TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công nghệ thông tin và Truyền thông

Tài liệu thiết kế phần mềm

(Software Design Document – SDD)

PHẦN MỀM THUÊ XE ĐẠP THEO GIỜ

Môn: Thiết kế và xây dựng phần mềm

Nhóm 6

Phạm Trung Kiên 20170088

Phạm Minh Khiêm 20170084

Lê Vũ Lợi 20173240

*Hà Nội, ngày 8 tháng 10 năm 2020*

Table of Contents

[Table of Contents 2](#_Toc59289786)

[1 Introduction 4](#_Toc59289787)

[1.1 Objective 4](#_Toc59289788)

[1.2 Scope 4](#_Toc59289789)

[1.3 Glossary 4](#_Toc59289790)

[1.4 References 4](#_Toc59289791)

[2 System Architecture and Architecture Design 5](#_Toc59289792)

[2.1 Architectural Patterns 5](#_Toc59289793)

[2.2 Interaction Diagrams 5](#_Toc59289794)

[2.3 Analysis Class Diagrams 7](#_Toc59289795)

[2.4 Unified Analysis Class Diagram 8](#_Toc59289796)

[2.5 Security Software Architecture 8](#_Toc59289797)

[3 Detailed Design 10](#_Toc59289798)

[3.1 Interface Design 10](#_Toc59289799)

[3.1.1 User Interface design 10](#_Toc59289800)

[3.1.1.1 Screen Configuration Standardization 10](#_Toc59289801)

[3.1.1.2 Screen Transition Diagrams 12](#_Toc59289802)

[3.1.1.3 Screen Specifications 12](#_Toc59289803)

[3.1.2 System Interface design : 21](#_Toc59289804)

[3.1.2.1 Interface design for subsystem Interbank : 21](#_Toc59289805)

[3.1.2.2 Subsystem design : 22](#_Toc59289806)

[3.1.2.2.2 Document subsystem elements 23](#_Toc59289807)

[3.1.2.2.3 Combined subsystem design 24](#_Toc59289808)

[3.2 Data Modeling 25](#_Toc59289809)

[3.2.1 Conceptual Data Modeling 25](#_Toc59289810)

[3.2.2 Database Design 25](#_Toc59289811)

[3.3 Non-Database Management System Files 27](#_Toc59289812)

[3.4 Class Design 28](#_Toc59289813)

[3.4.1 General Class Diagram 28](#_Toc59289814)

[3.4.2 Class Diagrams 29](#_Toc59289815)

[3.4.3 Class Design 29](#_Toc59289816)

[5 Design Considerations 49](#_Toc59289817)

[5.4 Goals and Guidelines 49](#_Toc59289818)

[5.5 Architectural Strategies 49](#_Toc59289819)

[5.6 Coupling and Cohesion 50](#_Toc59289820)

[5.7 Design Principles 50](#_Toc59289821)

[5.8 Design Patterns 50](#_Toc59289822)

# Introduction

## Objective

* Mục tiêu của tài liệu này là đưa ra thiết kế phần mềm cho ứng dụng EcoBikeRental cho thuê xe đạp theo giờ với nhiều bãi để xe để thuê/trả xe tự động trong khu đô thị Ecopark.
* Tài liệu này phục vụ cho designers, developers và testers của hệ thống có thể dễ dàng đi tới các bước tiếp theo của quy trình phát triển ứng dụng này.

## Scope

Thuê xe đạp dạo quanh là một trong những dịch vụ được thu hút nhất tại khu đô thị sinh thái Ecopark. Hiện nay có 2 điểm cho thuê và đỗ xe đạp tại khu đô thị. Để dịch vụ này tiếp tục phát triển mở rộng ra, cần giải quyết khâu hạ tầng, bao gồm hệ thống làn đường cho xe đạp, điểm dừng, trông giữ, bảo quản, và đặc biệt là hệ thống thông tin thuê xe và trả xe tự động có thể hoạt động 24/7

## Glossary

|  |  |  |
| --- | --- | --- |
| STT | Thuật ngữ, từ viêt tắt | Giải thích |
| 1 | API | viết tắt của Application Programming Interface, là phần mềm trung gian cho phép kết nối 2 ứng dụng với nhau |
|  |  |  |
|  |  |  |

## References

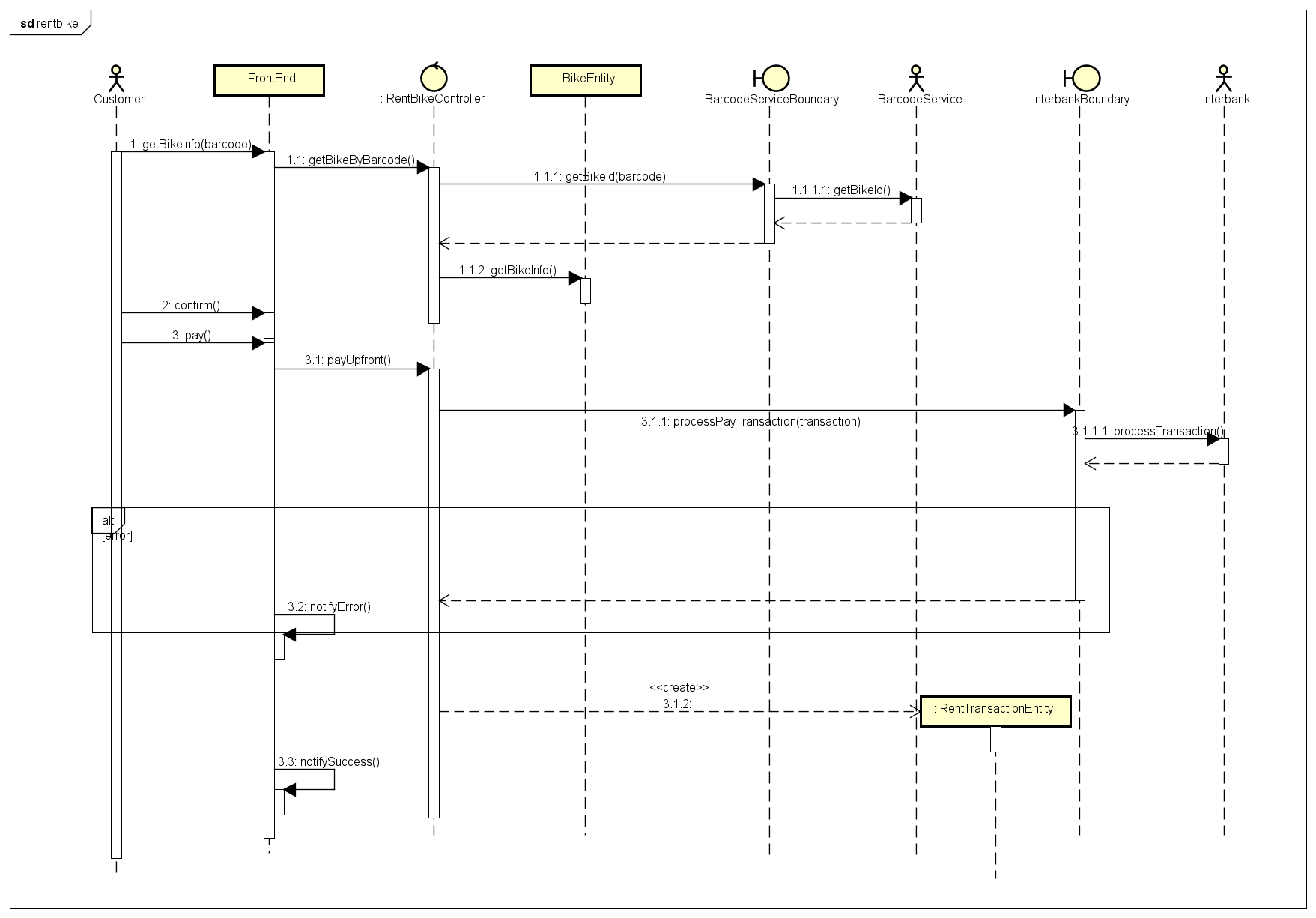
|  |  |
| --- | --- |
| STT | Tên tài liệu |
| 1 | D. Budgen. Software Design, 2nd Edition. Addison-Wesley. 2004 |
|  |  |
|  |  |

# System Architecture and Architecture Design

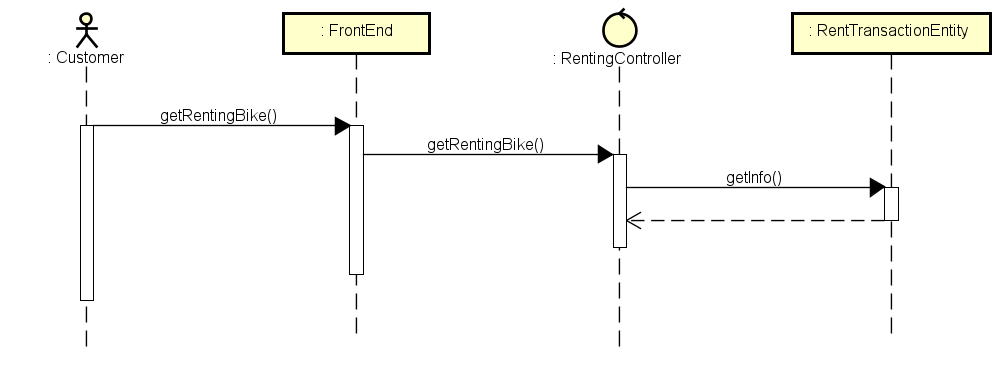
## Architectural Patterns

*Nhóm chọn thiết kế theo kiến trúc Frontend – Backend*

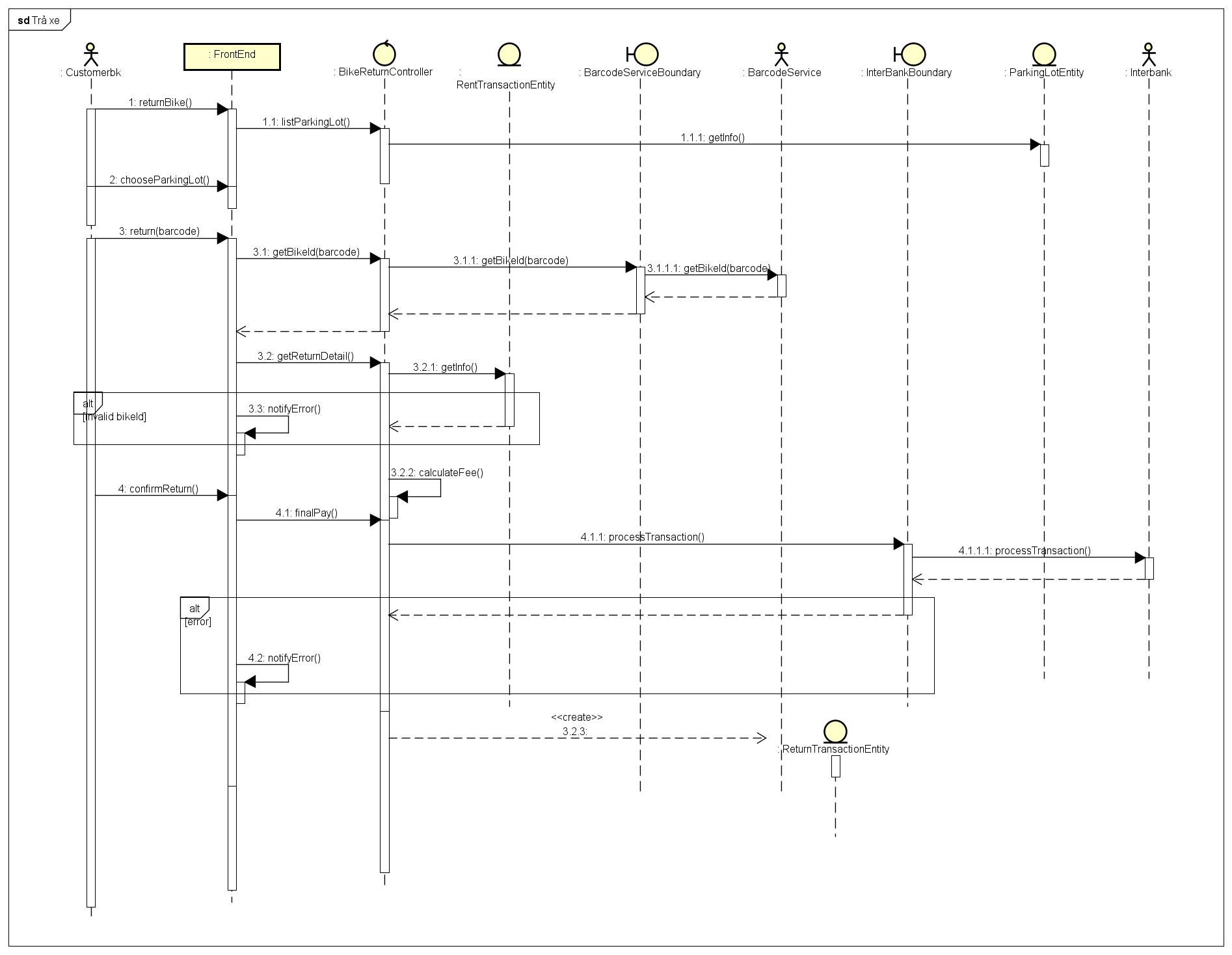
## Interaction Diagrams



Hình : Biểu đồ trình tự cho thuê xe

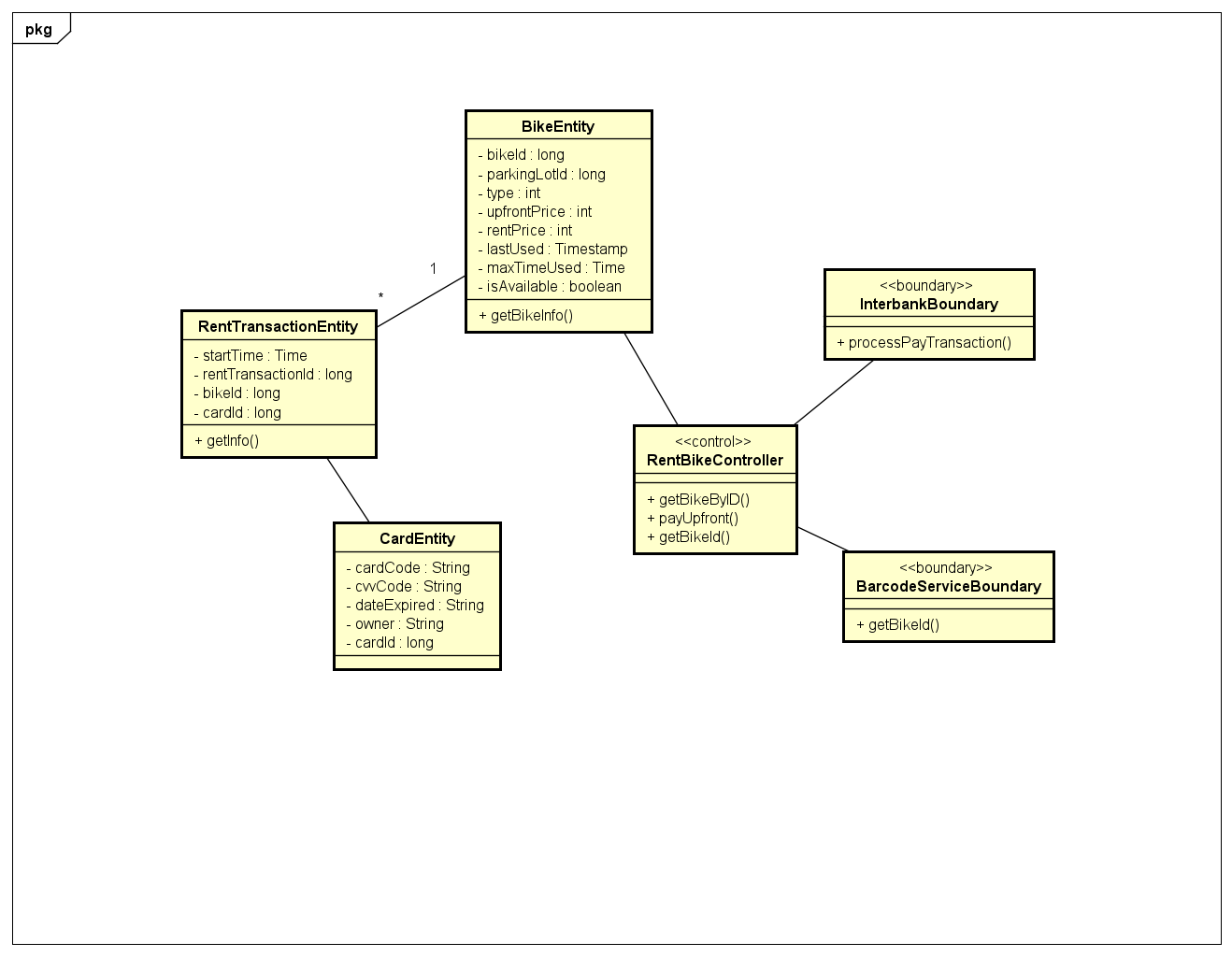


Hình : Biểu đồ trình tự cho xem xe đang thuê

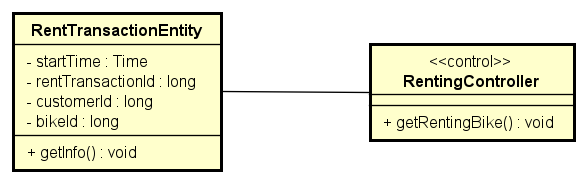


Hình : Biểu đồ trình tự cho trả xe

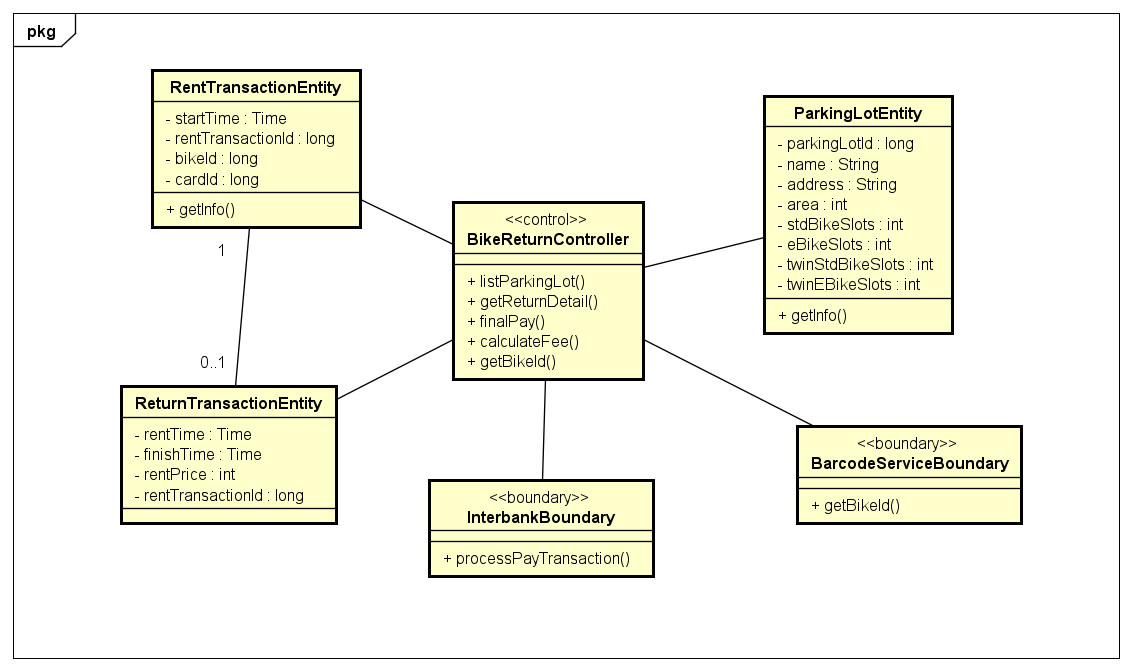
## Analysis Class Diagrams



Hình : Biểu đồ cho thuê xe

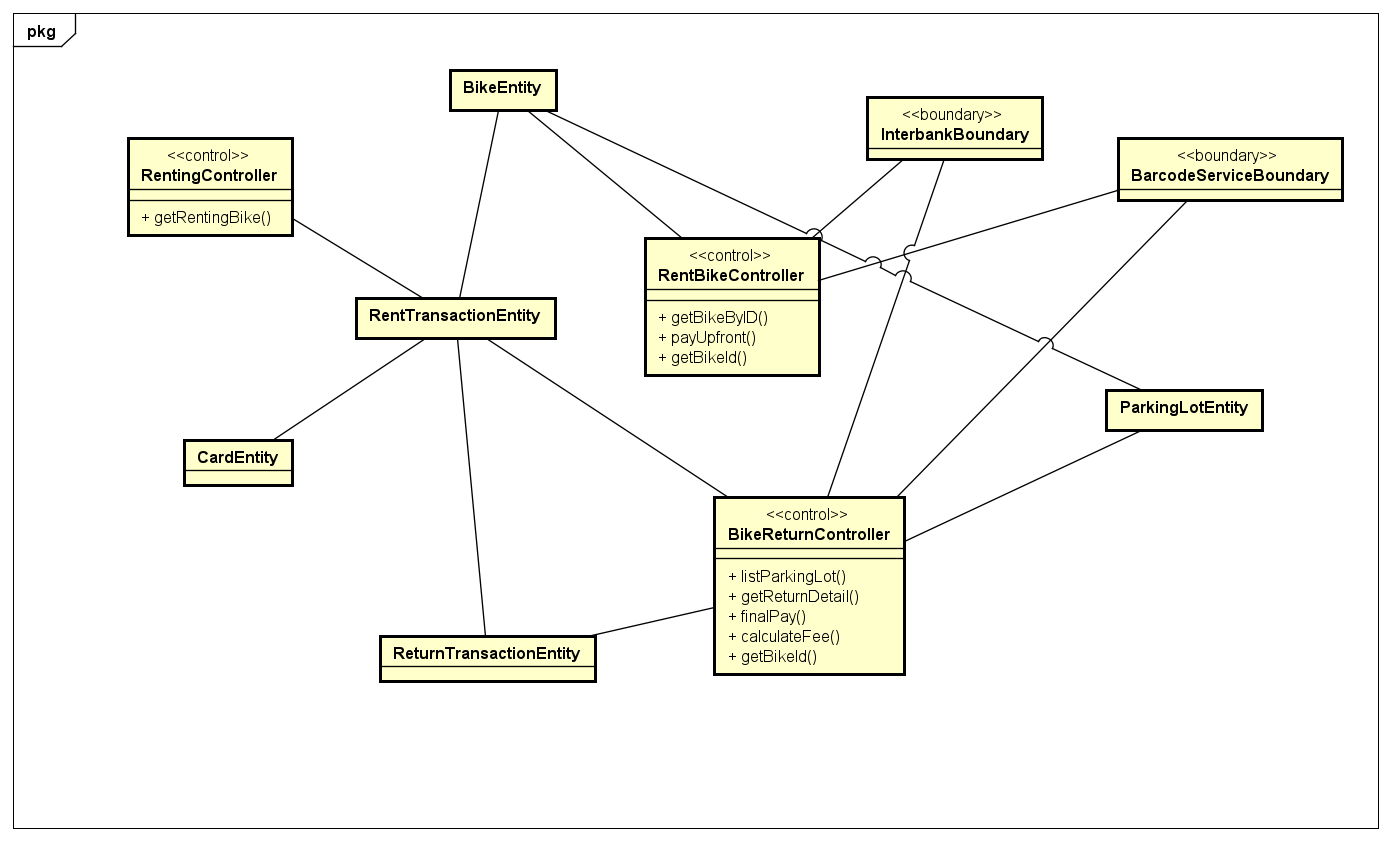


Hình : Biểu đồ cho xem xe đang thuê



Hình : Biểu đồ cho trả xe

## Unified Analysis Class Diagram



## Security Software Architecture

*<Describe the software components and configuration supporting the security and privacy of the system. Specify the architecture for (1) authentication to validate user identity before allowing access to the system;(2) authorization of users to perform functional activity once logged into the system, (3) encryption protocol to support the business risks and the nature of information, and (4) logging and auditing design, if required.>*

# Detailed Design

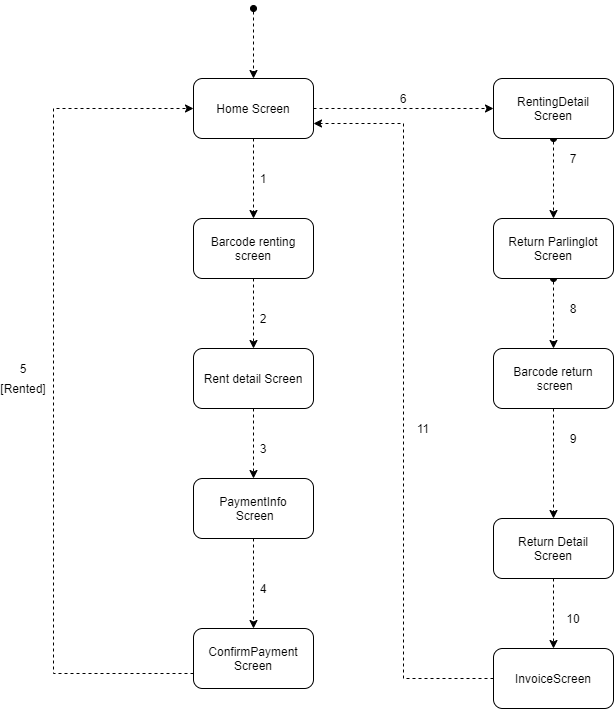
## Interface Design

### User Interface design

### Screen Configuration Standardization

|  |  |  |  |
| --- | --- | --- | --- |
| Home screen | Barcode renting screen | Rent detail screen | Payment info screen |
|  |  |  |  |
| Confirm payment screen | Home rented screen | Renting detail screen | Return parkinglot screen |
|  |  |  |  |
| Barcode return screen | Return detailed screen | Invoice screen | Bike not found alert |
|  |  |  |  |
| Success renting alert | Rent payment failed alert | Barcode return error | Payment error alert |
|  |  |  |  |

### Screen Transition Diagrams



### Screen Specifications

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Home screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Khu vực hiển thị các bãi đỗ xe* | *Initial* | *Hiển thị tên và địa chỉ của các bãi đỗ xe gần nhất* | |
| *Nút Xem* | *Click* | *Xem thông tin chi tiết bãi đỗ xe* | |
| *Nút Thuê xe* | *Click* | *Hiển thị màn hình nhập barcode* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Barcode renting screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Nút Thuê xe* | *Click* | *Thuê xe có barcode tương ứng* | |
| *Khu vực nhập mã vạch* | *Type* | *Nhập mã vạch* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Rent detail screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Khu vực hiển thị thông tin xe* | *Initial* | *Hiển thị thông tin chi tiết của xe tương ứng* | |
| *Khu vực hiển thị giá cọc* | *Initial* | *Hiển thị giá cọc* | |
| *Nút Tiếp tục* | *Click* | *Hiển thị màn hình thanh toán* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Payment info screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Khu vực nhập thông tin thanh toán* | *Initial* | *Nhập thông tin thanh toán* | |
| *Nút Thanh toán* | *Click* | *Thực hiện đặt cọc* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Confirm payment screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Khu vực hiển thị thông tin thanh toán* | *Initial* | *Hiển thị tên và địa chỉ của các bãi đỗ xe gần nhất* | |
| *Nút Thanh toán* | *Click* | *Xác nhận giao dịch* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Home rented screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Nút Bạn đang có 1 xe đang thuê* | *Click* | *Xem thông tin xe đang thuê* | |
| *Nút Xem* | *Click* | *Xem thông tin chi tiết bãi đỗ xe* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Renting detail screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Khu vực hiển thị thông tin xe đang thuê* | *Initial* | *Hiển thị thông tin chi tiết của xe đang thuê* | |
| *Nút Tạm dừng* | *Click* | *Tạm dừng thuê xe* | |
| *Nút Trả xe* | *Click* | *Hiển thị màn hình trả xe* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Return parkinglot screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Khu vực hiển thị các bãi đỗ xe* | *Initial* | *Hiển thị tên và địa chỉ của các bãi đỗ xe gần nhất* | |
| *Nút Chọn* | *Click* | *Chọn bãi đỗ xe để trả* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Barcode return screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Khu vực nhập mã vạch* | *Initial* | *Nhập mã vạch xe muốn trả* | |
| *Nút Trả xe* | *Click* | *Hiển thị màn hình trả xe* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Home screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Khu vực hiển thị thông tin trả xe* | *Initial* | *Hiển thị thông tin chi tiết trả xe* | |
| *Nút Thanh toán* | *Click* | *Xác nhận trả xe* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Invoice screen* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Khu vực hiển thị thông tin giao dịch trả xe* | *Initial* | *Hiển thị thông tin giao dịch trả xe* | |
| *Nút Quay về trang chủ* | *Click* | *Về trang chủ* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Bike not found alert* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Thông báo không thấy xe* | *Initial* | *Thông báo không thấy xe phù hợp mã vạch* | |
| *Nút OK* | *Click* | *Đóng thông báo* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Success renting alert* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Thông báo thuê xe thành công* | *Initial* | *Thông báo đã thuê xe thành công* | |
| *Nút OK* | *Click* | *Đóng thông báo* | |
|  |  |  | |

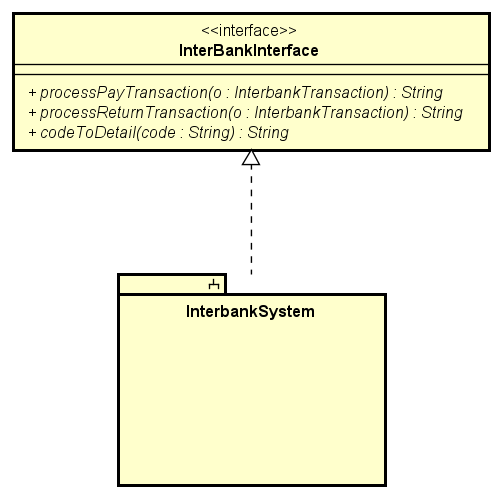
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Rent payment failed alert* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Thông báo thanh toán lỗi* | *Initial* | *Thông báo thanh toán gặp lỗi khi thuê xe* | |
| *Nút OK* | *Click* | *Đóng thông báo* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Barcode return error* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Thông báo mã vạch không khớp* | *Initial* | *Thông báo mã vạch nhập vào không khớp với xe nào* | |
| *Nút OK* | *Click* | *Đóng thông báo* | |
|  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *EcoBike* | | *Date of creation* | *Approved by* | *Reviewed by* | *Person in charge* |
| *Screen specification* | *Payment error alert* |  |  |  | *Phạm Minh Khiêm* |
|  | | *Control* | *Operation* | *Function* | |
| *Thông báo thanh toán lỗi* | *Initial* | *Thông báo thanh toán gặp lỗi khi trả xe* | |
| *Nút OK* | *Click* | *Đóng thông báo* | |
|  |  |  | |

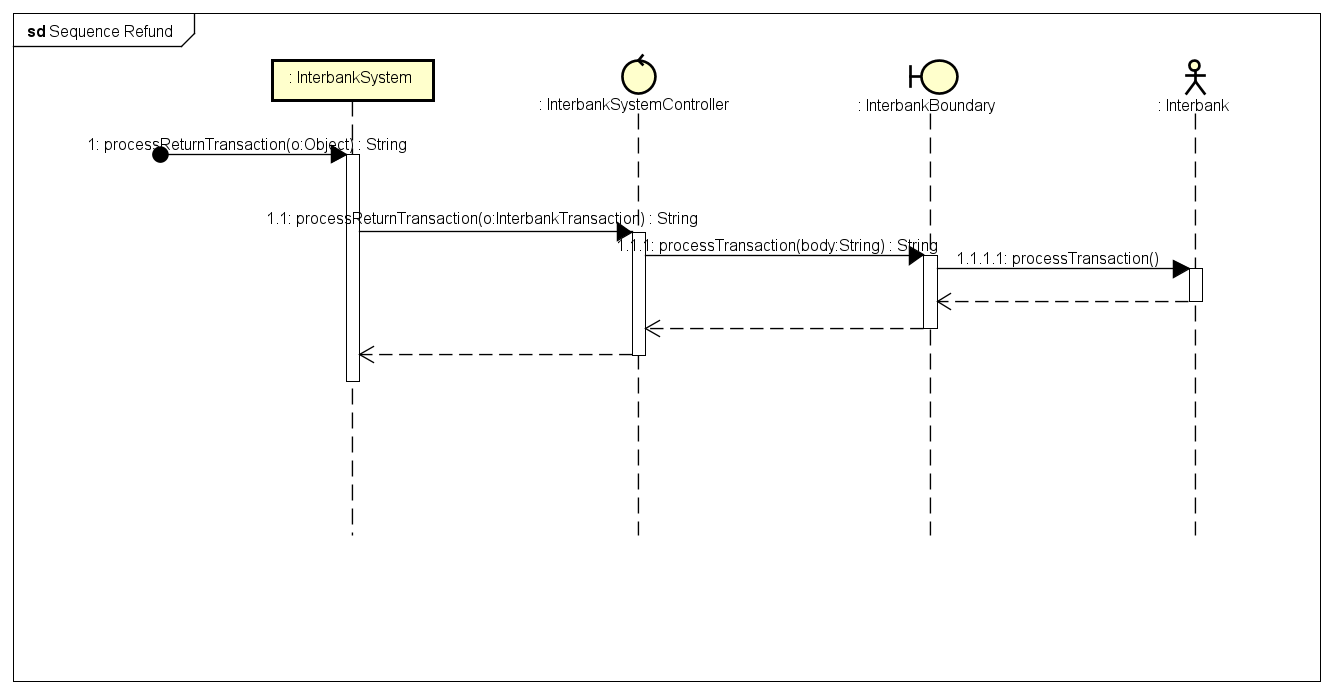
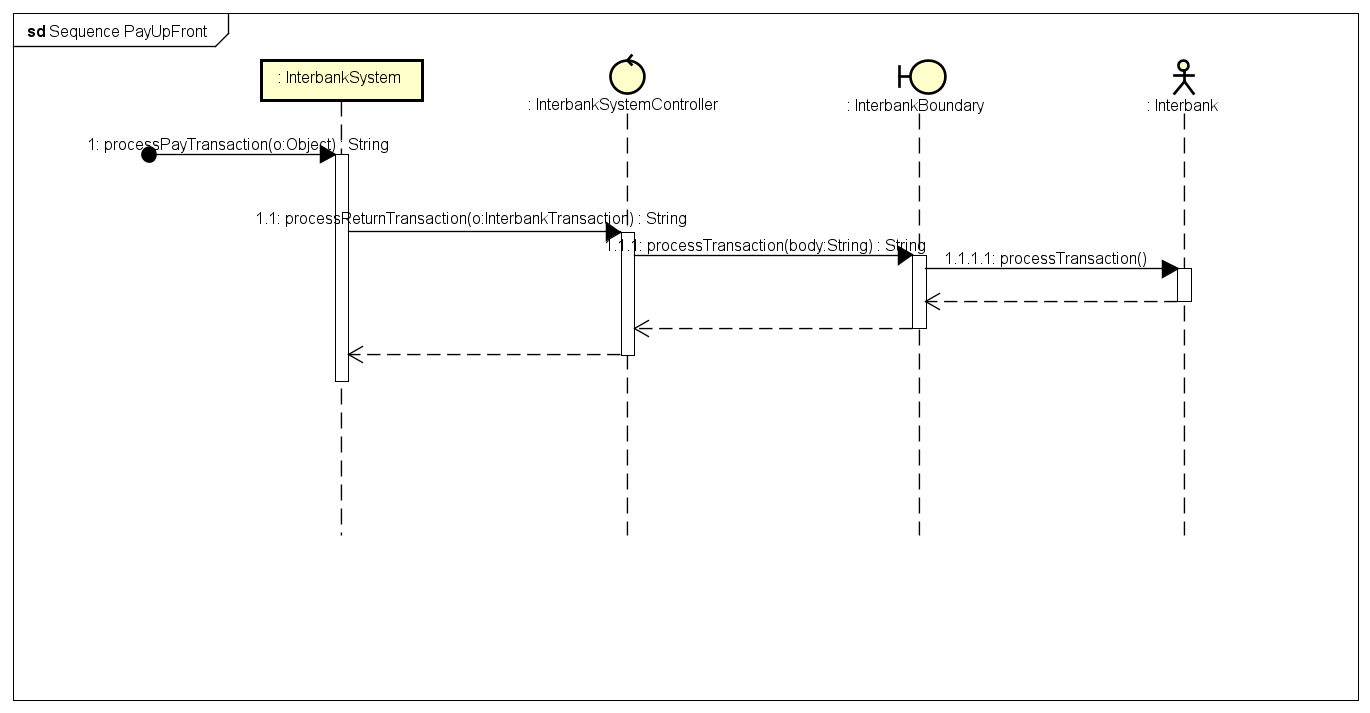
### System Interface design :

### Interface design for subsystem Interbank :

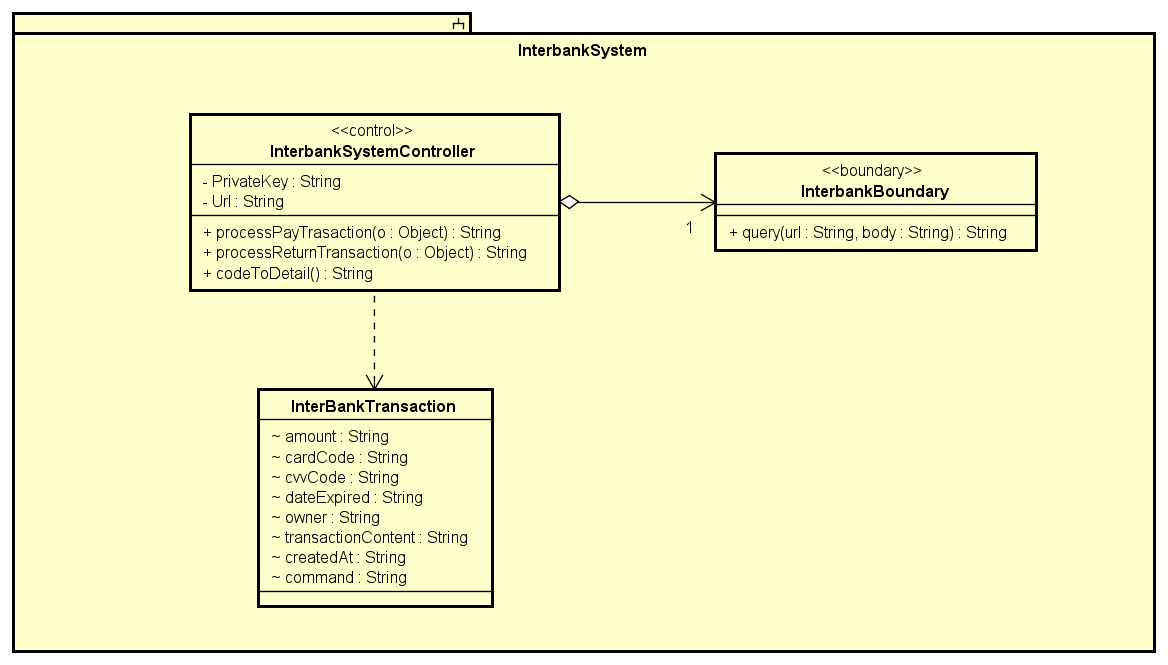


### Subsystem design :

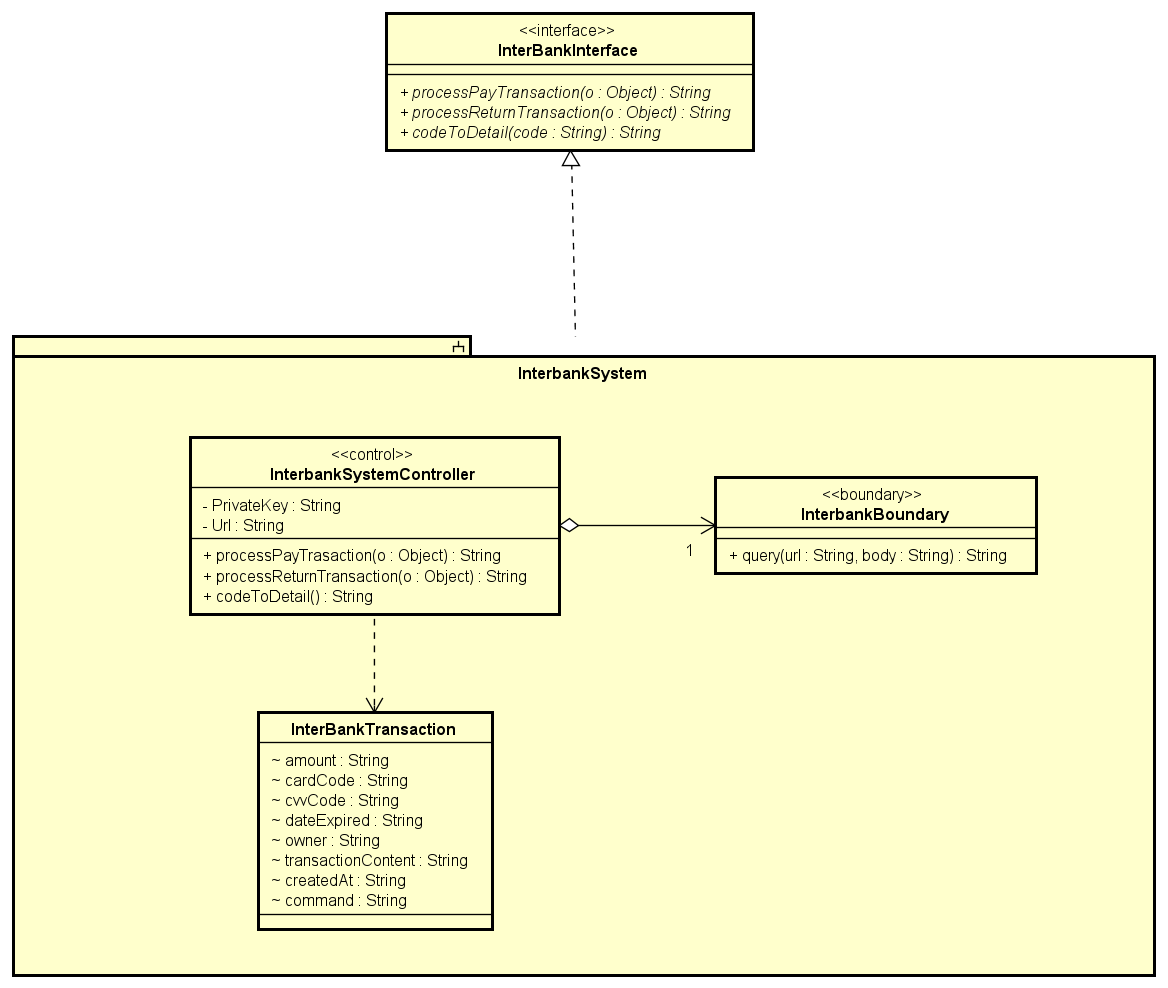
* + - * 1. **Distribute subsystem behavior to subsystem elements :**



### Document subsystem elements

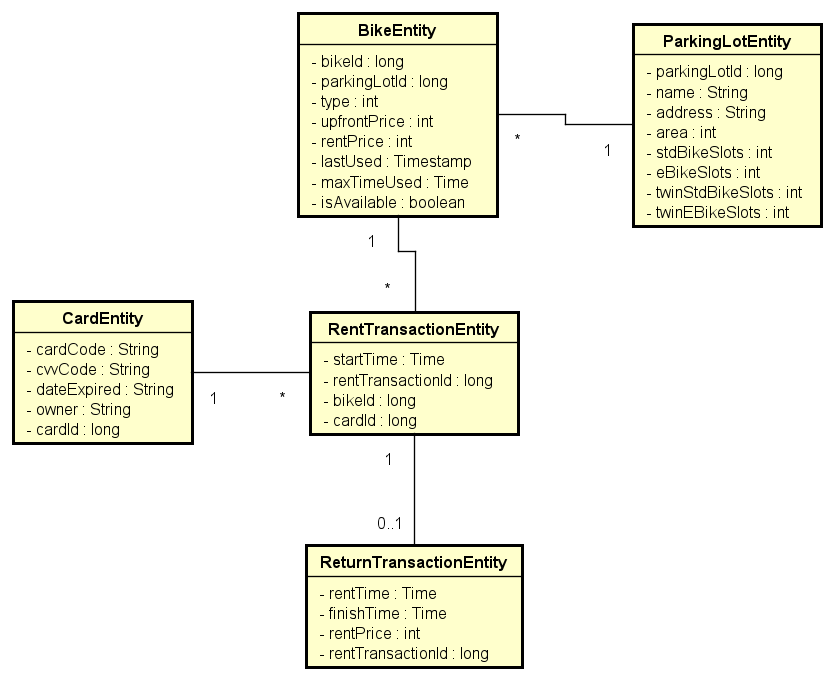


### Combined subsystem design



## Data Modeling

### Conceptual Data Modeling

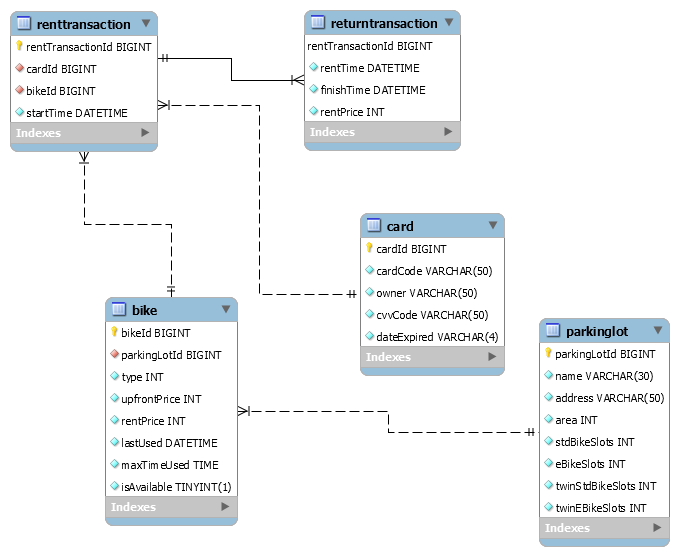


### Database Design

#### Database Management Systems

*Hệ quản trị dữ liệu mySQL*

#### Logical Data Model



#### Physical Data Model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| parkingLotId | bigint | NO | PRI | NULL | auto\_increment |
| name | varchar(30) | NO |  | NULL |  |
| address | varchar(50) | NO |  | NULL |  |
| area | int | NO |  | NULL |  |
| stdBikeSlots | int | NO |  | NULL |  |
| eBikeSlots | int | NO |  | NULL |  |
| twinStdBikeSlots | int | NO |  | NULL |  |
| twinEBikeSlots | int | NO |  | NULL |  |

Bang:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| cardId | bigint | NO | PRI | NULL | auto\_increment |
| cardCode | varchar(50) | NO |  | NULL |  |
| owner | varchar(50) | NO |  | NULL |  |
| cvvCode | varchar(50) | NO |  | NULL |  |
| dateExpired | varchar(4) | NO |  | NULL |  |

Bang:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| bikeId | bigint | NO | PRI | NULL | auto\_increment |
| parkingLotId | bigint | NO | MUL | NULL |  |
| type | int | NO |  | NULL |  |
| upfrontPrice | int | NO |  | NULL |  |
| rentPrice | int | NO |  | NULL |  |
| lastUsed | datetime | NO |  | NULL |  |
| maxTimeUsed | time | NO |  | NULL |  |
| isAvailable | tinyint(1) | NO |  | NULL |  |

Bang:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| rentTransactionId | bigint | NO | PRI | NULL | auto\_increment |
| cardId | bigint | NO | MUL | NULL |  |
| bikeId | bigint | NO | MUL | NULL |  |
| startTime | datetime | NO |  | NULL |  |

Bang:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Extra |
| rentTransactionId | bigint | NO | PRI | NULL |  |
| rentTime | datetime | NO |  | NULL |  |
| finishTime | datetime | NO |  | NULL |  |
| rentPrice | int | NO |  | NULL |  |

## Non-Database Management System Files

*<Provide the detailed description of all non-DBMS files if any and include a narrative description of the usage of each file that identifies if the file is used for input, output, or both, and if the file is a temporary file. Also provide an indication of which modules read and write the file and include file structures (refer to the data dictionary). As appropriate, the file structure information should include the following:*

*• Record structures, record keys or indexes, and data elements referenced within the records*

*• Record length (fixed or maximum variable length) and blocking factors*

*• Access method (e.g., index sequential, virtual sequential, random access, etc.)*

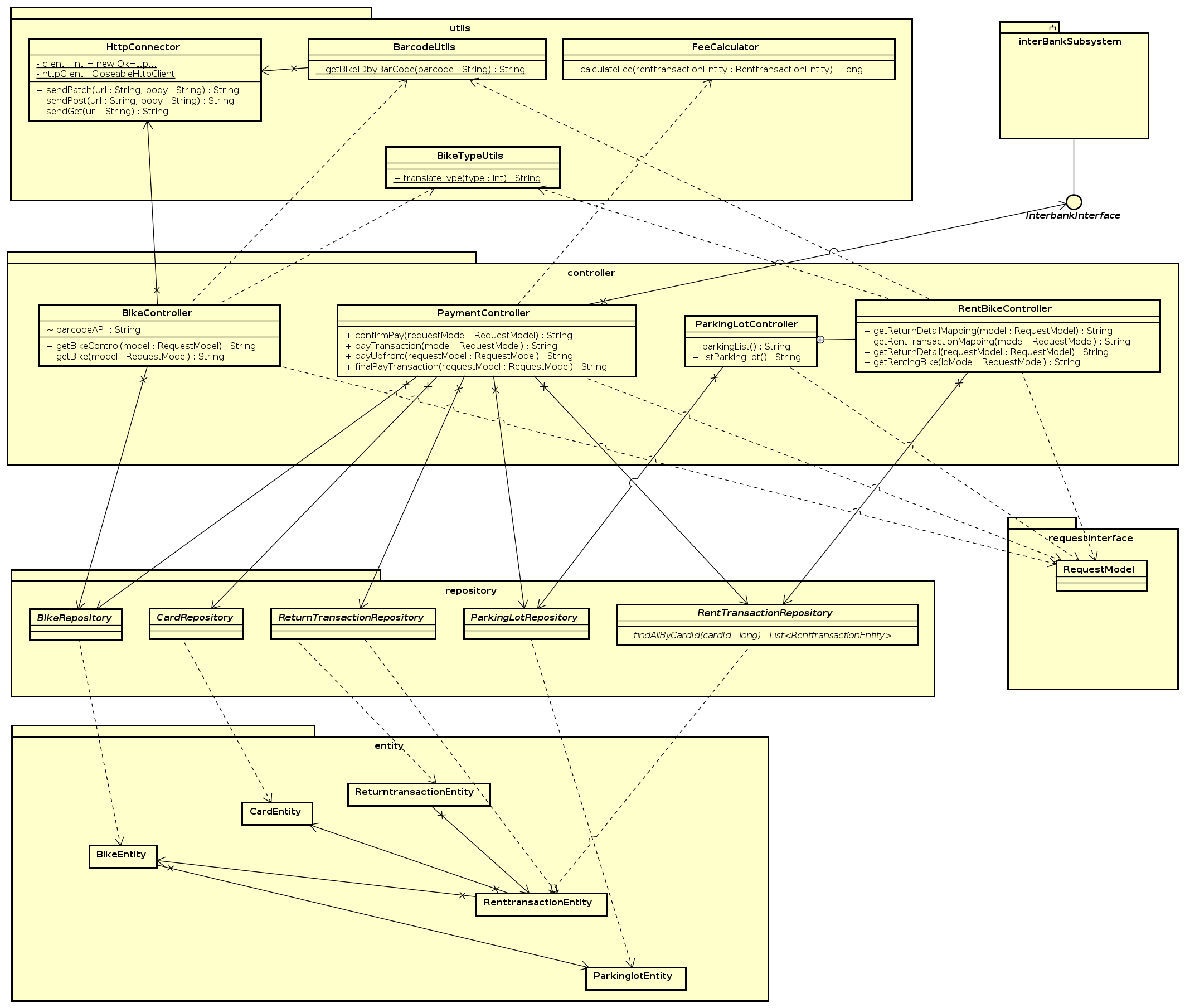
*• Estimate of the file size or volume of data within the file, including overhead resulting from file access methods*

*• Definition of the update frequency of the file (If the file is part of an online transaction-based system, provide the estimated number of transactions per unit of time, and the statistical mean, mode, and distribution of those transactions.)*

*• Backup and recovery specifications>*

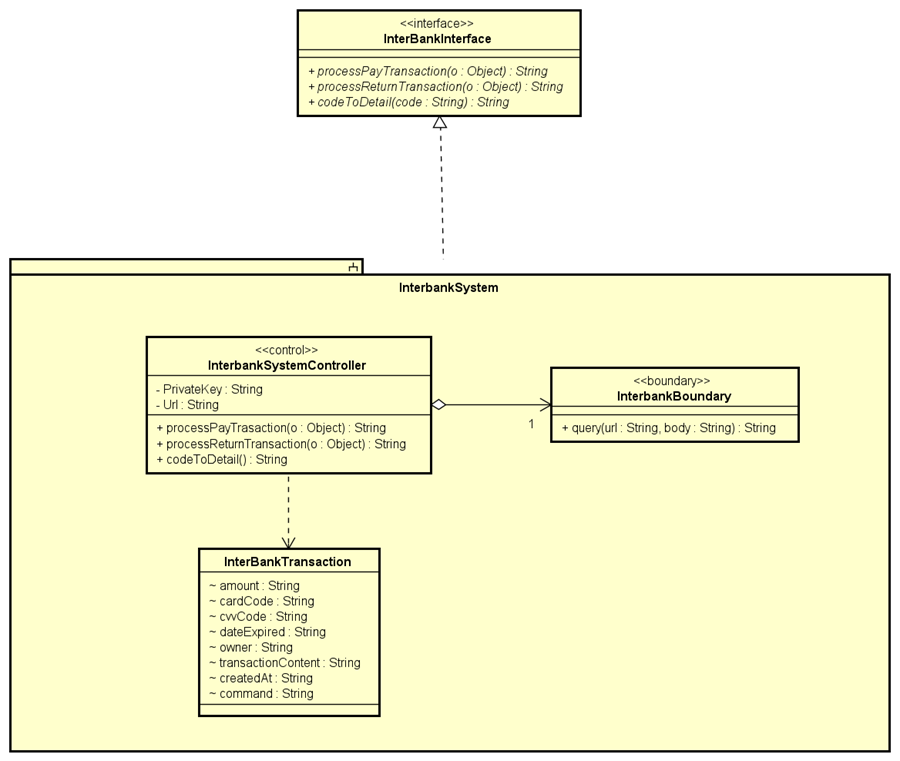
## Class Design

### General Class Diagram



### Class Diagrams

#### Class Diagram for Subsystem Interbank



### Class Design

#### Gói Controller

##### Lớp BikeController

**Mục đích sử dụng**

Điều khiển các tác vụ liên quan đến đối tượng xe.

Định nghĩa REST API tương ứng để front-end có thể tương tác đến.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | bikeRepository | BikeRepository |  | Đối tượng sử dụng để truy vấn bảng Bike trong cơ sở dữ liệu |
| 2 | barcodeAPI | String | https://barcodeservicebykv2.herokuapp.com/barcode | Đường dẫn đến API chuyển mã vạch |
| 3 | httpConnector | HttpConnector |  | gửi request đến API barcode yêu cầu chuyển mã vạch thành mã xe |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | getBikeControl | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Tạo API để front-end truy vấn thông tin xe:   * Method: POST * Path: /getBikeByBarcode | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Đóng gói các trường:   * barcode |
|  | | | | |
| **2** | ***Tên*** | getBikeByBarcode | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Lấy thông tin xe tương ứng với barcode truyền vào | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Tương tự getBikeControl |

**Method**

Không

**State**

Không

##### Lớp PaymentController

**Mục đích sử dụng**

Lớp điều khiển các tác vụ liên quan đến thanh toán hóa đơn thuê và trả xe.

Định nghĩa REST API tương ứng để front-end có thể tương tác đến.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | parkinglotRepository | ParkingLotRepository |  | Đối tượng sử dụng để truy vấn bảng ParkingLot trong cơ sở dữ liệu |
| 2 | bikeRepository | BikeRepository |  | Đối tượng sử dụng để truy vấn bảng Bike trong cơ sở dữ liệu |
| 3 | rentTransactionRepository | RentTransactionRepository |  | Đối tượng sử dụng để truy vấn bảng RentTransaction trong cơ sở dữ liệu |
| 4 | returnTransactionRepository | ReturnTransactionRepository |  | Đối tượng sử dụng để truy vấn bảng ReturnTransaction trong cơ sở dữ liệu |
| 5 | cardRepository | CardRepository |  | Đối tượng sử dụng để truy vấn bảng Card trong cơ sở dữ liệu |
| 6 | interbankSubsystem | InterbankInterface |  | Đối tượng sử dụng để giao tiếp với Interbank API |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | payUpFrontControl | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Định nghĩa API để yêu cầu front-end thanh toán cọc:   * Method: POST * Path: /payUpFront | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Đóng gói các trường:   * barcode * cardCode * cardOwner * cvv * expireDate |
|  | | | | |
| **2** | ***Tên*** | finalPayControl | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Định nghĩa API yêu cầu thanh toán trả xe | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Đóng gói các trường:   * barcode * cardID * parkinglotID |
|  | | | | |
| **3** | ***Tên*** | payUpFront | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Phương thức dùng để thanh toán cọc | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Tương tự payUpFrontControl |
|  | | | | |
| **4** | ***Tên*** | finalPay | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Phương thức thanh toán khi người dùng trả xe | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Tương tự finalPayControl |

**Method**

Không

**State**

Không

##### Lớp ParkingLotController

**Mục đích sử dụng**

Lớp điều khiển các tác vụ liên quan đến truy vấn các bãi xe.

Định nghĩa REST API tương ứng để front-end có thể tương tác đến.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | parkinglotRepository | ParkingLotRepository |  | Đối tượng sử dụng để truy vấn bảng ParkingLot trong cơ sở dữ liệu |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | parkingList | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Định nghĩa API để front-end truy vấn thông tin xe:   * Method: GET * Path: /listParkingLot | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
|  |  |  |
|  | | | | |
| **2** | ***Tên*** | listParkingLot | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Lấy thông tin danh sách các bãi xe trong hệ thống | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
|  |  |  |

**Method**

Không

**State**

Không

##### Lớp RentBikeController

**Mục đích sử dụng**

Lớp điều khiển các tác vụ liên quan đến thông tin thuê và trả xe.

Định nghĩa REST API tương ứng để front-end có thể tương tác đến.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | rentTransactionRepository | RentTransactionRepository |  | Đối tượng sử dụng để truy vấn bảng RentTransaction trong cơ sở dữ liệu |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | getReturnDetailControl | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Định nghĩa API truy vấn thông tin trả xe:   * Method: POST * Path: /returnDetail | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Đóng gói các trường:   * barcode * cardID |
|  | | | | |
| **2** | ***Tên*** | getRentTransactionControl | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Định nghĩa API truy vấn thông tin xe đang thuê:   * Method: POST * Path: /getRentTransaction | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Đóng gói các trường:   * cardID |
|  | | | | |
| **3** | ***Tên*** | getReturnDetail | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Lấy thông tin dùng để trả xe | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Tương tự getReturnDetailControl |
|  | | | | |
| **4** | ***Tên*** | getRentTransaction | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Lấy thông tin xe đang được thuê | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| model | RequestModel | Tương tự getRentTransactionControl |

**Method**

Không

**State**

Không

#### Gói repository

Sử dụng Java Persistence API để thực hiện các thao tác CRUD đến database bằng cách tạo ra các repository interface:

• BikeRepository

• CardRepository

• ParkingLotRepository

• RentTransactionRepository

• ReturnTransactionRepository

#### Gói entity

##### Lớp BikeEntity

**Mục đích sử dụng**

Định nghĩa một đối tượng Bike, mapping với các trường thông tin tương ứng trong CSDL.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | bikeId | long |  | Mã định danh xe |
| 2 | parkingLotId | long |  | Mã bãi xe |
| 3 | type | int |  | Loại xe |
| 4 | upfrontPrice | int |  | Giá cọc của xe |
| 5 | rentPrice | int |  | Giá thuê |
| 6 | lastUsed | Timestamp |  | Lần cuối sử dụng |
| 7 | maxTimeUsed | Time |  | Thời gian sử dụng tối đa (chỉ có với xe điện) |
| 8 | isAvailable | byte |  | Có đang được thuê hay không? |
| 8 | parkinglotByParkingLotId | ParkingLotEntity |  | Bãi xe tương ứng với xe |
| 9 | renttransactionsByBikeId | Collection<RentTransactionEntity> |  | Danh sách các giao dịch thuê xe tương ứng với xe |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | getParkingLotByParkingLotId | ***Kiểu dữ liệu trả về*** | ParkingLot |
| ***Mô tả*** | Lấy về bãi xe tương ứng với xe | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
|  |  |  |
|  | | | | |
| **2** | ***Tên*** | getRentTransactionsByBikeId | ***Kiểu dữ liệu trả về*** | Collection<RentTransactionEntity> |
| ***Mô tả*** | Lấy về danh sách các giao dịch thuê xe tương ứng với xe | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
|  |  |  |

**Method**

Không

**State**

Không

##### Lớp CardEntity

**Mục đích sử dụng**

Định nghĩa một đối tượng Card, mapping với các trường thông tin tương ứng trong CSDL.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | cardId | long |  | ID định danh thẻ mỗi lần thanh toán |
| 2 | cardCode | String |  | Mã định danh thẻ |
| 3 | owner | String |  | Tên chủ thẻ |
| 4 | cvvCode | String |  | Card Verification Value |
| 5 | dateExpired | String |  | Ngày hết hạn |
| 6 | renttransactionsByCardId | Collection<RentTransactionEntity> |  | Danh sách các giao dịch thuê xe sử dụng thẻ cardID |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | getRentTransactionByCardId | ***Kiểu dữ liệu trả về*** | Collection<RentTransactionEntity> |
| ***Mô tả*** | Lấy về danh sách giao dịch thuê xe sử dụng thẻ có cardID tương ứng | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
|  |  |  |

**Method**

Không

**State**

Không

##### Lớp ReturnTransactionEntity

**Mục đích sử dụng**

Định nghĩa một đối tượng ReturnTransaction, mapping với các trường thông tin tương ứng trong CSDL.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | rentTransactionId | long |  | Mã giao dịch thuê xe tương ứng |
| 2 | rentTime | Timestamp |  | Thời điểm bắt đầu thuê xe |
| 3 | finishTime | Timestamp |  | Thời điểm trả xe |
| 4 | rentPrice | int |  | Tổng tiền thuê xe |
| 5 | renttransactionsByRentTransactionId | RenttransactionEntity |  | Đối tượng RentTransactionEntity của giao dịch thuê xe tương ứng. |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | getRenttransactionsByRentTransactionId | ***Kiểu dữ liệu trả về*** | RentTransactionEntity |
| ***Mô tả*** | Lấy về đối tượng RentTransactionEntity của giao dịch thuê xe tương ứng | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
|  |  |  |

**Method**

Không

**State**

Không

##### Lớp ParkinglotEntity

**Mục đích sử dụng**

Định nghĩa một đối tượng ParkingLot, mapping với các trường thông tin tương ứng trong CSDL.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | parkingLotId | long |  | Mã bãi xe |
| 2 | name | String |  | Tên bãi xe |
| 3 | address | String |  | Địa chỉ bãi xe |
| 4 | area | int |  | Diện tích bãi xe |
| 5 | stdBikeSlots | int |  | Số lượng chỗ trống xe đạp đơn |
| 6 | eBikeSlots | int |  | Số lượng chỗ trống xe đạp điện đơn |
| 7 | twinStdBikeSlots | int |  | Số lượng chỗ trống xe đạp đôi |
| 8 | twinEBikeSlots | int |  | Số lượng chỗ trống của xe đạp điện đôi |
| 9 | bikesByParkingLotId | Collection<BikeEntity> |  | Danh sách các xe có trong bãi |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | getBikeByParkinglotId | ***Kiểu dữ liệu trả về*** | Collection<BikeEntity> |
| ***Mô tả*** | Lấy về danh sách các xe có trong bãi | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
|  |  |  |

**Method**

Không

**State**

Không

##### Lớp RentTransactionEntity

**Mục đích sử dụng**

Định nghĩa một đối tượng RentTransaction, mapping với các trường thông tin tương ứng trong CSDL.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | rentTransactionId | long |  | Mã giao dịch thuê xe |
| 2 | cardId | long |  | ID định danh thẻ mỗi lần thanh toán |
| 3 | bikeId | long |  | Mã định danh xe |
| 4 | startTime | Timestamp |  | Thời điểm bắt đầu thuê xe |
| 5 | cardByCardId | CardEntity |  | Thông tin thẻ tương ứng với cardId |
| 6 | bikeByBikeId | BikeEntity |  | Thông tin xe đang thuê tương ứng |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | getCardByCardId | ***Kiểu dữ liệu trả về*** | CardEntity |
| ***Mô tả*** | Lấy về thông tin thẻ tương ứng với cardId | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
|  |  |  |
|  | | | | |
| **2** | ***Tên*** | getBikeByBikeId | ***Kiểu dữ liệu trả về*** | BikeEntity |
| ***Mô tả*** | Lấy về thông tin xe tương ứng với xe được thuê | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
|  |  |  |

**Method**

Không

**State**

Không

#### Gói utils

##### Lớp HttpConnector

**Mục đích sử dụng**

Thực hiện HTTP connect, các method POST, GET, PATCH, ...

**Attribute**

Không

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | sendPatch | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Gửi một HTTP request dạng PATCH và lấy kết quả trả về | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| url | String | Đường dẫn tới máy chủ cần request |
| body | String | Nội dung gói tin gửi trong request |
|  | | | | |
| **2** | ***Tên*** | sendPost | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Gửi một HTTP request dạng POST và lấy kết quả trả về | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| url | String | Đường dẫn tới máy chủ cần request |
| body | String | Nội dung gói tin gửi trong request |

**Method**

Không

**State**

Không

##### Lớp BikeTypeUtils

**Mục đích sử dụng**

Mapping từ mã loại xe (dạng int) sang tên loại xe (dạng String) tương ứng.

**Attribute**

Không

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | translateType | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Mapping từ mã loại xe (dạng int) sang tên loại xe (dạng String) tương ứng. | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| type | int | Mã loại xe:   * 1: Xe đạp đơn * 2: Xe đạp đôi * 3: Xe đạp điện * 4: Xe đạp điện đôi |

**Method**

Không

**State**

Không

##### Lớp BarcodeUtils

**Mục đích sử dụng**

Gọi Barcode API, chuyển mã vạch thành mã xe tương ứng.

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | httpConnector | HttpConnector |  | Đối tượng sử dụng để thực hiện gửi HTTP request đến Barcode API |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | getBikeByBarcode | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Lấy về thông tin xe có mã vạch tương ứng | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| barcode | String | Mã vạch của xe |

**Method**

Không

**State**

Không

##### Lớp FeeCalculator

**Mục đích sử dụng**

Tính toán chi phí thuê xe

**Attribute**

Không

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | calculateFee | ***Kiểu dữ liệu trả về*** | Long |
| ***Mô tả*** | Tính toán chi phí thuê xe | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| renttransactionEntity | RentTransactionEntity | Giao dịch thuê xe tương ứng |

**Method**

Không

**State**

Không

#### Gói requestInterface

##### Lớp RequestModel

**Mục đích sử dụng**

Chứa thông tin nhận được từ front-end

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | cardID | String | NULL | ID định danh thẻ mỗi lần thanh toán |
| 2 | barcode | String | NULL | Mã vạch của xe |
| 3 | parkinglotID | String | NULL | Mã bãi xe |
| 4 | cardCode | String | NULL | Mã thẻ thanh toán |
| 5 | cardOwner | String | NULL | Tên chủ thẻ |
| 6 | cvv | String | NULL | Card Verification Value |
| 7 | expireDate | String | NULL | Ngày hết hạn |

**Operation**

Không

**Method**

Không

**State**

Không

#### Interbank subsystem

##### InterbankInterface

**Mục đích sử dụng**

Định nghĩa các phương thức để kết nối thanh toán

**Attribute**

Không

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | processPayTransaction | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Xử lý giao dịch loại thanh toán | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| transaction | Object | Thông tin thanh toán |
|  | | | | |
| **2** | ***Tên*** | processReturnTransaction | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Xử lý giao dịch loại hoàn tiền | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| transaction | Object | Thông tin thanh toán |
|  | | | | |
| **3** | ***Tên*** | codeToDetail | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Hàm giải mã lỗi tương ứng | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| code | String | Mã lỗi trả về |

**Method**

Không

**State**

Không

##### InterbankSystemController

**Mục đích sử dụng**

Điều khiển các phương thức thanh toán với Interbank

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | transactionPath | String | api/card/processTransaction | Đường dẫn tới API xử lý giao dịch |
| 2 | baseUrl | String | https://ecopark-system-api.herokuapp.com/ | Tên miền của API |
| 3 | secretKey | String |  | Mã bí mật của app với API |
| 4 | interbankBoundary | InterbankBoundary |  | Đối tượng sử dụng để giao tiếp với API |

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | processTransaction | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Phương thức thực hiện giao dịch | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| transaction | InterbankTransaction | Thông tin giao dịch thanh toán |
|  | | | | |
| **2** | ***Tên*** | processReturnTransaction | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Phương thức thực hiện giao dịch hoàn tiền | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| transaction | Object | Thông tin giao dịch hoàn tiền |
|  | | | | |
| **3** | ***Tên*** | processPayTransaction | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Phương thức thực hiện giao dịch thanh toán | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| transaction | Object | Thông tin giao dịch hoàn tiền |
|  | | | | |
| **4** | ***Tên*** | codeToDetail | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Hàm chuyển mã lỗi thành message tương ứng | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| code | String | Mã lỗi |

**Method**

Không

**State**

Không

##### Lớp InterbankBoundary

**Mục đích sử dụng**

Giao tiếp với API Interbank, trả về mã lỗi.

**Attribute**

Không

**Operation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | ***Tên*** | processTransaction | ***Kiểu dữ liệu trả về*** | String |
| ***Mô tả*** | Phương thức truy vấn API | | |
| ***Danh sách tham số*** | *Tên* | *Kiểu dữ liệu* | *Mô tả* |
| url | String | Địa chỉ của API thanh toán |
| body | String | Nội dung của giao dịch thanh toán |

**Method**

Không

**State**

Không

##### Lớp InterbankTransaction

**Mục đích sử dụng**

Chức các thông tin dùng để thực hiện thanh toán với Interbank

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu* | *Giá trị mặc định* | *Mô tả* |
| 1 | amount | String | NULL | Số tiền giao dịch |
| 2 | cardCode | String | NULL | Mã định danh thẻ |
| 3 | cvvCode | String | NULL | Card Verification Value |
| 4 | dateExpired | String | NULL | Ngày hết hạn |
| 5 | owner | String | NULL | Tên chủ thẻ |
| 6 | transactionContent | String | NULL | Nội dung giao dịch |
| 7 | createdAt | String | NULL | Thời điểm tạo |
| 8 | command | String | NULL | Loại giao dịch: thanh toán hoặc hoàn tiền |

**Operation**

Không

**Method**

Không

**State**

Không

# Design Considerations

***<Describe issues which need to be addressed or resolved before attempting to devise a complete design solution>***

## Goals and Guidelines

*<Describe any goals, guidelines, principles, or priorities which dominate or embody the design of the system and its software.*

*Examples of such goals might be: an emphasis on speed versus memory use; or working, looking, or “feeling” like an existing product.*

*Guidelines include coding guidelines and conventions.*

*For each such goal or guideline, describe the reason for its desirability unless it is implicitly obvious.*

*Describe any design policies and/or tactics that do not have sweeping architectural implications (meaning they would not significantly affect the overall organization of the system and its high-level structures), but which nonetheless affect the details of the interface and/or implementation of various aspects of the system (e.g., choice of which specific product to use)*>

## Architectural Strategies

*<Describe any design decisions and/or strategies that affect the overall organization of the system and its higher-level structures. These strategies should provide insight into the key abstractions and mechanisms used in the system architecture. Describe the reasoning employed for each decision and/or strategy (possibly referring to previously stated design goals and principles) and how any design goals or priorities were balanced or traded-off.*

*Examples of design decisions might concern (but are not limited to) things like the following:*

*• Use of a particular type of product (programming language, database, library, commercial off-the-shelf (COTS) product, etc.)*

*• Reuse of existing software components to implement various parts/features of the system*

*• Future plans for extending or enhancing the software*

*• User interface paradigms (or system input and output models)*

*• Hardware and/or software interface paradigms*

*• Error detection and recovery*

*• Memory management policies*

*• External databases and/or data storage management and persistence*

*• Distributed data or control over a network*

*• Generalized approaches to control*

*• Concurrency and synchronization*

*• Communication mechanisms*

*• Management of other resources*

>

## Coupling and Cohesion

*<Evaluate your design and describe which levels of coupling and cohesion that your design is at. Give proofs for your assumptions. Explain if there is any special design or exceptions>*

## Design Principles

*<Does your design follow the SOLID principles for the new requirements/changing requirements? Give proofs for your assumptions. Explain if there is any special design or exceptions>*

## Design Patterns

*<Do you use any design patterns for your design? If yes, describe detailly why you use those design patterns? Describe in detail on the solutions and how to implement each design pattern>*