikeDAO**

*Parameter*:

* bikeID – mã xe
* inUse – xe có đang sử dụng không
* dockID – mã bãi xe

*Exception*:

* Không

**Method** Không **State** Không

#### Class “DBBinder”

Table

Description automatically generated

**Hình 4-40: Biểu đồ lớp DBBinder**

**Đặc tả các thuộc tính**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default* | *Description* |
| 1 | DB\_URL | String | jdbc:mysql://localhost:3306/ecoBikeN2 | Đường dẫn của Database |
| 2 | USER\_NAME | String | “root” | Tên đăng nhập |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | PASSWORD | String |  | Mật khẩu |

**Bảng 4-43: Đặc tả các thuộc tính của DBBinder Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | getConnection | int | Lấy kết nối |
| 2 | execute | void | Thực thi câu lệnh vào MySQL |
| 3 | query | String | Truy vấn dữ liệu vào MySQL |

**Bảng 4-44: Đặc tả các phương thức của DBBinder**

*Parameter*:

* command – các câu lệnh MySQL mong muốn thực hiện.

*Exception*:

* Các exception khi sử dụng JDBC.

**Method** Không **State** Không

#### Class “DockDAO”

Graphical user interface, text, application

Description automatically generated

**Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

**Hình 4-41: Biểu đồ lớp DockDAO**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | getAllDocks | String | Lấy thông tin tất cả bãi xe |
| 2 | updateRemainCapacity | void | Cập nhật chỗ trống |

**Bảng 4-45: Đặc tả các phương thức của DockDAO**

*Parameter*:

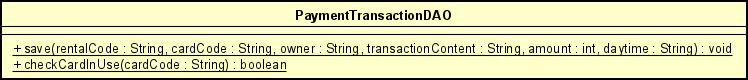
* dockID – mã bãi xe
* difference – sự thay đổi vào cơ sở dữ liệu mong muốn: “+1”/ “-1”, …

*Exception*:

* Các exception khi sử dụng JDBC.

**Method** Không **State** Không

#### Class “PaymentTransactionDAO”



**Hình 4-42: Biểu đồ lớp PaymentTransactionDAO Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | save | void | Lưu giao dịch |
| 2 | checkCardInUse | boolean | Kiểm tra thẻ đang sử dụng |

**Bảng 4-46: Đặc tả các phương thức của PaymentTransactionDAO**

*Parameter*:

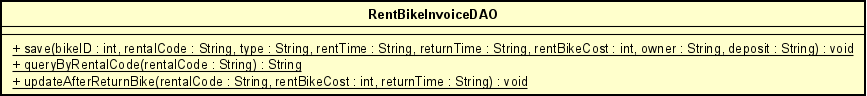
* rentalCode – mã thuê xe
* cardCode – mã thẻ
* owner – người sở hữu
* transactionContent – nội dung giao dịch
* amount – tổng số tiền
* daytime – thời gian giao dịch

*Exception*:

* Các exception khi sử dụng JDBC.

**Method** Không **State** Không

#### Class “RentBikeInvoiceDAO”



**Hình 4-43: Biểu đồ lớp RentBikeInvoiceDAO Đặc tả các thuộc tính**

Không

**Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | save | void | Lưu hóa đơn |
| 2 | queryByRentalCode | String | Truy vấn bằng mã trả xe |
| 3 | updateAfterReturnBike | void | Cập nhật sau khi trả xe |

**Bảng 4-47: Đặc tả các phương thức của RentBikeInvoiceDAO**

*Parameter*:

* bikeID – mã xe
* rentalCode – mã thuê xe
* type – loại xe
* owner – người sở hữu
* rentTime – thời gian bắt đầu thuê xe
* deposit – tiền cọc
* returnTime – thời gian trả xe
* rentBikeCost – giá thuê xe

*Exception*:

* Các exception khi sử dụng JDBC.

**Method** Không **State** Không

#### Class “BikeDetailScreen”

Table

Description automatically generated

**Hình 4-44: Biểu đồ lớp BikeDetailScreen**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | textArea | TextField | Xâu ký tự hiển thị thông tin về xe |
| 2 | backButton | Button | Nút Back |

**Bảng 4-48: Đặc tả các thuộc tính của BikeDetailScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | setBikeInfo | void | Thiết lập thông tin xe |
| 2 | handleBackButtonClick | void | Xử lý sự kiện nhấn nút Back |

**Bảng 4-49: Đặc tả các phương thức của BikeDetailScreen**

*Parameter*:

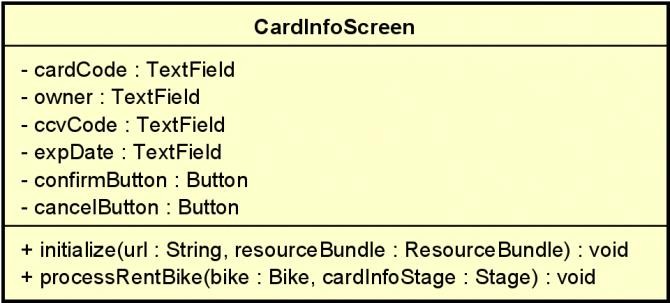
* bike – xe
* dock – bãi xe

*Exception*:

* Không

**Method** Không **State** Không

#### Class “CardInfoScreen”



**Hình 4-45: Biểu đồ lớp CardInfoScreen**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | cardCode | TextField | Mã thẻ |
| 2 | owner | TextField | Người sở hữu |

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | CVV | TextField | Cvv |
| 4 | expDate | TextField | Ngày hết hạn |
| 5 | confirmButton | Button | Phím xác nhận |
| 6 | cancelButton | Button | Phím hủy |

**Bảng 4-50: Đặc tả các thuộc tính của CardInfoScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | initialize | void | Khởi tạo |
| 2 | processRentBike | void | Xử lý thuê xe |

**Bảng 4-51: Đặc tả các phương thức của CardInfoScreen**

*Parameter*:

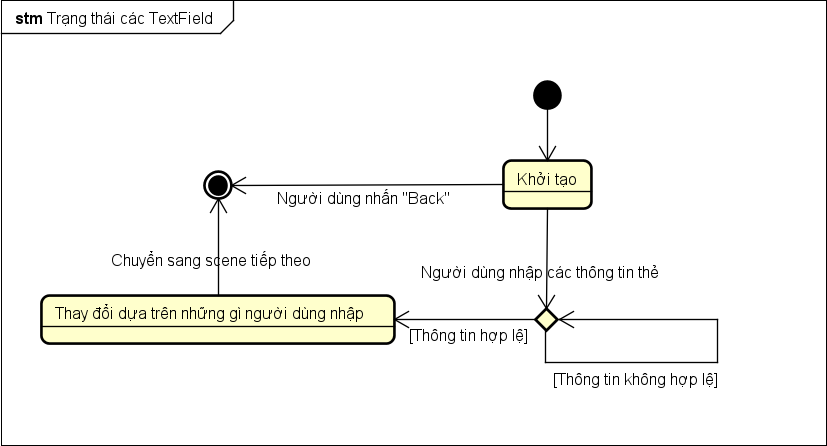
* url – đường dẫn tới fxml scene
* resourceBundle – tài nguyên đóng gói để khởi tạo fxml scene (sử dụng thư viện)
* bike – xe
* cardInfoStage – Stage để gọi tới CardInfoScreen

*Exception*:

* Không

**Method** Không **State**

Trạng thái thay đổi các TextField khi người dùng nhập vào các thông tin về thẻ:



**Hình 4-46: Biểu đồ trạng thái cho các Text Field**

#### Class “CardInfoScreenForReturnBike”

Text

Description automatically generated with low confidence

**Hình 4-47: Biểu đồ lớp CardInfoScreenForReturnBike Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | cardCode | String | Mã thẻ |
| 2 | owner | String | Người sở hữu |
| 3 | CVV | String | Cvv |
| 4 | expDate | String | Ngày hết hạn |
| 5 | confirmButton | Button | Phím xác nhận |
| 6 | cancelButton | Button | Phím hủy |

**Bảng 4-52: Đặc tả các thuộc tính của CardInfoScreenForReturnBike Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | initialize | void | Khởi tạo cho Screen |
| 2 | confirmCard | void | Xác nhận thẻ |
| 3 | getCardInfo | Card | Lấy thông tin thẻ |

**Bảng 4-53: Đặc tả các phương thức của CardInfoScreenForReturnBike**

*Parameter*:

* url – đường dẫn tới fxml scene
* resourceBundle – tài nguyên đóng gói để khởi tạo fxml scene (sử dụng thư viện)
* bike – xe
* cardInfoStage – Stage để gọi tới CardInfoScreen

*Exception*:

* Không

**Method**

Không

**State**

Không

#### Class “DockListScreen”

Table

Description automatically generated

**Hình 4-48: Biểu đồ lớp DockListScreen**

**Đặc tả các thuộc tính**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default* | *Description* |
| 1 | reset | boolean | false | Reset |
| 2 | backButton | Button |  | Nút Back |

**Bảng 4-54: Đặc tả các thuộc tính của DockListScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | setDockInfo | void | Đặt thông tin bãi xe |
| 2 | handleDoubleClickOnDockList | void | Xử lý sự kiện nhấp 2 lần vào DockList |
| 3 | showViewDockScreen | void | Đưa ra màn hình ViewDockScreen |
| 4 | handleBackButtonClick | void | Xử lý sự kiện click nút Back |

**Bảng 4-55: Đặc tả các phương thức của DockListScreen**

*Parameter*:

* dock – bãi xe

*Exception*:

* Không

**Method** Không **State** Không

#### Class “DockScreen”

Table

Description automatically generated with medium confidence

**Hình 4-49: Biểu đồ lớp DockScreen**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | backButton | Button | Nút Back |

**Bảng 4-56: Đặc tả các thuộc tính của DockScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | handleBackButtonClick | void | Xử lý sự kiện click nút Back |
| 2 | handleDoubleClickOnBikeList | void | Xử lý sự kiện nhấp 2 lần vào BikeList |
| 3 | setDockInfo | void | Đặt thông tin bãi xe |
| 4 | showViewBikeScreen | void | Đưa ra màn hình ViewBikeScreen |

**Bảng 4-57: Đặc tả các phương thức của DockScreen**

*Parameter*:

* bike – xe
* dock – bãi xe

*Exception*:

* Không

**Method** Không **State** Không

#### Class “Lobby”

Table

Description automatically generated

**Hình 4-50: Biểu đồ lớp Dock**

**Đặc tả các thuộc tính**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default* | *Description* |
| 1 | reset | boolean | false | Reset |
| 2 | closeButton | Button |  | Nút Close |

**Bảng 4-58: Đặc tả các thuộc tính của Lobby Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | handleMainScreenClick | void | Xử lý sự kiện nhấn MainScreen |

**Bảng 4-59: Đặc tả các phương thức của Lobby**

*Parameter*:

* Không

*Exception*:

* Không

**Method** Không **State** Không

#### Class “MainScreen”

Text, table

Description automatically generated with medium confidence

**Hình 4-51: Biểu đồ lớp MainScreen**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | rentBikeButton | Button | Nút thuê xe |
| 2 | returnBikeButton | Button | Nút trả xe |
| 3 | viewDockListButton | Button | Nút xem danh sách bãi xe |
| 4 | viewRentalBikeButton | Button | Nút xem xe đã thuê |

**Bảng 4-60: Đặc tả các thuộc tính của MainScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | initialize | Dock | Khởi tạo MainScreen |
| 2 | resetDockInfo | float | Reset thông tin bãi xe |
| 3 | handleRentButtonClick | Bike | Xử lý sự kiện nhấn nút thuê xe |
| 4 | handleReturnButtonClick | String | Xử lý sự kiện nhấn nút trả xe |
| 5 | handleViewDockListClick | String | Xử lý sự kiện nhấn nút xem danh sách bãi xe |
| 6 | handleViewRentalBikeClick | boolean | Xử lý sự kiện nhấn nút ViewRentalBike |

**Bảng 4-61: Đặc tả các phương thức của MainScreen**

*Parameter*:

* url – đường dẫn tới fxml scene
* resourceBundle – tài nguyên đóng gói để khởi tạo fxml scene (sử dụng thư viện)

*Exception*:

* Không

**Method** Không **State** Không

#### Class “RentalBikeScreen”

Text

Description automatically generated

**Hình 4-52: Biểu đồ lớp RentalBikeScreen**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | bikeCode | String | Mã xe |
| 2 | type | String | Loại xe |
| 3 | remainBattery | int | Phần trăm pin còn lại |
| 4 | rentDuration | int | Thời gian thuê xe |
| 5 | rentCost | String | Giá thuê xe |
| 6 | OK | Button | Nút OK |

**Bảng 4-62: Đặc tả các thuộc tính của RentalBikeScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | setRentalBikeInfo | void | Đặt thông tin trả xe |
| 2 | processOKClick | void | Xử lý sự kiện click nút OK |

**Bảng 4-63: Đặc tả các phương thức của RentalBikeScreen**

*Parameter*:

* rentalCode – mã thuê xe

*Exception*:

* Không

**Method** Không **State** Không

#### Class “RentBikeInfoScreen”

Text, table

Description automatically generated

**Hình 4-53: Biểu đồ lớp RentBikeInfoScreen Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | textArea | TextField | Khu vực nhập barcode |
| 2 | confirmButton | Button | Nút xác nhận |
| 3 | backButton | Button | Nút Back |

**Bảng 4-64: Đặc tả các thuộc tính của RentBikeInfoScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | initialize | void | Khởi tạo |
| 2 | setBikeInfo | void | Đặt thông tin xe |
| 3 | displayCardScreen | void | Hiển thị CardScreen |
| 4 | handleBackButtonClick | void | Xử lý sự kiện click nút Back |

**Bảng 4-65: Đặc tả các phương thức của RentBikeInfoScreen**

*Parameter*:

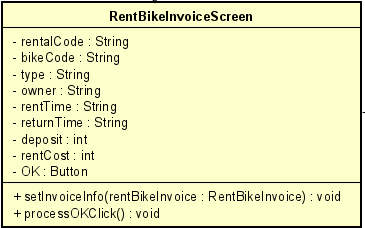
* url – đường dẫn tới fxml scene
* resourceBundle – tài nguyên đóng gói để khởi tạo fxml scene (sử dụng thư viện)
* bike – xe
* stage – Stage để gọi tới RentBikeInfoScreen

*Exception*:

* Không

**Method** Không **State** Không

#### Class “RentBikeInvoiceScreen”



**Hình 4-54: Biểu đồ lớp RentBikeInvoiceScreen Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | rentalCode | String | Mã thuê xe |
| 2 | bikeCode | String | Mã xe |
| 3 | type | String | Loại xe |
| 4 | owner | String | Người sở hữu |
| 5 | rentTime | String | Thời gian thuê |
| 6 | returnTime | String | Thời gian trả |
| 7 | deposit | int | Tiền cọc |
| 8 | rentCost | int | Giá thuê xe |
| 9 | OK | Button | Nút OK |

**Bảng 4-66: Đặc tả các thuộc tính của RentBikeInvoiceScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | setInvoiceInfo | void | Đặt thông tin hóa đơn |
| 2 | processOKClick | void | Xử lý sự kiện click nút OK |

**Bảng 4-67: Đặc tả các phương thức của RentBikeInvoiceScreen**

*Parameter*:

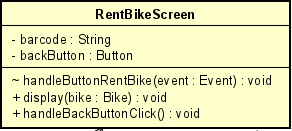
* rentBikeInvoice – hóa đơn thuê xe

*Exception*:

* Không

**Method** Không **State** Không

#### Class “RentBikeScreen”



**Hình 4-55: Biểu đồ lớp RentBikeScreen**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | barcode | String | Mã barcode |
| 2 | backButton | Button | Nút Back |

**Bảng 4-68: Đặc tả các thuộc tính của RentBikeScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | handleButtonRentBike | void | Xử lý sự kiện nhấn nút thuê xe |
| 2 | display | void | Hiển thị màn hình nhập thẻ |
| 3 | handleBackButtonClick | void | Xử lý sự kiện nhấn nút Back |

**Bảng 4-69: Đặc tả các phương thức của RentBikeScreen**

*Parameter*:

* event – sự kiện bắt được khi nhấn nút, sử dụng thư viện của JavaFX
* bike – xe muốn thuê

*Exception*:

* Không

**Method** Không **State** Không

#### Class “ReturnBikeScreen”

Table

Description automatically generated

**Hình 4-56: Biểu đồ lớp ReturnBikeScreen**

**Đặc tả các thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Description* |
| 1 | rentalCode | String | Mã thuê xe |
| 2 | backButton | Button | Nút Back |

**Bảng 4-70: Đặc tả các thuộc tính của ReturnBikeScreen Đặc tả các phương thức**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | initialize | void | Khởi tạo |
| 2 | handleDoubleClickOnDockList | void | Xử lý nhấn 2 lần vào danh sách bãi xe |
| 3 | getCardInfoForReturnBike | Card | Lấy thông tin thẻ |
| 4 | processReturnBike | void | Xử lý trả xe |
| 5 | showRentBikeInvoiceInfo | void | Hiển thị thông tin hóa đơn thuê xe |
| 6 | setDockInfo | void | Đặt thông tin cho bãi xe |
| 7 | getDockFromString | Dock | Lấy thông tin bãi xe từ String |
| 8 | handleBackButtonClick | void | Xử lý nhấn nút Back |

**Bảng 4-71: Đặc tả các phương thức của ReturnBikeScreen**

*Parameter*:

* url – đường dẫn tới fxml scene
* rb – tài nguyên đóng gói để khởi tạo fxml scene (sử dụng thư viện)
* rentalCode – mã thuê xe
* card – thẻ
* d – bãi xe
* rentBikeInvoice – hóa đơn thuê xe

*Exception*:

* Không

**Method**

Không

**State**

Không