

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện công nghệ thông tin và truyền thông

Tài liệu mô tả thiết kế phần mềm

Version 1.3

HỆ THỐNG THUÊ XE ECOBIKE

Môn: Thiết kế và xây dựng phần mềm

Nhóm 12

Giáo viên hướng dẫn: TS. Nguyễn Thị Thu Trang

1. Trần Xuân Lộc – 20183577
2. Vũ Quang Huy – 20183930
3. Trần Hữu Hiếu – 20180078

Hà Nội, ngày 7 tháng 1 năm 2022

Table of Contents

Giới thiệu	4
Mục đích	4
Phạm vi	4
Từ điển thuật ngữ	4
Tham khảo	5
Mô tả chung	6
Tổng quan	6
Assumptions/Constraints/Risks	6
Giả định	6
Hạn chế	6
Rủi ro	6
Kiến trúc hệ thống và thiết kế kiến trúc	7
Lựa chọn kiến trúc phần mềm	7
Biểu đồ tương tác	8
Biểu đồ lớp phân tích	13
Biểu đồ lớp phân tích hợp nhất	17
Security Software Architecture	17
Thiết kế chi tiết	18
Thiết kế giao diện người dùng	18
Chuẩn hoá cấu hình màn hình	18
Sơ đồ dịch chuyển màn hình	19
Đặc tả màn hình	19
Mô hình hoá dữ liệu	29
Mô hình khái niệm	29
Thiết kế cơ sở dữ liệu	29
Hệ thống quản lý cơ sở dữ liệu	29
Sơ đồ cơ sở dữ liệu	29
Thiết kế chi tiết cơ sở dữ liệu	31
Non-Database Management System Files	35
Thiết kế lớp	35
Biểu đồ lớp tổng quan	35
Biểu đồ lớp	36
Biểu đồ lớp trong gói Controller	36

Biểu đồ lớp trong gói Service	36
Thiết kế lớp	38
Lớp BikeReturnController	38
Lớp HomeController	39
Lớp BikeController	40
Lớp BikeRentController	41
Lớp DockController	42
Lớp PaymentController	43
Lớp BikeRentService	43
Lớp BikeReturnService	44
Lớp BikeService	45
Lớp DockService	46
Lớp RentalFeeCalculatorInterface	47
Lớp BikeCategoryDao	47
Lớp DockDao	48
Lớp BikeRentDao	49
Lớp BikeReturnDao	50
Lớp DepositTransactionDao	51
Lớp PaymentTransactionDao	52
Lớp Dock	52
Lớp DockHasBike	53
Lớp BikeReturn	54
Lớp BikeCategory	55
Lớp BikeRent	55
Lớp DepositTransaction	56
Lớp PaymentTransaction	57
Lớp BodyToConvertMD5Dto	58
Lớp RequestProcessTransactionDto	59
Lớp TransactionToConvertMD5Dto	60
Lớp RequestTransactionDto	61
Lớp RequestResetBalanceDto	62
Lớp NumberOfBikeCategoryDto	62
InterbankInterface	63
Design Considerations	67
Mục tiêu và định hướng	67
Architectural Strategies	68

Coupling and Cohesion	68
Design Principles	70
Design Patterns	73

1 Giới thiệu

1.1 Mục đích

Mục đích của Tài liệu thiết kế phần mềm này là cung cấp mô tả về cách thiết kế một hệ thống thuê xe điện tử để giúp cho các nhà phát triển, nhà quản lý có được sự hiểu biết cách chi tiết đầy đủ về hệ thống thuê xe sẽ được xây dựng. Tài liệu thiết kế được tạo ra từng bước một và sẽ được cập nhật từng bước trong vòng đời phát triển hệ thống.

1.2 Phạm vi

Hệ thống ra đời cho phép người dùng có thể xem, tìm kiếm thông tin các bãi xe xung quanh khu vực mình đang đứng, thông tin các loại xe mà mình muốn thuê, lựa chọn xe muốn thuê. Sau đó thực hiện các thao tác thuê xe và đặt cọc xe. Sau khi nhận được xe bạn có thể sử dụng và xem tình trạng hiện tại của xe. Khi kết thúc thuê, bạn trả xe và sẽ nhận lại số tiền cọc cũng như thanh toán tiền thuê với hệ thống. Những quy trình này hoàn toàn đơn giản dễ hiểu nhanh chóng thực hiện. Vì vậy phạm vi tác động của hệ thống là tất cả mọi người dùng, bất cứ ai cũng có thể đặt hàng và nhận hàng một cách cực kỳ dễ dàng.

EcobikeRental là một hệ thống đa nền tảng hoạt động 24/7, cho phép người dùng mới có thể làm quen dễ dàng. Hệ thống này có thể phục vụ 100 người dùng cùng lúc mà hiệu suất không thay đổi đáng kể, đồng thời có thể hoạt động 200 giờ liên tục không lỗi. Ngoài ra, hệ thống có thể hoạt động trở lại bình thường trong vòng 2 giờ sau khi xảy ra lỗi. Thời gian đáp ứng tối đa của hệ thống (nếu không được nêu rõ) là 1 giây lúc bình thường hoặc 2 giây lúc cao điểm.

Các loại xe mà hệ thống phục vụ bao gồm các loại xe điện trên thị trường hiện nay: xe đạp đơn thường, xe đạp đôi thường, xe đạp đơn điện ...

1.3 Từ điển thuật ngữ

<i>STT</i>	<i>Thuật ngữ</i>	<i>Giải thích</i>	<i>Ví dụ</i>	<i>Ghi chú</i>
1	Thẻ tín dụng	là loại thẻ cho phép chủ sở hữu được sử dụng trong hạn mức tín dụng tuần hoàn được cấp, và chủ thẻ cần thanh toán mức trả nợ tối thiểu vào ngày đến hạn		
2	Mã vạch	là cách biểu diễn thông tin trực quan trên bề mặt của sản phẩm, hàng hóa mà máy móc có thể đọc được		

1.4 Tham khảo

- [1] TS. Nguyễn Thị Thu Trang (2021). Slide bài giảng thiết kế và xây dựng phần mềm.
- [2] TS. Nguyễn Thị Thu Trang (2021). EcoBikeRental-ProblemStatement-VI.pdf
- [3] Tài liệu hướng dẫn thực hành thiết kế và xây dựng phần mềm – IT4490 (2021).

2 Mô tả chung

2.1 Tổng quan

Phần mềm thuê xe đạp EcoBikeRental được thiết kế như một ứng dụng web theo mô hình MVC. Tuy việc áp dụng ứng dụng web vào một giải pháp cho thuê xe là không thực tế, do ứng dụng yêu cầu người dùng di chuyển di động ngoài trời, và cũng không đảm bảo phần cứng (VD: barcode scanner) để giải quyết đúng nhu cầu người dùng, nhưng với mục đích luyện tập trong môn học, phần mềm hoạt động đúng như mục đích.

2.2 Assumptions/Constraints/Risks

2.2.1 Giả định

Với mục đích demo, phần mềm EcoBikeRental giả định rằng người dùng có thể nhập barcode dưới dạng chuỗi số vào trong ô nhập barcode để phục vụ mục đích thuê xe.

2.2.2 Hạn chế

Hạn chế hàng đầu của phần mềm EcoBikeRental là việc thiếu đi thiết bị barcode scanner cho việc quét barcode để thuê xe. Để bù đắp cho sự thiếu hụt này, người dùng sẽ phải nhập tay barcode vào một ô textbox.

2.2.3 Rủi ro

Với mục đích là demo, phần mềm EcoBikeRental không cài đặt các chức năng liên quan đến bảo mật và xác thực người dùng. Vì vậy, trong suốt thời gian thực thi của phần mềm, phần mềm giả định chỉ có một người dùng duy nhất. Việc này có thể gây ra một vài giả định sai lầm cho đội ngũ phát triển, mà không thể phát hiện ngay cả khi chương trình demo đã chạy một cách hoàn chỉnh.

3 Kiến trúc hệ thống và thiết kế kiến trúc

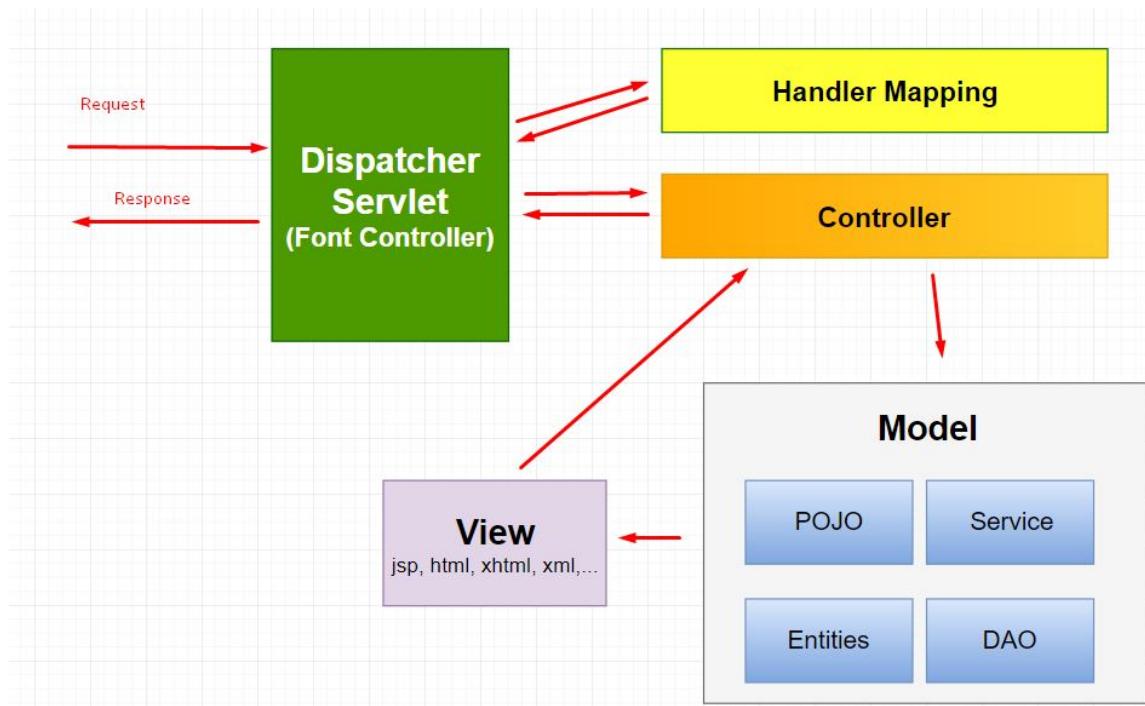
3.1 Lựa chọn kiến trúc phần mềm

Dựa trên các thiết kế đã được dựng và yêu cầu nghiệp vụ của phần mềm, nhóm quyết định sử dụng ngôn ngữ lập trình Java với framework Java Spring MVC và giao diện jsp để xây dựng mã nguồn của phần mềm.

Spring MVC là một framework MVC sử dụng một trình điều phối để gọi các phương thức trong controller và chuyển tiếp điều khiển đến một view (khung nhìn). Đây là lợi ích đầu tiên của việc sử dụng Spring MVC: Bạn không cần viết servlet điều phối của riêng mình. Dưới đây là danh sách các tính năng mà Spring MVC được trang bị để phát triển nhanh hơn:

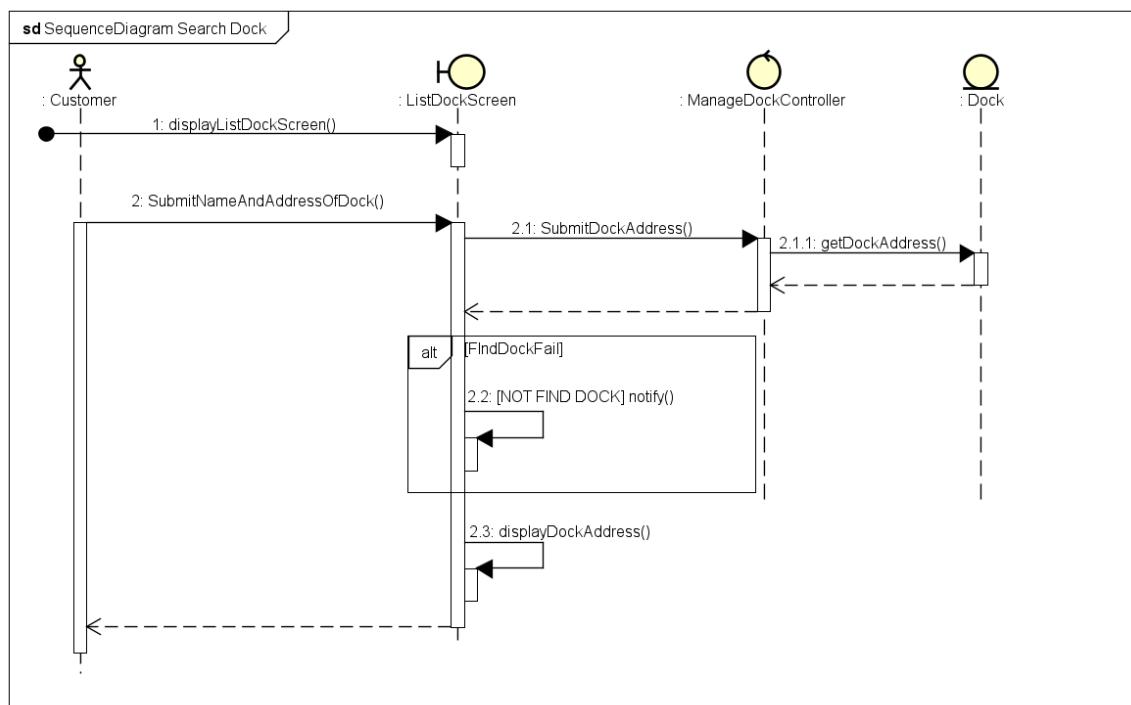
Spring MVC cung cấp một trình điều phối, lưu bản ghi của bạn.

- Spring MVC sử dụng tệp cấu hình bằng XML mà bạn có thể chỉnh sửa mà không cần biên dịch lại ứng dụng.
- Spring MVC khởi tạo các lớp controller và gắn các bean với các đầu vào của người dùng.
- Spring MVC tự động liên kết đầu vào của người dùng với đúng kiểu dữ liệu. Ví dụ, Spring MVC có thể tự động phân tích cú pháp một chuỗi và đặt thuộc tính kiểu float hoặc thập phân.
- Spring MVC xác thực đầu vào của người dùng và chuyển hướng người dùng trở lại form đầu vào nếu xác thực không thành công. Xác thực đầu vào là tùy chọn và có thể được thực hiện theo chương trình hoặc theo khai báo. Trước đó, Spring MVC cung cấp trình xác thực tích hợp cho hầu hết các tác vụ bạn có thể gặp phải khi xây dựng một ứng dụng web.
- Spring MVC là một phần của Spring framework. Bạn nhận được mọi thứ mà Spring cung cấp.
- Spring MVC hỗ trợ quốc tế hóa (internationalization) và bản địa hóa (localization). Điều này có nghĩa là bạn có thể hiển thị tin nhắn bằng nhiều ngôn ngữ tùy thuộc vào ngôn ngữ người dùng.
- Spring MVC hỗ trợ nhiều công nghệ hiển thị (view). Hầu hết trường hợp bạn sẽ sử dụng JSP, nhưng các công nghệ khác, bao gồm Velocity và FreeMarker, cũng được hỗ trợ.

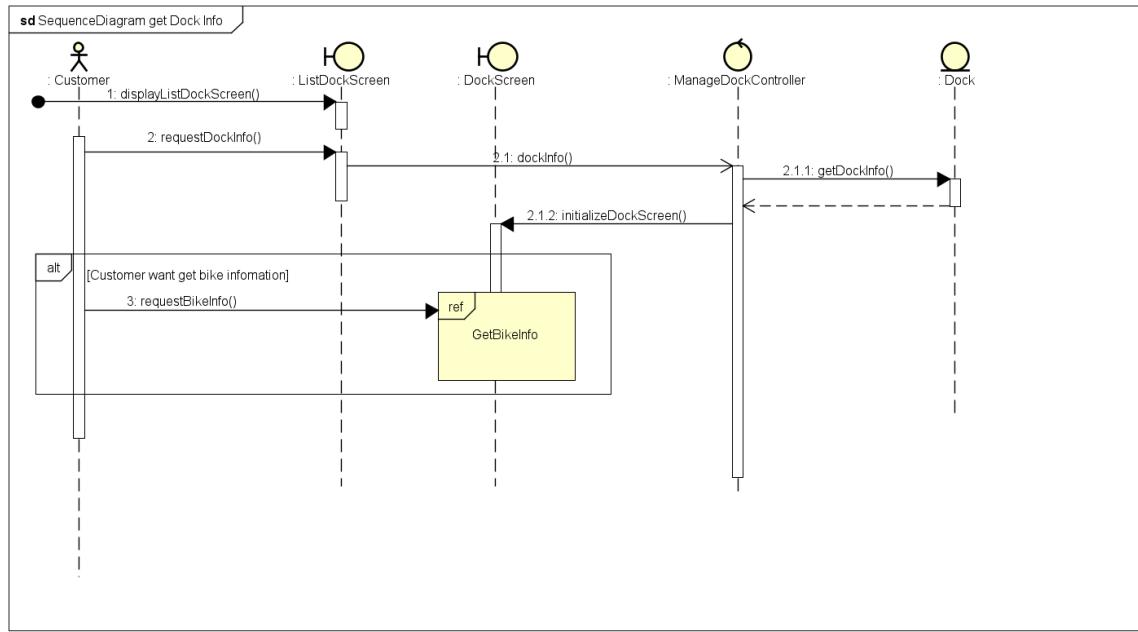


3.2 Biểu đồ tương tác

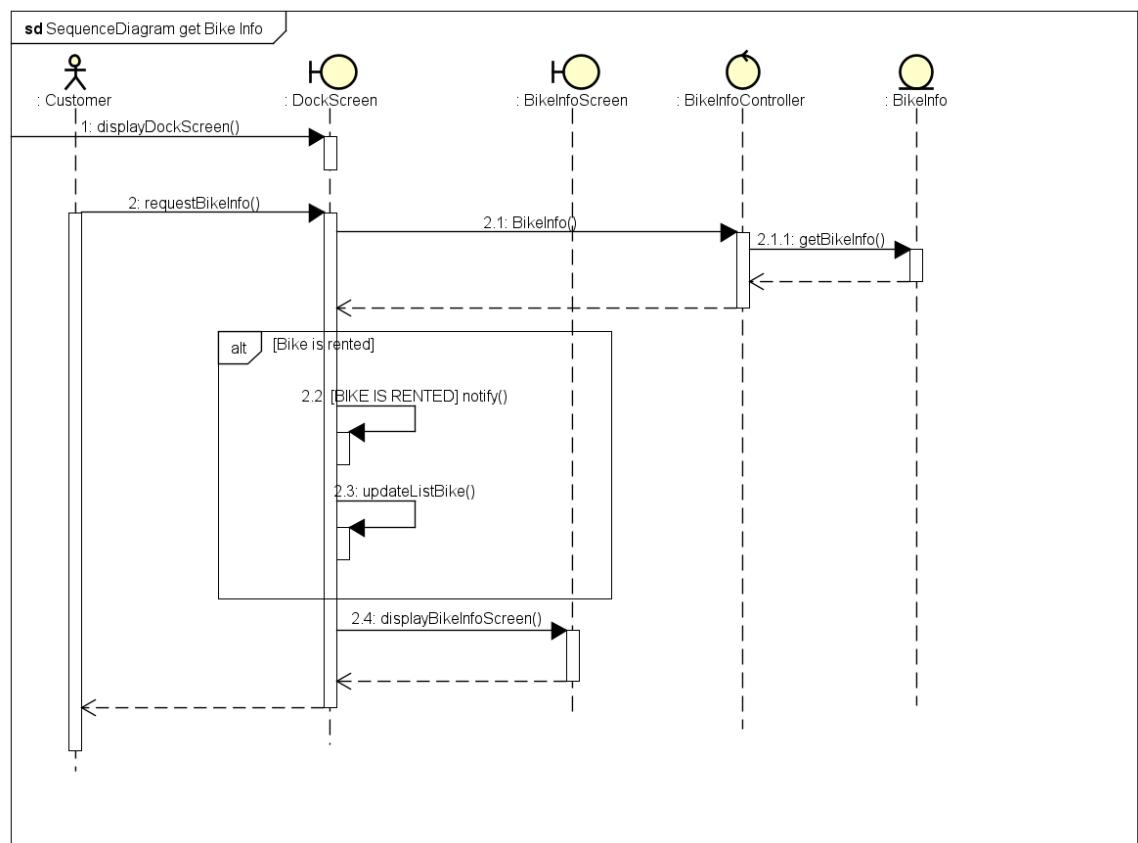
3.2.1 Biểu đồ “Tìm kiếm bãi xe”



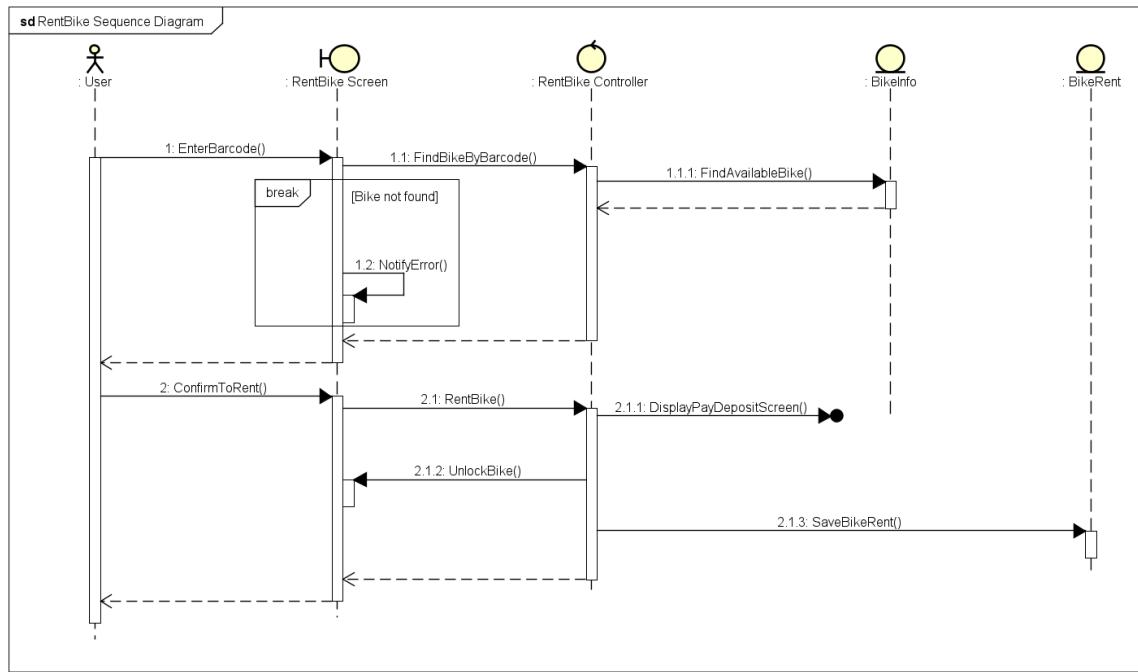
3.2.2 Biểu đồ “Lấy thông tin bãi xe”



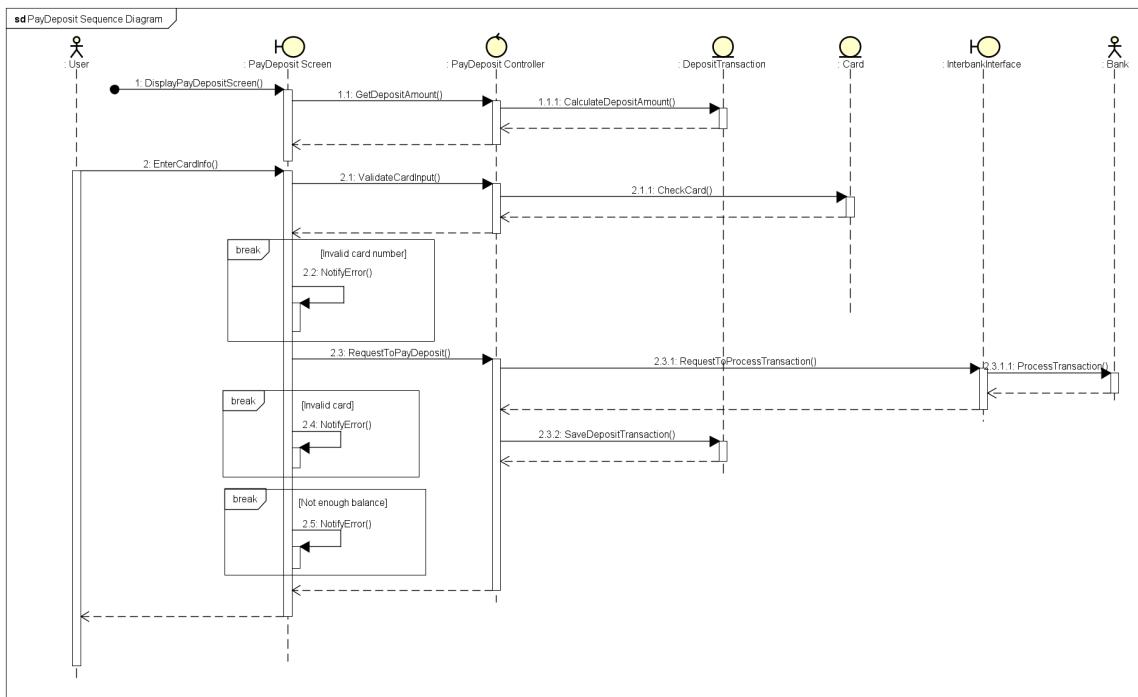
3.2.3 Biểu đồ “Lấy thông tin xe trong bãi”



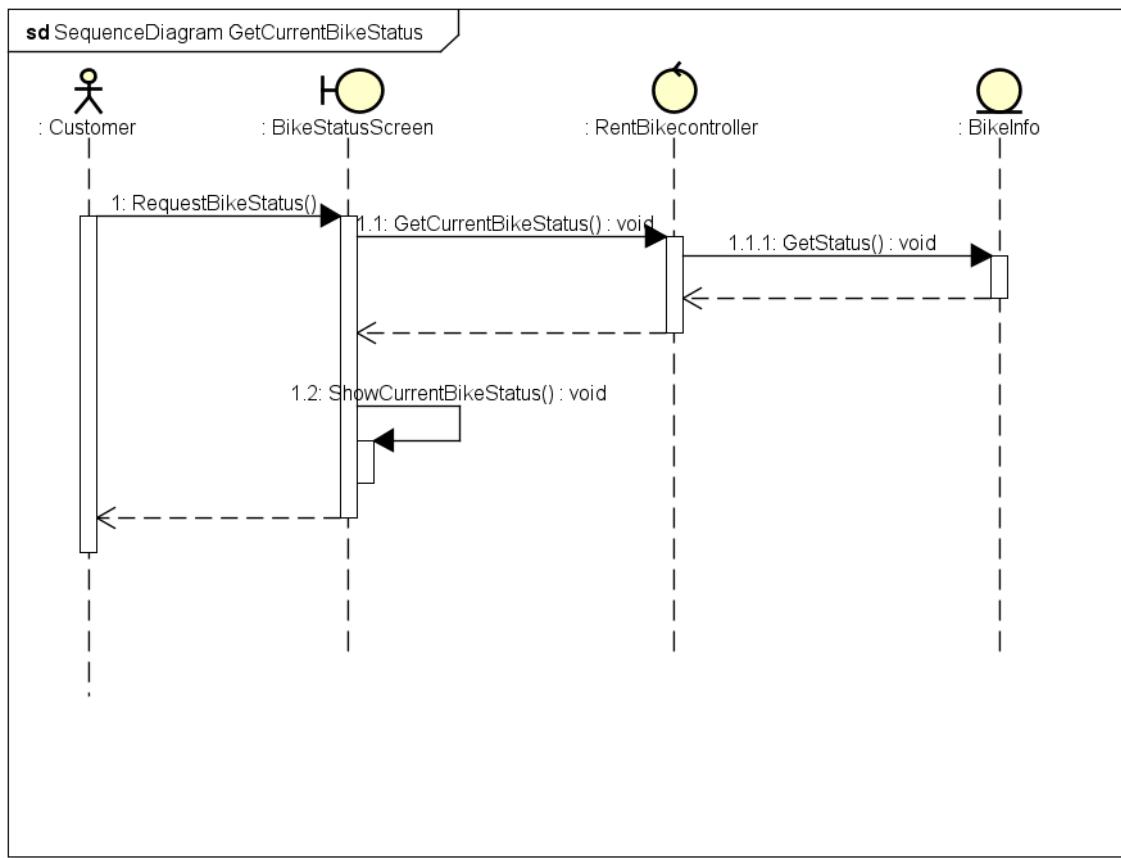
3.2.4 Biểu đồ “Thuê xe”



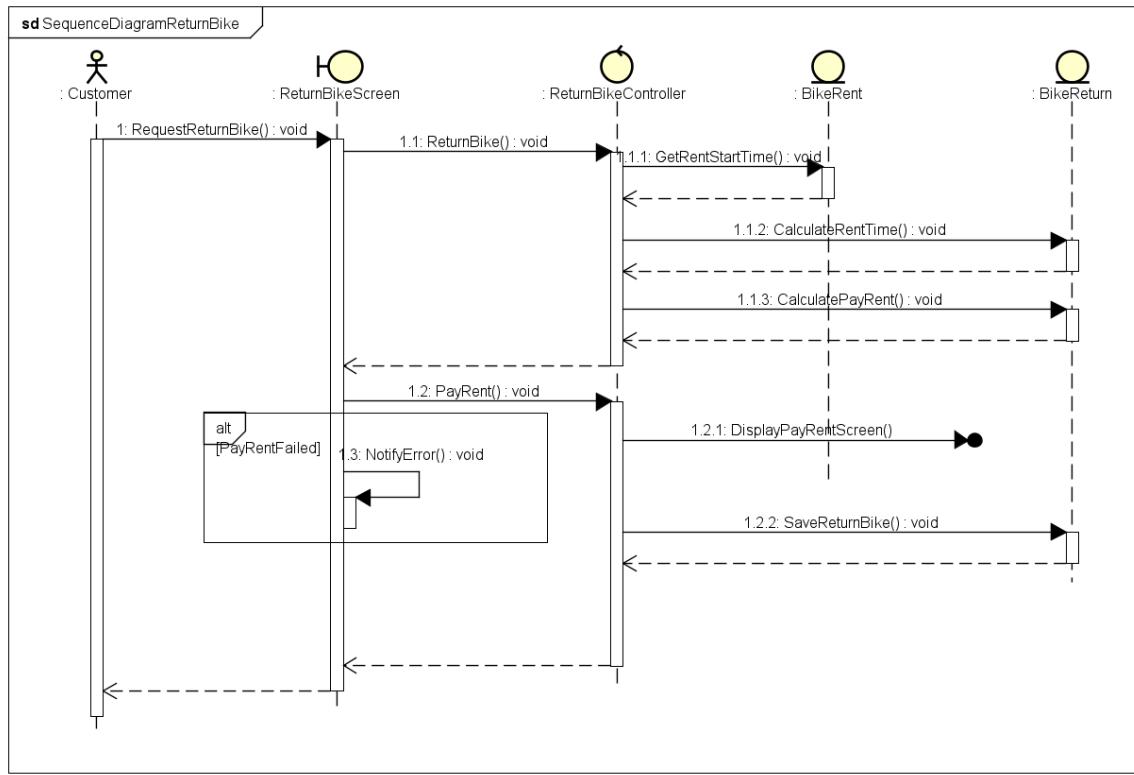
3.2.5 Biểu đồ “Trả tiền đặt cọc”



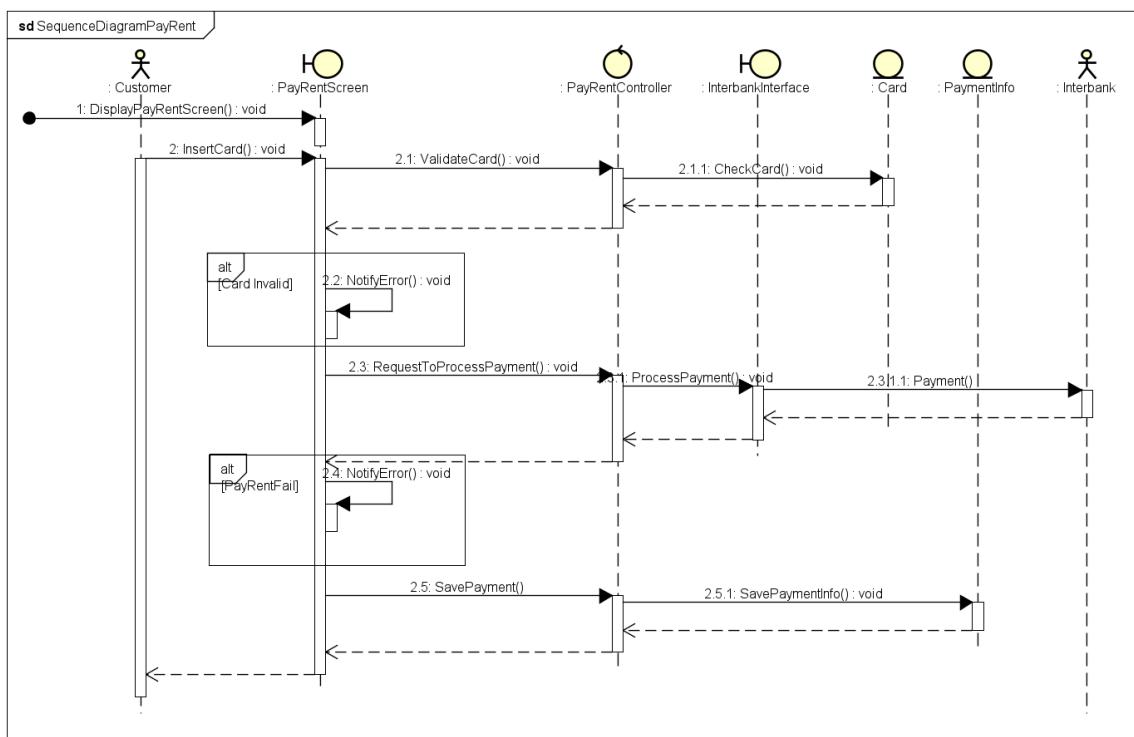
3.2.6 Biểu đồ “Tình trạng xe hiện đang thuê”



3.2.7 Biểu đồ “Trả xe”

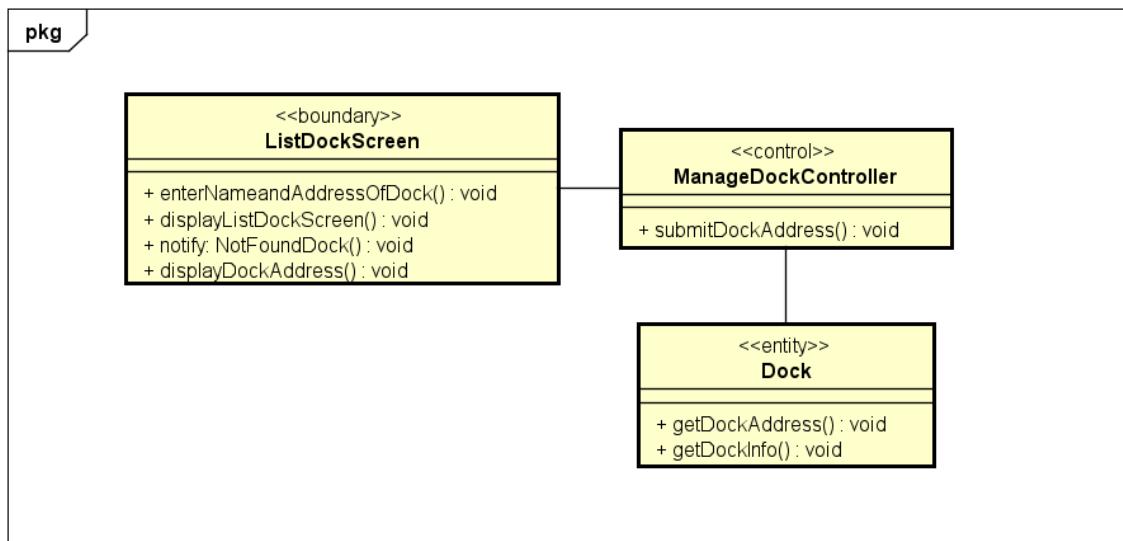


3.2.8 Biểu đồ “Trả tiền thuê xe”

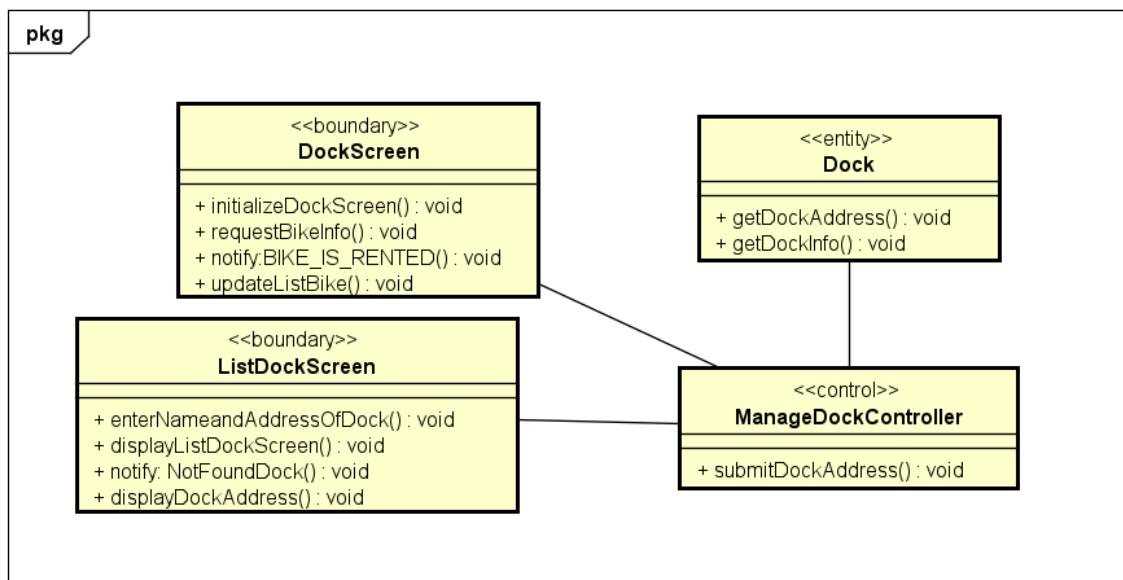


3.3 Biểu đồ lớp phân tích

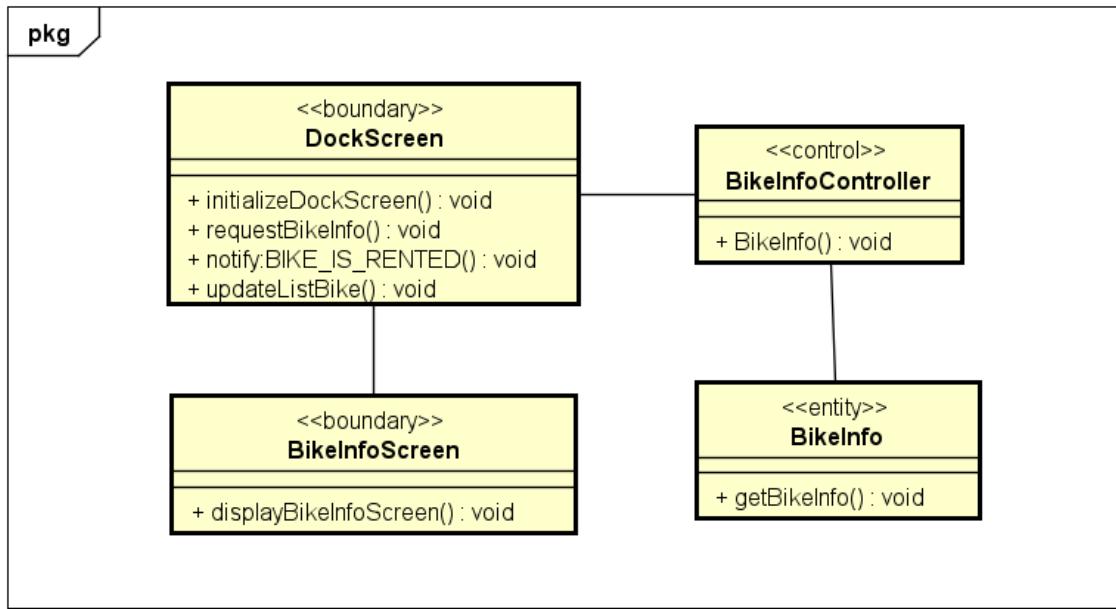
3.3.1 Biểu đồ “Tìm kiếm bãi xe”



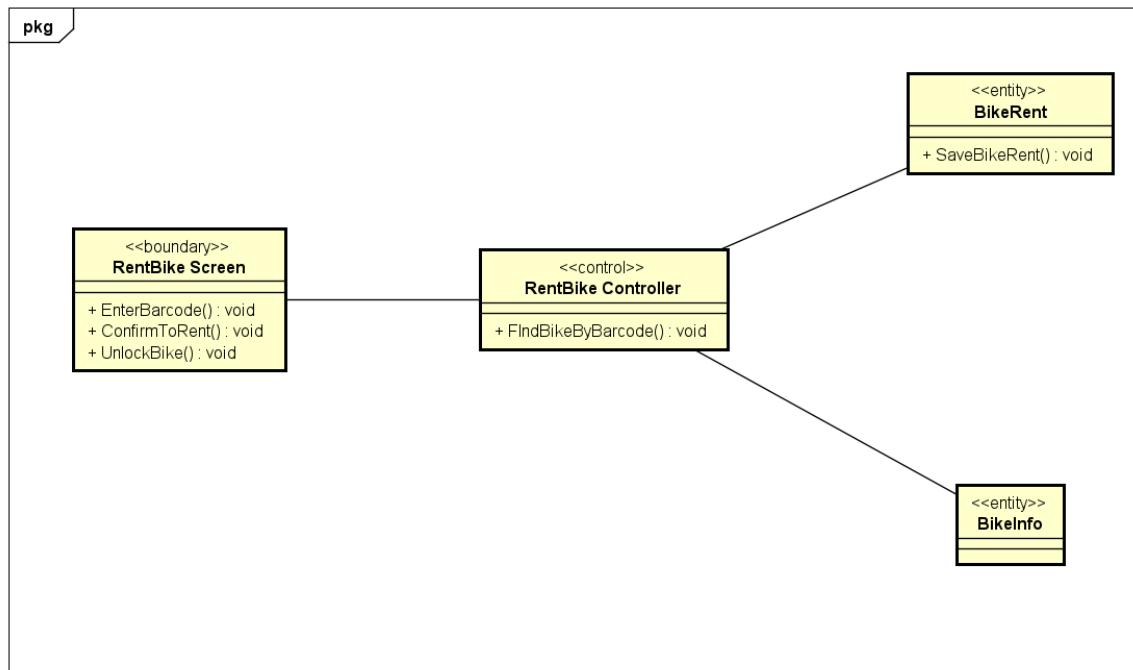
3.3.2 Biểu đồ “Lấy thông tin bãi xe”



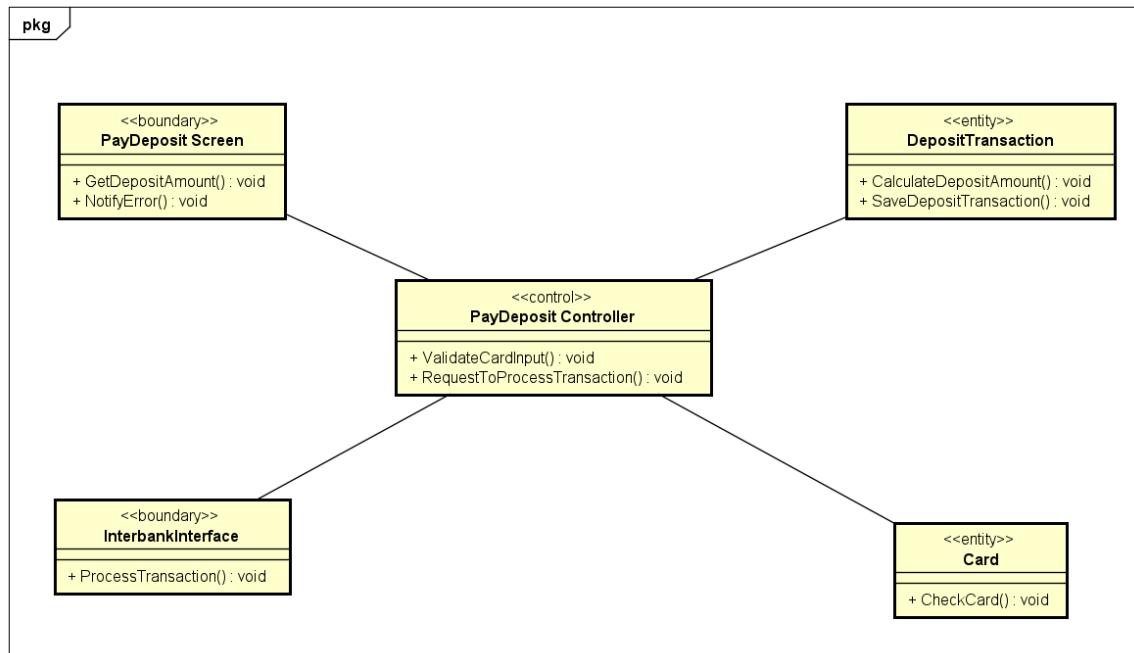
3.3.3 Biểu đồ “Lấy thông tin xe trong bãi”



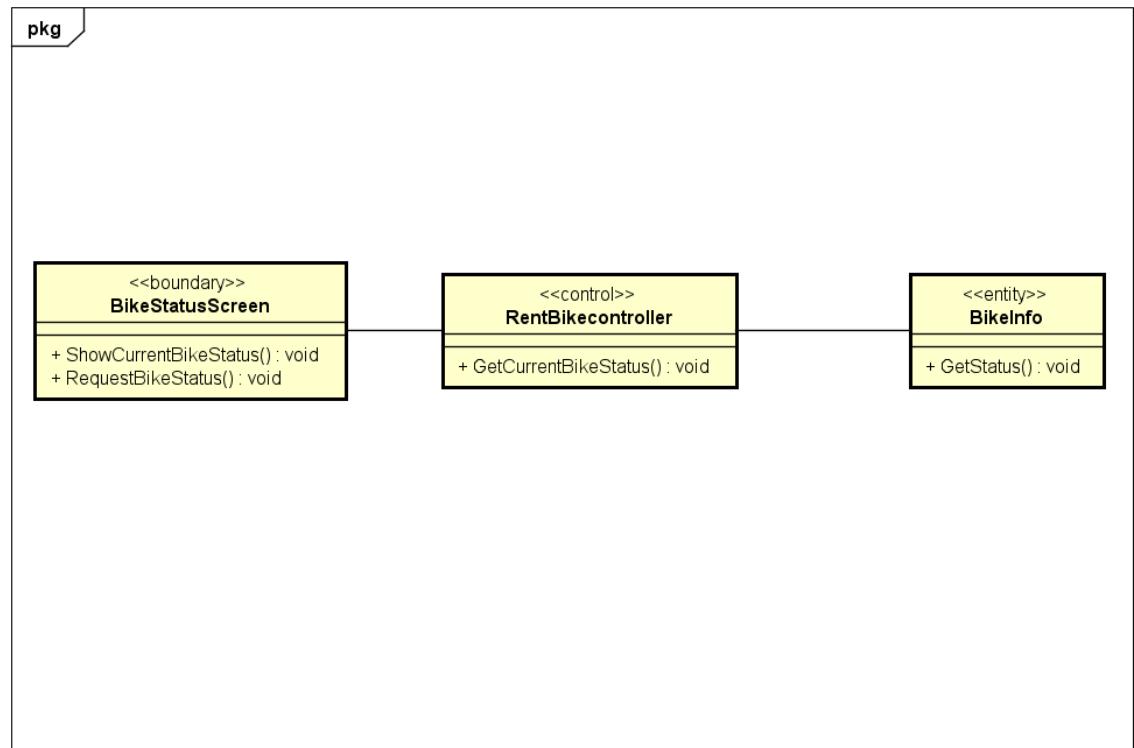
3.3.4 Biểu đồ “Thuê xe”



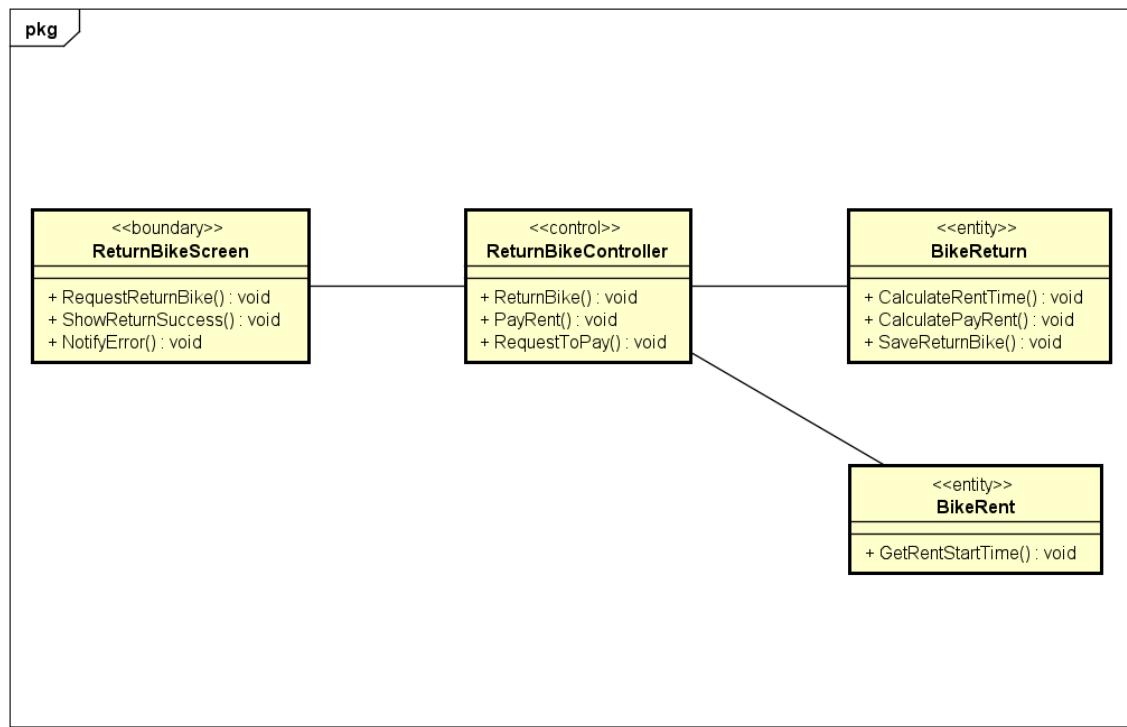
3.3.5 Biểu đồ “Trả tiền đặt cọc”



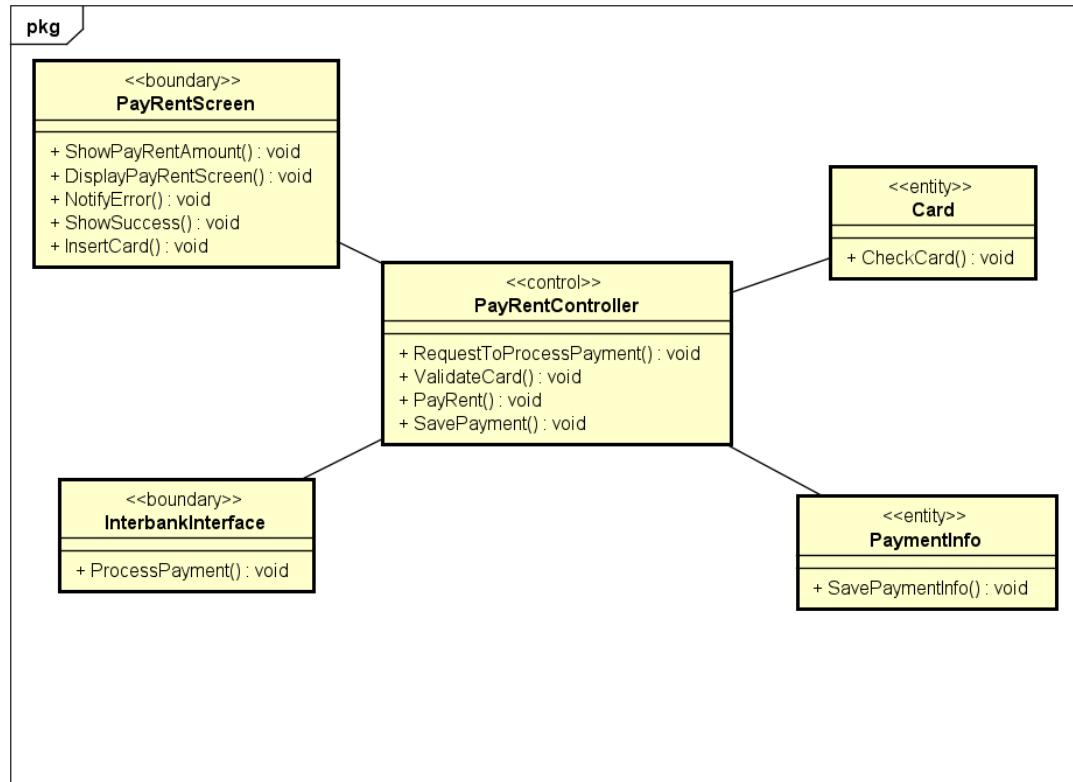
3.3.6 Biểu đồ “Tình trạng xe hiện đang thuê”



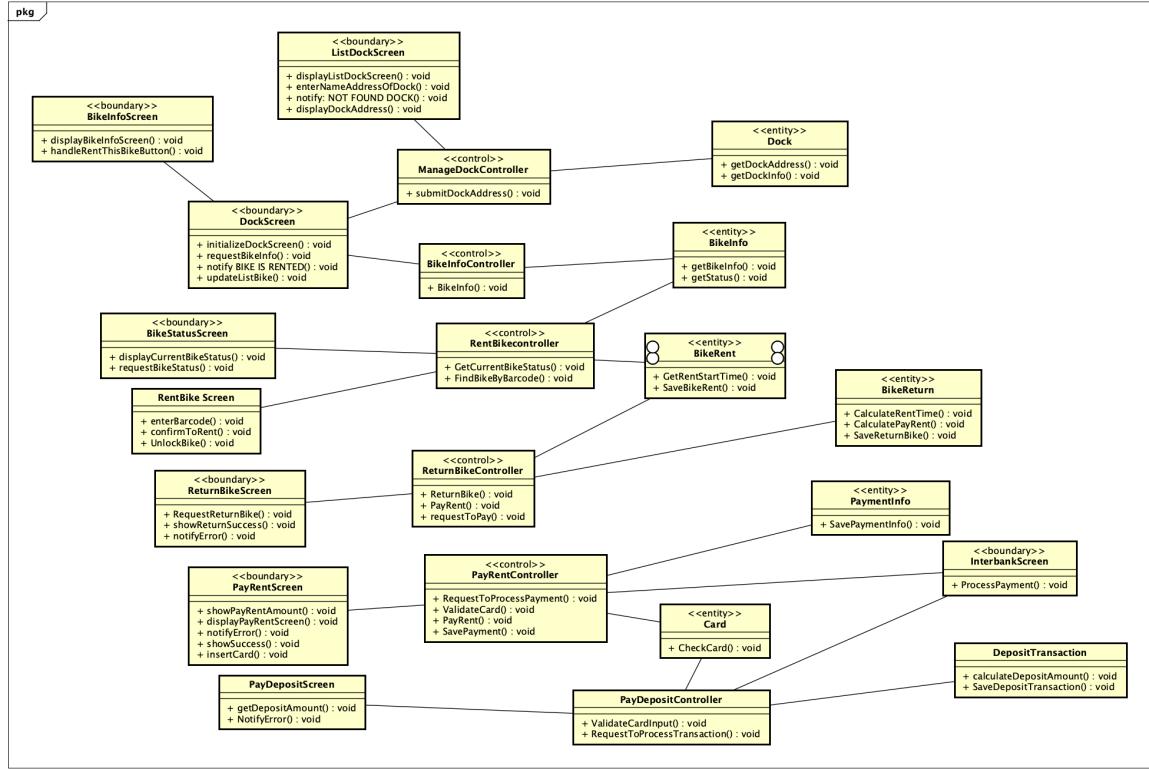
3.3.7 Biểu đồ “Trả xe”



3.3.8 Biểu đồ “Trả tiền thuê xe”



3.4 Biểu đồ lớp phân tích hợp nhất



3.5 Security Software Architecture

Hệ thống không cài đặt các chức năng liên quan đến xác thực người dùng.

4 Thiết kế chi tiết

4.1 Thiết kế giao diện người dùng

4.1.1 Chuẩn hóa cấu hình màn hình

Display

- Số lượng màu được hỗ trợ: 16,777,216 màu
- Độ phân giải: 1300 x 800 pixels

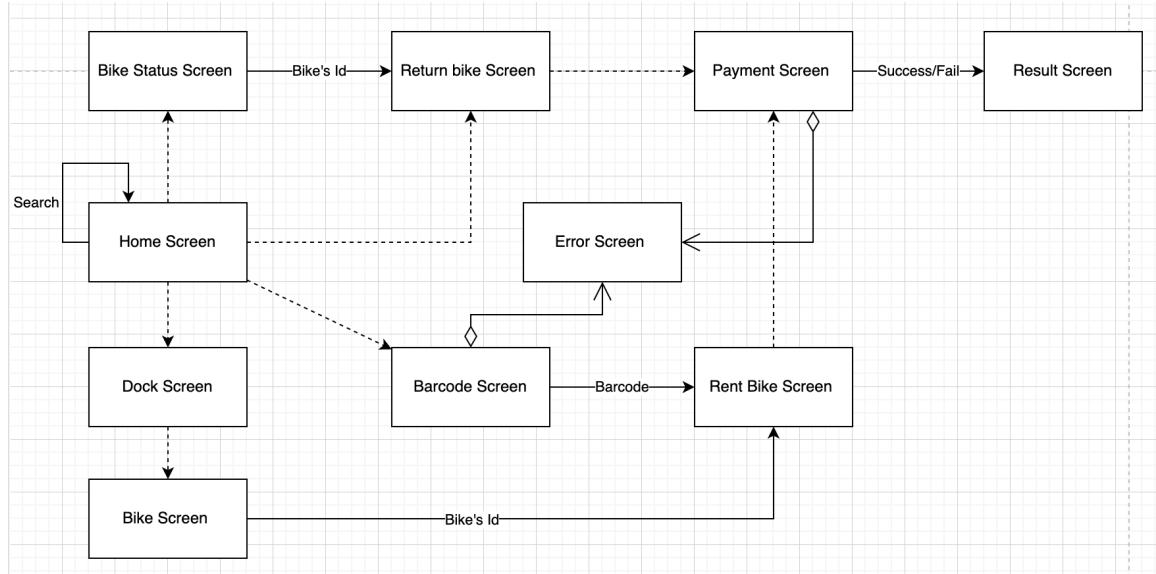
Screen

- Vị trí của button: Ở dưới cùng (theo chiều dọc) và ở giữa (theo chiều ngang) của khung.
- Vị trí của message: Ở giữa trung tâm khung màn hình
- Vị trí của screen title: Title đặt ở góc trên bên trái của màn hình.
- Sự nhất quán trong hiển thị chữ số: dấu phẩy để phân cách hàng nghìn và chuỗi chỉ bao gồm các ký tự, chữ số, dấu phẩy, dấu chấm, dấu cách, dấu gạch dưới và ký hiệu gạch nối.

Control

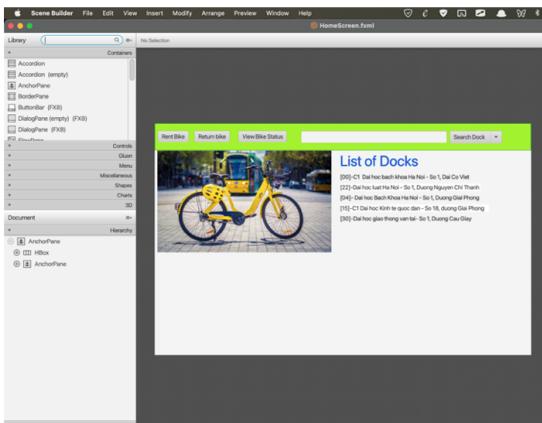
- Kích thước text: medium size (20px).
- Font: Segoe UI. Color: #000000
- Xử lý check input: Nên kiểm tra xem input có empty hay không. Tiếp theo, kiểm tra xem input có đúng format hay không.
- Dịch chuyển màn hình: Không có các khung chồng lên nhau. Các màn hình được tách biệt.
- Thứ tự các màn hình trong hệ thống:
 - Home Screen
 - Dock Screen
 - Bike Status Screen - Xem trạng thái xe hiện tại đang thuê
 - Barcode Screen - Màn hình nhập Barcode
 - Bike Screen
 - Rent Bike Screen
 - Return Bike Screen
 - Payment Screen
 - Result Screen
 - Error Screen

4.1.2 Sơ đồ dịch chuyển màn hình



4.1.3 Đặc tả màn hình

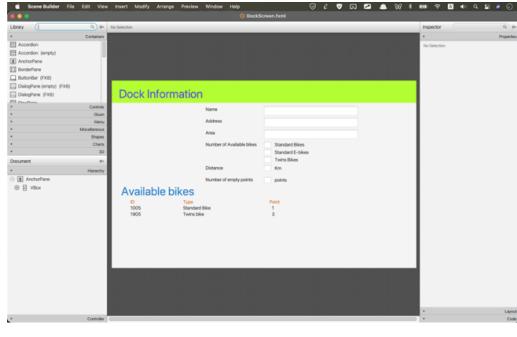
4.1.3.1 Home Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Home Screen	14/11/2021			Trần Hữu Hiếu
		Control	Operation	Function	
Khu vực hiển thị danh sách bãi xe	Khởi tạo	Hiển thị danh sách bãi xe			
Khu vực hiển thị danh sách bãi xe	Double clicks	Người dùng xem chi tiết một bãi xe			
Nút "Thuê xe"	Click	Người dùng muốn thuê xe			
Nút "Trả xe"	Click	Người dùng muốn trả xe			
Nút "xem thông tin xe đang thuê"	Click	Người dùng muốn xem thông tin xe đang thuê			
Khu vực để điền thông tin bãi cần tìm	Gõ từ bàn phím	Người dùng muốn tìm bãi xe người dùng muốn			
Nút tìm bãi	Click	Người dùng tìm bãi xe			

Định nghĩa trường thuộc tính:

Screen Name	HomeScreen	Type	Field Attribute	Remarks
Attribute	Number of digits (bytes)	Type	Field Attribute	Remarks
Số thứ tự bãi xe	1	Numeral	Đen	Căn phải
Tên bãi xe	30	String	Đen	Căn phải
Địa chỉ bãi xe	30	String	Đen	Căn phải

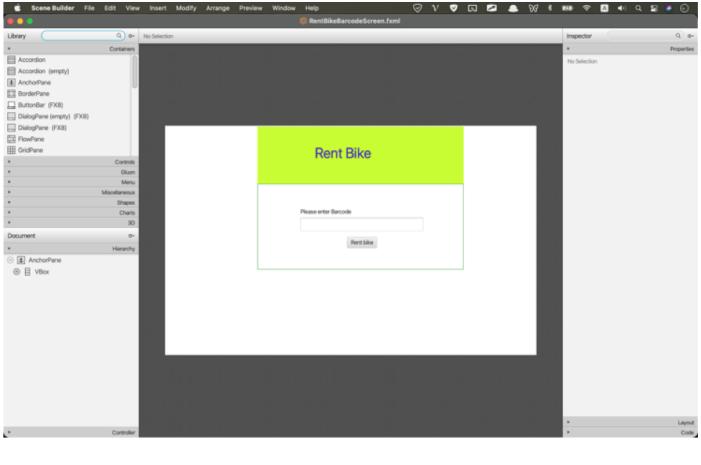
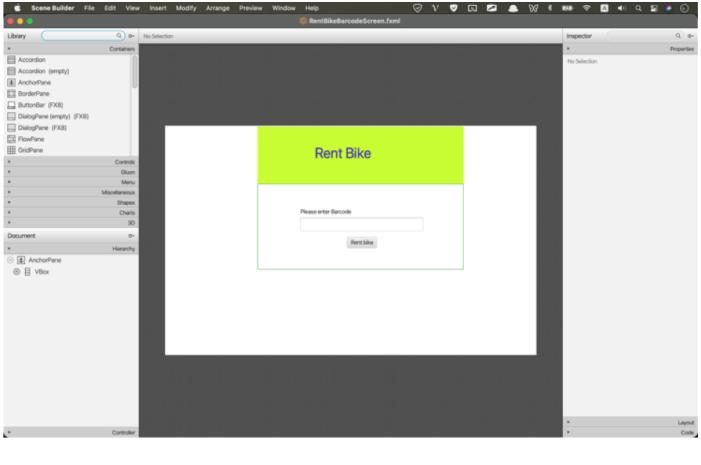
4.1.3.2 Dock Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	DockScreen	14/11/2021			Trần Hữu Hiếu
		Control	Operation	Function	
<p>Khu vực hiển thị thông tin chi tiết bãi xe bao gồm mã xe, tên bãi, địa chỉ, khu vực, số điểm đỗ xe, số xe trong bãi.</p>		Khởi tạo		Hiển thị thông tin chi tiết bãi xe bao gồm mã xe, tên bãi, địa chỉ, khu vực, số điểm đỗ xe, số xe trong bãi.	
<p>Khu vực hiển thị danh sách các xe hiện có trong bãi. Bao gồm id, loại xe, điểm đỗ.</p>		Khởi tạo		Hiển thị danh sách các xe trong bãi	
<p>Khu vực hiển thị danh sách các xe trong bãi</p>		Double Clicks		Xem thông tin chi tiết xe	

Định nghĩa trường thuộc tính:

Screen Name	DockScreen			
Attribute	Number of digits (bytes)	Type	Field Attribute	Remarks
Mã xe	20	String	Đen	Căn trái
Tên bãi	20	String	Đen	Căn trái
Địa chỉ	30	String	Đen	Căn trái
Khu vực	20	String	Đen	Căn trái
Số điểm đỗ xe còn trống	10	Numeral	Đen	Căn trái
Số xe trong bãi	15	Numeral	Đen	Căn trái

4.1.3.3 BarcodeScanner Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Barcode for renting bike Screen	14/11/2021			Trần Hữu Hiếu
		Control	Operation	Function	
	Khu vực nhập barcode	Nhập từ bàn phím	Người dùng nhập từ bàn phím barcode mà người dùng muốn thuê		
	Nút thuê xe	Click	Người dùng xác nhận mong muốn thuê xe với barcode như trên để hệ thống kiểm tra và xác nhận lại		

Định nghĩa trường thuộc tính:

Screen Name	Barcode			
Attribute	Number of digits (bytes)	Type	Field Attribute	Remarks
Barcode	10	Numeral	Đen	Căn trái

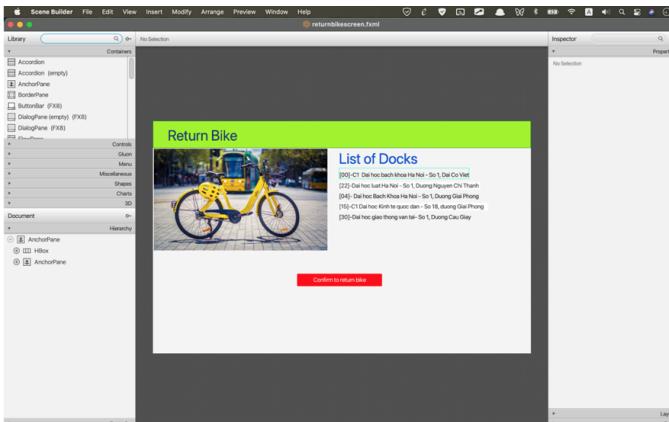
4.1.3.4 Rent bike Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge																					
Screen specification	Bike Rent Screen	14/11/2021			Trần Hữu Hiếu																					
<table border="1"> <thead> <tr> <th>Control</th><th>Operation</th><th>Function</th></tr> </thead> <tbody> <tr> <td>Khu vực hiển thi thông tin chi tiết về xe mà người dùng đã nhập barcode vào</td><td>Khởi tạo</td><td>Hiển thị thông tin chi tiết về xe mà người dùng đã nhập barcode vào</td></tr> <tr> <td>Nút xác nhận</td><td>Click</td><td>Người dùng xác nhận việc thuê xe</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> </tbody> </table>						Control	Operation	Function	Khu vực hiển thi thông tin chi tiết về xe mà người dùng đã nhập barcode vào	Khởi tạo	Hiển thị thông tin chi tiết về xe mà người dùng đã nhập barcode vào	Nút xác nhận	Click	Người dùng xác nhận việc thuê xe												
Control	Operation	Function																								
Khu vực hiển thi thông tin chi tiết về xe mà người dùng đã nhập barcode vào	Khởi tạo	Hiển thị thông tin chi tiết về xe mà người dùng đã nhập barcode vào																								
Nút xác nhận	Click	Người dùng xác nhận việc thuê xe																								

Định nghĩa trường thuộc tính:

Screen Name	Bike Renting			
Attribute	Number of digits (bytes)	Type	Field Attribute	Remarks
ID	10	Numeral	Đen	Căn trái
Barcode	20	String	Đen	Căn trái
Số lượng bàn đạp	10	Numeral	Đen	Căn trái
Số lượng yên xe	10	Numeral	Đen	Căn trái
Số lượng ghế sau	10	Numeral	Đen	Căn trái
Bãi xe hiện tại	30	String	Đen	Căn trái
Biển số xe	10	String	Đen	Căn trái

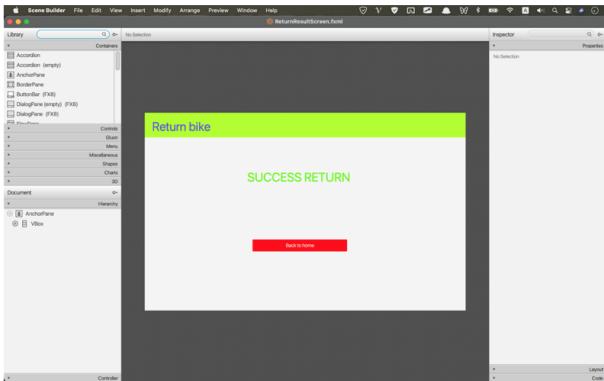
4.1.3.5 Return bike Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Return Bike Screen	14/11/2021			Trần Hữu Hiếu
		Control	Operation	Function	
Khu vực hiển thị danh sách bãi xe	Khởi tạo	Hiện thị danh sách bãi xe gần khách hàng			
Khu vực hiển thị danh sách bãi xe	Click	Chọn bãi xe mà khách hàng muốn trả xe			
Nút xác nhận trả xe	Click	Khách hàng xác nhận trả xe			

Định nghĩa trường thuộc tính:

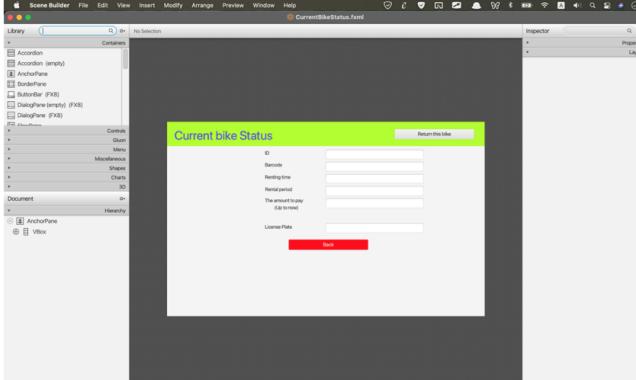
Screen Name	Return bike			
Attribute	Number of digits (bytes)	Type	Field Attribute	Remarks
Id bãi xe còn trống	30	String	Đen	Căn trái
Tên bãi xe	30	String	Đen	Căn trái
Địa chỉ bãi xe	30	String	Đen	Căn trái

4.1.3.6 Return Result Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Home Screen	14/11/2021			Trần Hữu Hiếu
		Control	Operation	Function	
Nút trở về homeScreen	Click	Trở về trang chủ			
Khu vực thông báo trả xe thành công	Khởi tạo	Hiển thị ra màn hình thông báo xe đã được trả			

Định nghĩa trường thuộc tính:

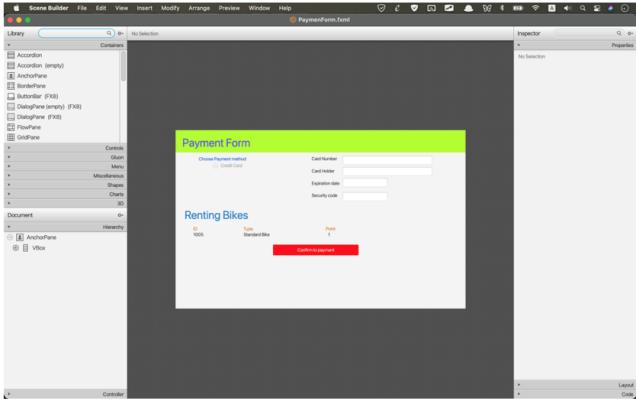
4.1.3.7 Bike Status Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Current Bike Status Screen	14/11/2021			Trần Hữu Hiếu
		Control	Operation	Function	
Khu vực hiển thị thông tin xe	Khởi tạo	Hiển thi thông tin xe như Id, barcode, thời gian thuê, số tiền cần phải trả cho đến bây giờ và biến số xe.			
Nút trở lại màn hình trước đó	Click	Quay trở về màn hình trước đó của khách hàng			
Nút trả xe	Click	Đi đến màn hình trả xe			

Định nghĩa trường thuộc tính:

Screen Name	Current Bike Status	Type	Field Attribute	Remarks
Attribute	Number of digits (bytes)	Type	Field Attribute	Remarks
Id	10	Numeral	Đen	Căn trái
Barcode	10	Numeral	Đen	Căn trái
Renting time	10	Numeral	Đen	Căn trái
Renting period	30	String	Đen	Căn trái
The amount to pay	10	Numeral	Đen	Căn trái
License Plate	20	String	Đen	Căn trái

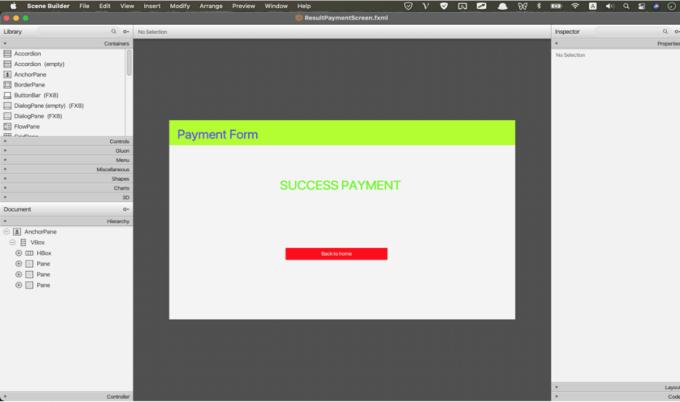
4.1.3.8 Payment Form Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	PaymenForm Screen	14/11/2021			Trần Hữu Hiếu
					
Control	Operation	Function			
Khu vực thông tin thẻ mà khách hàng dùng	Nhập thông tin từ bàn phím	Nhập thông tin cần thiết như số thẻ, chủ thẻ, mã bảo mật, ngày hết hạn			
Khu vực hiển thị xe đang thuê	Khởi tạo	Hiển thị xe khách hàng đang thuê			

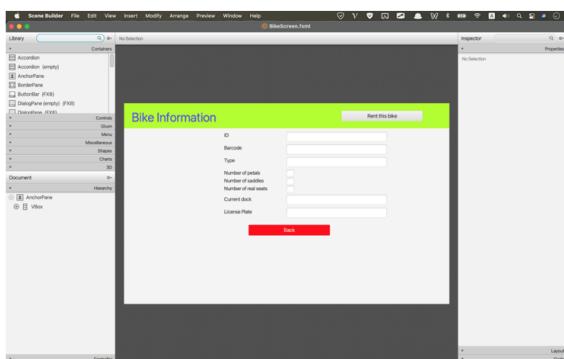
Định nghĩa trường thuộc tính:

Screen Name	Payment Form	Type	Field Attribute	Remarks
Attribute	Number of digits (bytes)	Type	Field Attribute	Remarks
SỐ thẻ	20	String	Đỏ	Căn trái
Tên thẻ	20	String	Đỏ	Căn trái
Ngày hết hạn	20	String	Đỏ	Căn trái
Mã bảo mật	20	String	Đỏ	Căn trái
Id	10	Numeral	Đỏ	Căn trái
Loại xe	20	String	Đỏ	Căn trái
Vị trí	10	Numeral	Đỏ	Căn trái

4.1.3.9 Payment Result Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Payment Result Screen	14/11/2021			Trần Hữu Hiếu
		Control	Operation	Function	
Khu vực hiện thị thanh toán thành công	Khởi tạo	Hiển thị thanh toán thành công			
Nút "Back to home"	Click	Trở về trang chủ			

4.1.3.10 Bike Info Screen

AIMS SOFTWARE		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Bike Infor Screen	14/11/2021			Trần Hữu Hiếu
		Control	Operation	Function	
		Nút thuê xe	Click	Di chuyển đến trang thuê xe	
		Nút trở lại màn hình trước đó	Click	Di chuyển đến màn hình trước đó	
		Khu vực hiển thị chi tiết thông tin xe	Khởi tạo	Hiển thị thông tin chi tiết xe như Id, barcode, số lượng bàn đạp, yên xe, ghế sau, biển số xe hiện tại, bãi xe đang chứa xe này.	

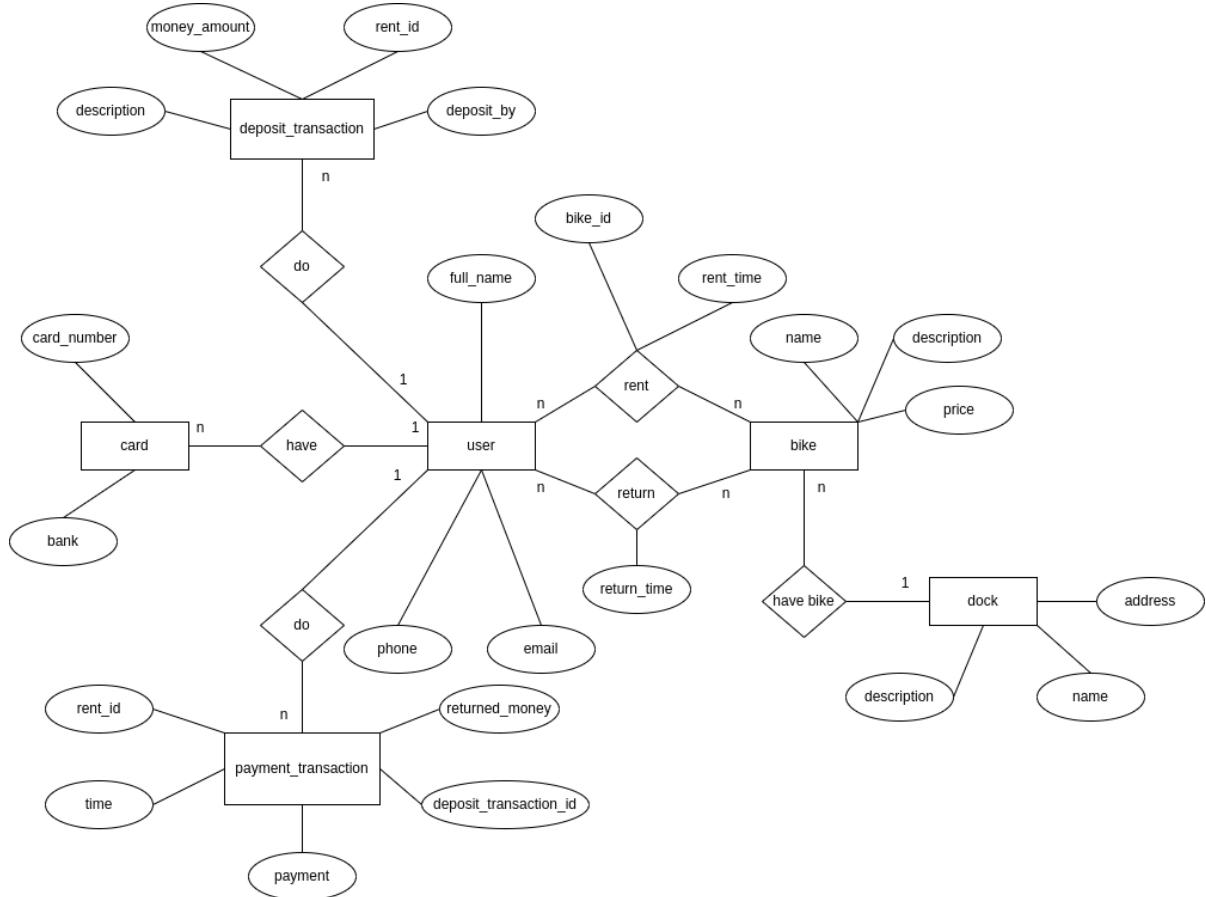
Định nghĩa trường thuộc tính:

Screen Name	BikeInforScreen			
Attribute	Number of digits (bytes)	Type	Field Attribute	Remarks
ID	10	Numeral	Đen	Căn trái
Barcode	20	String	Đen	Căn trái
Số lượng bàn đạp	10	Numeral	Đen	Căn trái
Số lượng yên xe	10	Numeral	Đen	Căn trái
Số lượng ghế sau	10	Numeral	Đen	Căn trái
Bãi xe hiện tại	30	String	Đen	Căn trái
Biển số xe	10	String	Đen	Căn trái

4.2 Mô hình hóa dữ liệu

4.2.1 Mô hình khái niệm

Dưới đây là mô hình thực thể - liên kết biểu diễn thông tin hệ thống cần lưu trữ



4.2.2 Thiết kế cơ sở dữ liệu

4.2.2.1 Hệ thống quản lý cơ sở dữ liệu

Hệ thống quản lý cơ sở dữ liệu được sử dụng là MySQL.

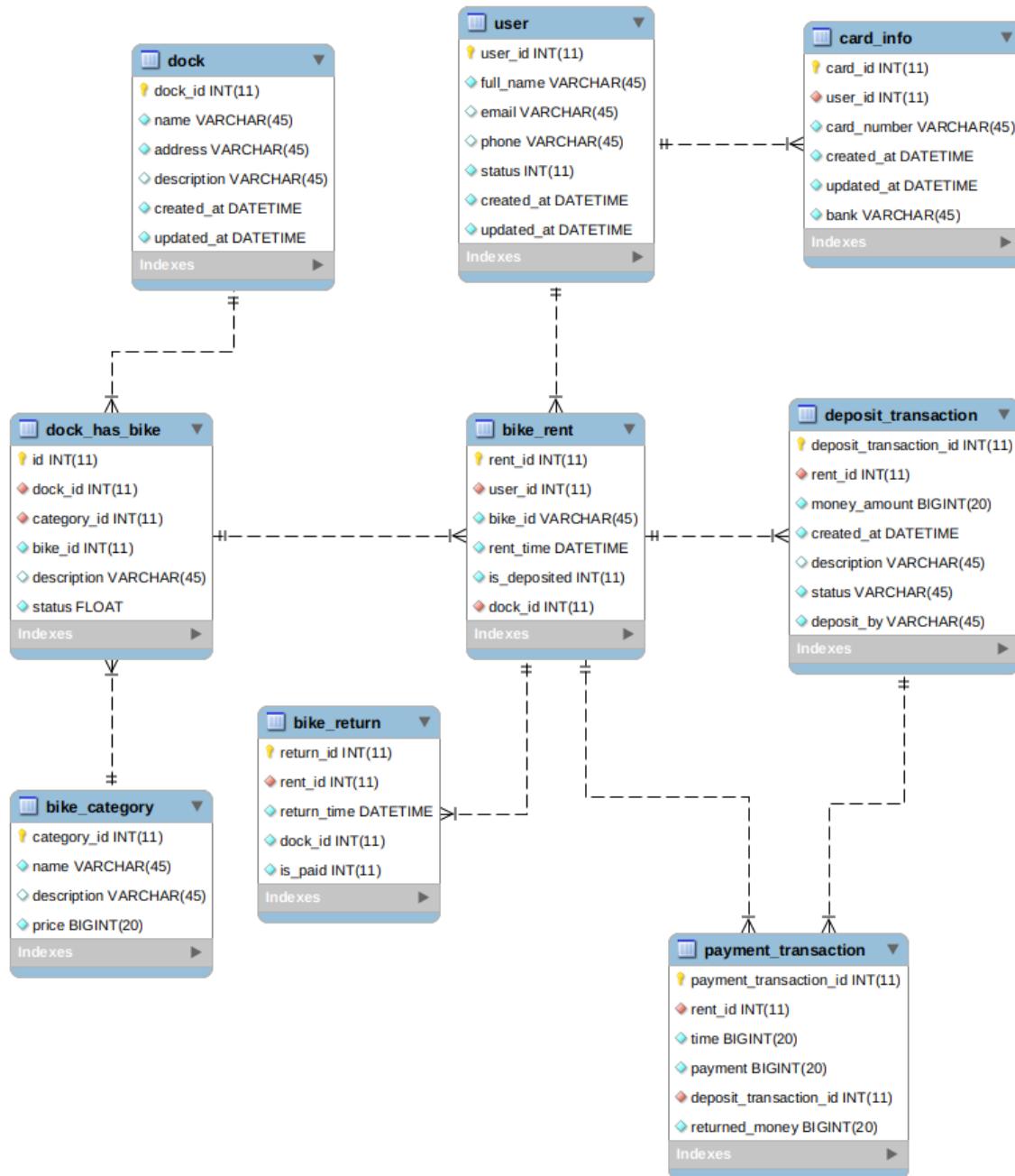
MySQL được lựa chọn do là một phần mềm mã nguồn mở đáng tin cậy và hoàn toàn phù hợp cho nhu cầu cần đến một hệ quản trị cơ sở dữ liệu có quan hệ.

4.2.2.2 Sơ đồ cơ sở dữ liệu

Quá trình thiết kế cơ sở dữ liệu từ E-R diagram:

Từ biểu đồ thực thể - liên kết, các bảng của cơ sở dữ liệu được dựng lên một cách tự nhiên thông qua các quan hệ

Sơ đồ thiết kế cơ sở dữ liệu:



4.2.2.3 Thiết kế chi tiết cơ sở dữ liệu

- dock

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		dock_id	INT(11)	NULL	Yes	Định danh của bãi xe
2			name	VARCHAR(45)	NULL	Yes	Tên bãi
3			address	VARCHAR(45)	NULL	Yes	Địa chỉ của bãi
4			description	VARCHAR(45)	NULL	Yes	Mô tả về bãi
5			create_at	DATETIME	NULL	no	Thời gian tạo
6			update_at	DATETIME	NULL	Yes	Thời gian cập nhật
7			province	String	NULL	Yes	Tỉnh thành
8			image	String	NULL	No	Hình ảnh
9			pointNumber	Integer	NULL	Yes	Số điểm đỗ

- Bike

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		id	INT(11)	NULL	Yes	
2		x	dockid	INT(11)	NULL	Yes	Đối tượng bãi xe
3		x	categoryid	INT(11)	NULL	Yes	id của category
4			active	INT(11)	NULL	Yes	Có hoạt động hay không

5			description	VARCHAR(45)	NULL	Yes	Mô tả
6			status	FLOAT	NULL	no	Trạng thái
7			point	INT(11)	NULL	Yes	Điểm đỗ xe
8			barcode	String	NULL	Yes	mã vạch của xe

- bike_category

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		category_id	INT(11)		Yes	ID
2			name	VARCHAR(45)		Yes	Tên loại xe
3			description	VARCHAR(45)		Yes	Mô tả
4			price	BIGINT(20)		Yes	Giá cả
5			image	String		No	Hình ảnh
6			Coefficient	float		Yes	Hệ số

- bike_rent

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		rend_id	INT(11)	NULL	Yes	ID
2		x	user_id	INT(11)	NULL	no	User's ID
3		x	bike_id	VARCHAR(45)	NULL	Yes	Bike's ID
4			rent_time	DATETIME	NULL	Yes	Thời điểm thuê xe

5			is_deposited	INT(11)	NULL	Yes	Xem đã đặt cọc hay chưa
6		x	dock_id	INT(11)	NULL	Yes	ID của bãi xe
7			rentTime	String	NULL	Yes	Thời gian thuê xe

- bike_return

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		return_id	INT(11)	NULL	Yes	Return's ID
2		x	rent_id	INT(11)	NULL	Yes	Rent's ID
3			return_time	DATETIME	NULL	Yes	Thời gian trả xe
4		x	dock_id	INT(11)	NULL	Yes	Dock's ID
5			is_paid	INT(11)	NULL	Yes	kiểm tra xem đã thanh toán hay chưa

- user

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		user_id	INT(11)	NULL	Yes	user's ID
2			full_name	VARCHAR(45)	NULL	Yes	Tên đầy đủ
3			email	VARCHAR(45)	NULL	Yes	Electronic mail
4			phone	VARCHAR(45)	NULL	Yes	Số điện thoại

5			status	INT(11)	NULL	Yes	Trạng thái
6			created_at	DATETIME	NULL	Yes	Ngày tạo
7			update_at	DATETIME	NULL	Yes	Ngày cập nhật

- card_info

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		card_id	INT(11)	NULL	Yes	Định danh của thẻ thanh toán
2		x	user_id	INT(11)	NULL	Yes	user's ID
3			card_number	VARCHAR(45)	NULL	Yes	Số thẻ
4			created_at	DATETIME	NULL	Yes	Ngày tạo
5			updated_at	DATETIME	NULL	Yes	Ngày cập nhật
6			bank	VARCHAR(45)	NULL	Yes	Ngân hàng cấp phát thẻ

- deposit_transaction

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		deposit_transaction_id	INT(11)	NULL	Yes	Định danh của giao dịch đặt cọc
2		x	rent_id	INT(11)	NULL	Yes	rent's ID
3			money_amount	BIGINT(20)	NULL	Yes	Số tiền đã đặt cọc
4			created_at	DATETIME	NULL	Yes	Ngày tạo

5			description	VARCHAR(45)	NULL	Yes	Mô tả
6			status	VARCHAR(45)	NULL	Yes	Trạng thái
7			deposit_by	VARCHAR(45)	NULL	Yes	Đặt cọc bởi

- payment_transaction

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		payment_transaction_id	INT(11)	NULL	Yes	Định danh của giao dịch thanh toán
2		x	rent_id	INT(11)	NULL	Yes	rent's ID
3			time	BIGINT(20)	NULL	Yes	Thời gian thuê xe
4			payment	BIGINT(11)	NULL	Yes	Số tiền
5		x	deposit_transaction_id	INT(11)	NULL	Yes	Định danh của giao dịch đặt cọc
6			returned_money	BIGINT(20)	NULL	Yes	Số tiền trả lại khách hàng

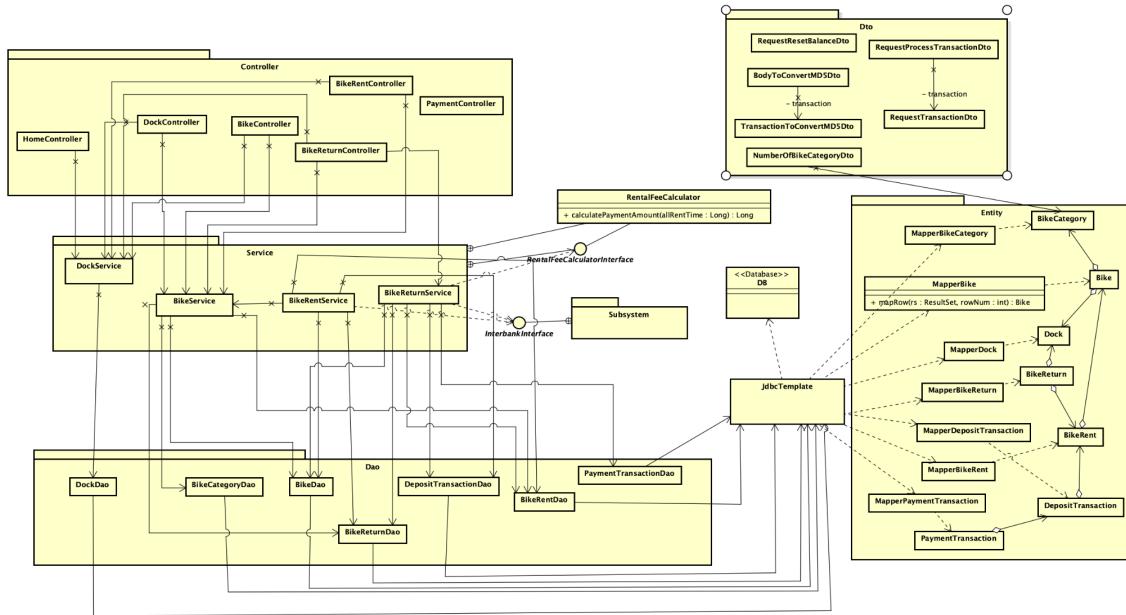
4.3 Non-Database Management System Files

Không có.

4.4 Thiết kế lớp

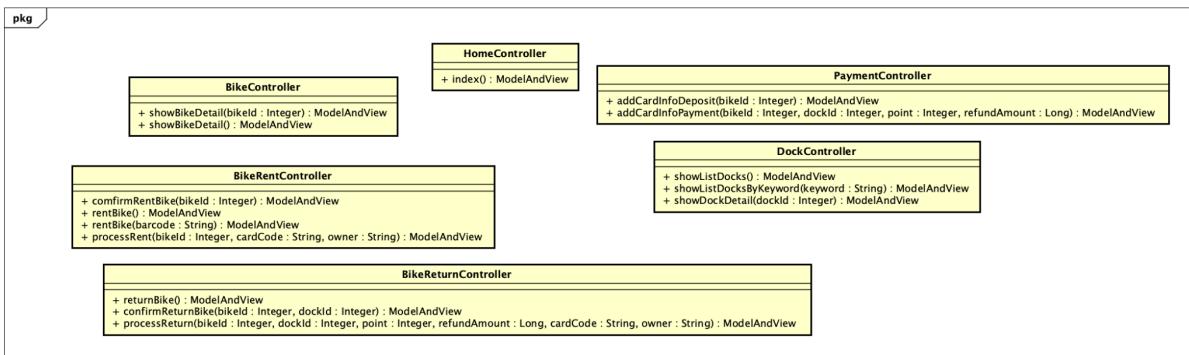
Trong thiết kế theo mô hình MVC, các nghiệp vụ thường được tách ra và nằm trong các lớp Service. Các lớp controller chỉ còn có nhiệm vụ điều hướng dữ liệu từ service ra giao diện người dùng.

4.4.1 Biểu đồ lớp tổng quan

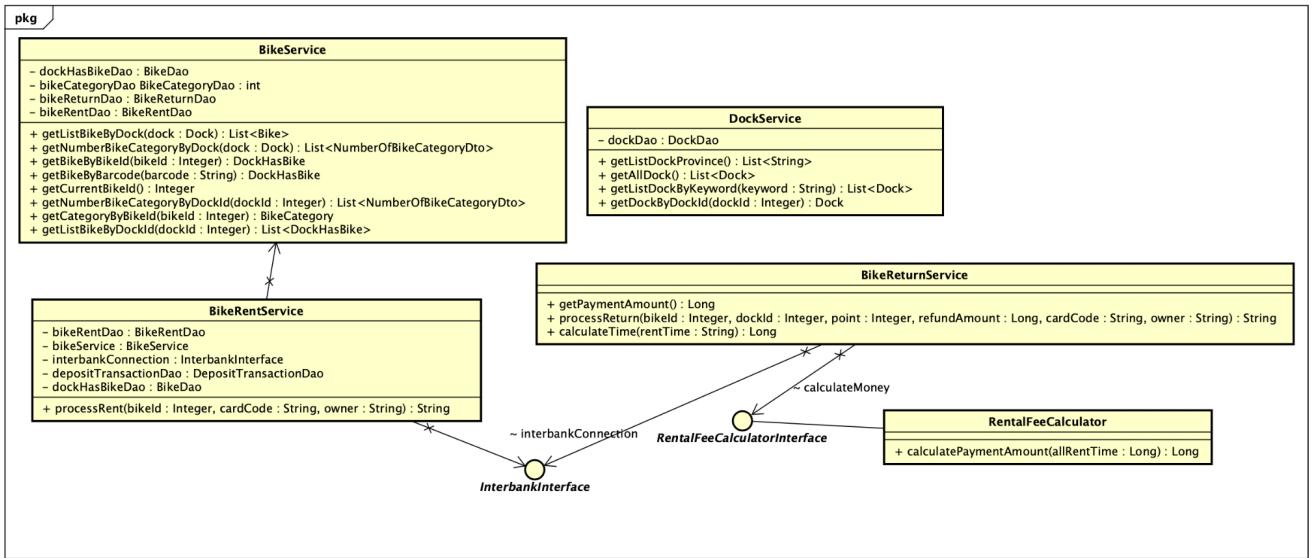


4.4.2 Biểu đồ lớp

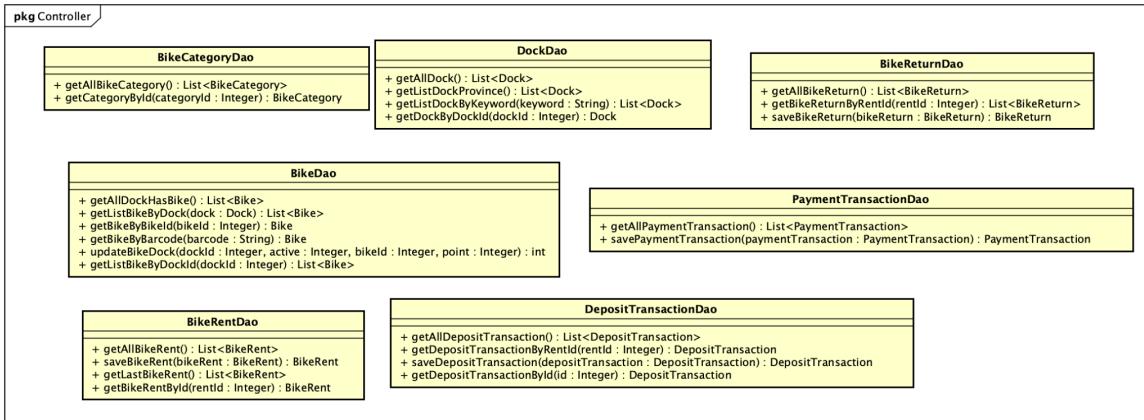
4.4.2.1 Biểu đồ lớp trong gói Controller



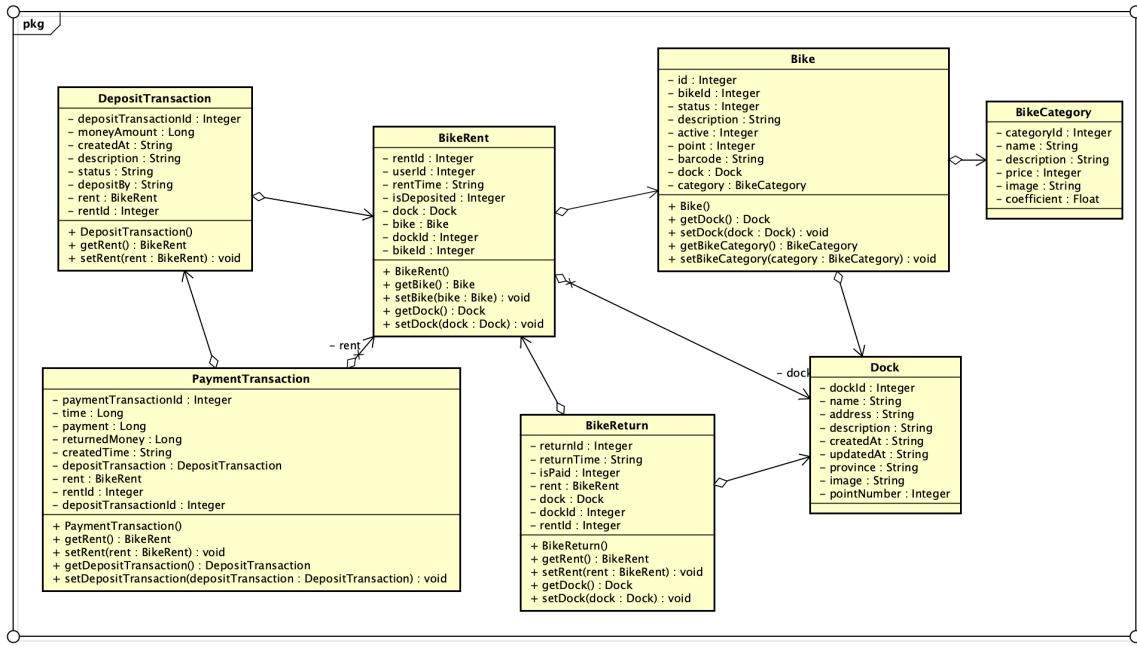
4.4.2.2 Biểu đồ lớp trong gói Service



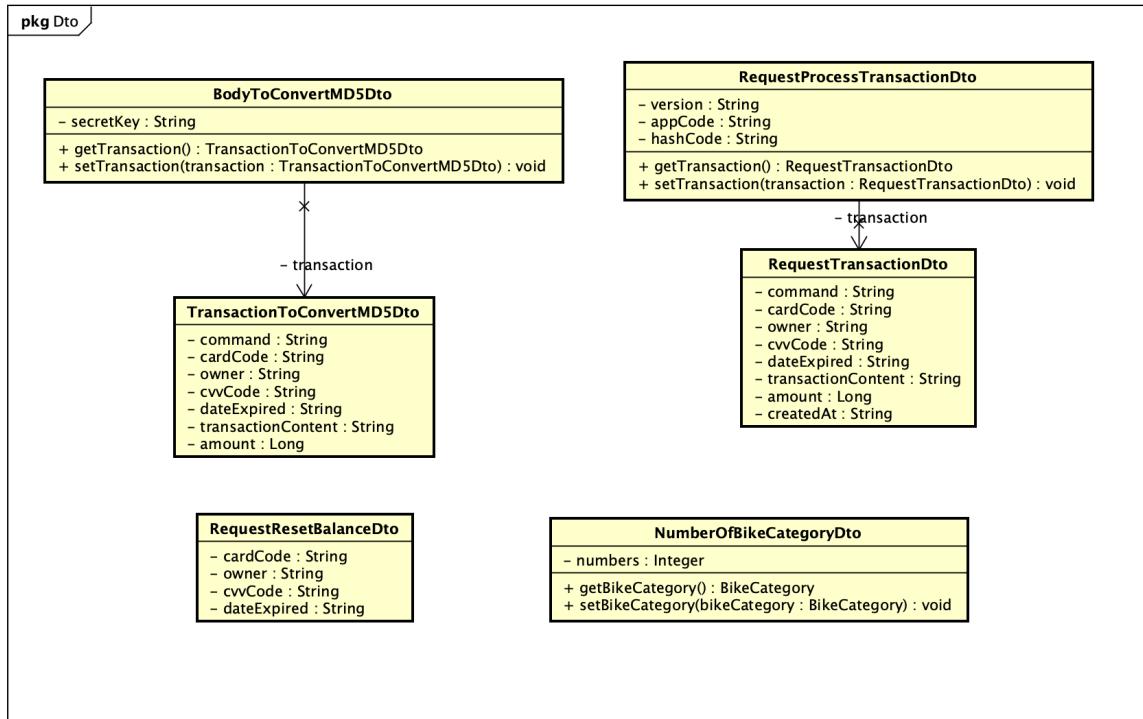
4.4.2.3 Biểu đồ lớp trong gói Dao



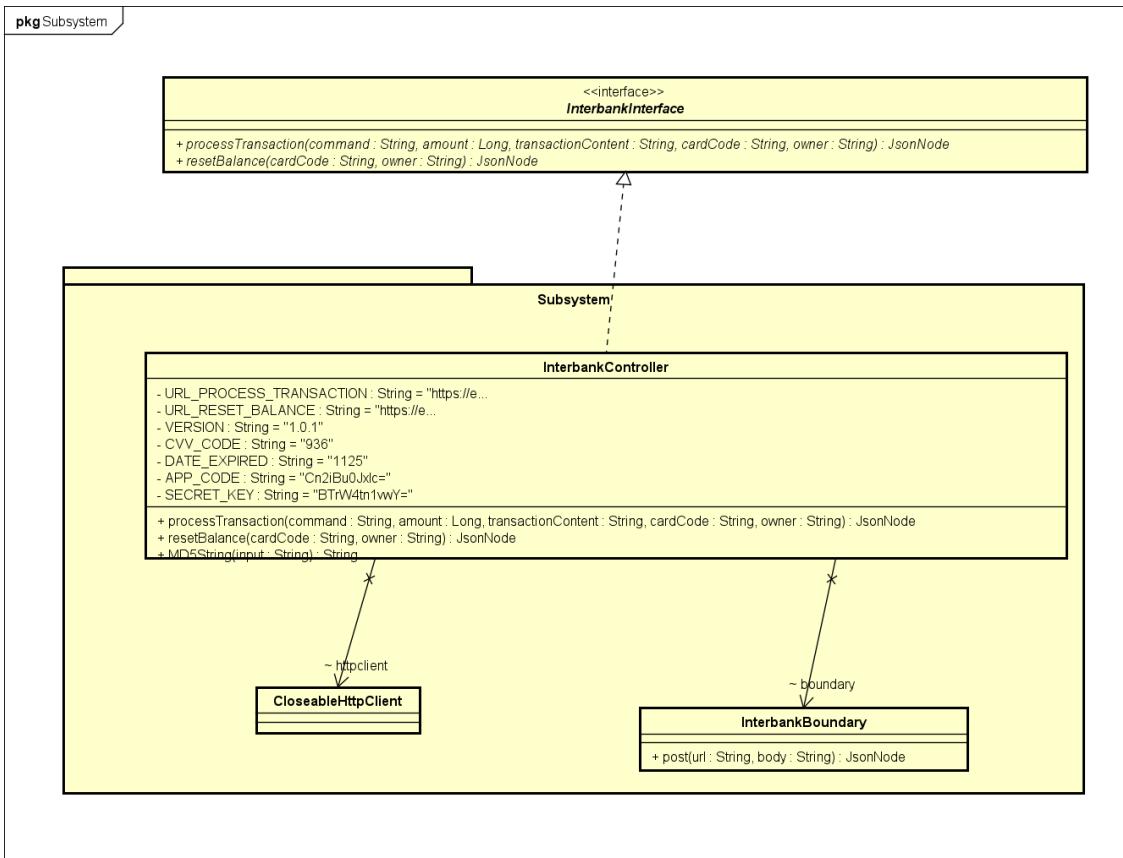
4.4.2.4 Biểu đồ lớp trong gói Entity



4.4.2.5 Biểu đồ lớp trong gói Dto

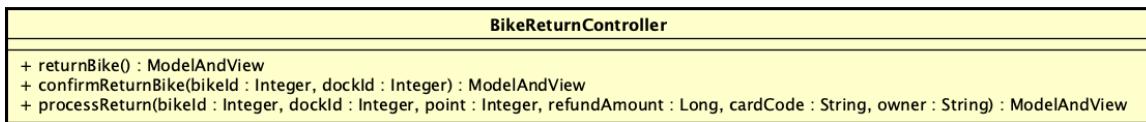


4.4.2.6 Biểu đồ subsystem



4.4.3 Thiết kế lớp

4.4.3.1 Lớp BikeReturnController



Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	returnBike	ModelAndView	Xử lý yêu cầu hiển thị danh sách các bãi xe khi khách hàng ấn nút Return Bike

2	confirmReturnBike	ModelAndView	Xử lý yêu cầu xác nhận trả xe về một bãi xe mà khách hàng đã chọn sau khi click vào nút confirmReturnBike
3	processReturn	ModelAndView	Xử lý yêu cầu trả xe của khách hàng

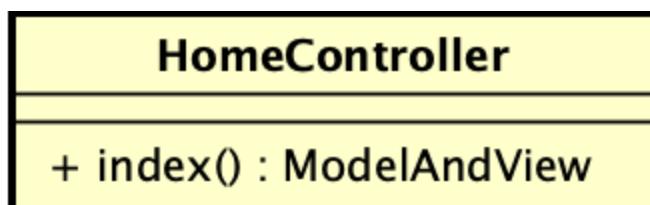
Tham số :

- bikeId : Định danh của xe
- dockId : Định danh của bãi
- Point : Điểm trả xe
- refund Amount : Số tiền cần hoàn trả
- cardCode: Mã thẻ của khách hàng
- owner: Chủ thẻ

Exception: Không

State: không

4.4.3.2 Lớp HomeController



Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	index()	ModelAndView	Xử lý yêu cầu trả về danh sách bãi xe và danh sách các tỉnh thành tương ứng
2			

Tham số :

- Không

Exception:

- Không

State

Không

4.4.3.3 Lớp BikeController



Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	showBikeDetail	ModelAndView	Xử lý yêu cầu xem thông tin chi tiết xe khi khách hàng click đúp vào 1 xe nào đó trong bãi.
2	showCurrentBike	ModelAndView	Xử lý yêu cầu xem thông tin chi tiết xe mà khách hàng đang thuê

Tham số :

- bikeld: Định danh của xe sẽ được hiển thị

Exception: Không

Method

Không

State

Không

4.4.3.4 Lớp BikeRentController

BikeRentController
+ comfirmRentBike(bikeld : Integer) : ModelAndView + rentBike() : ModelAndView + rentBike(barcode : String) : ModelAndView + processRent(bikeld : Integer, cardCode : String, owner : String) : ModelAndView

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	confirmRentBike	ModelAndView	Xử lý yêu cầu xác nhận thuê xe khi khách hàng click vào nút confirm Rent Bike
2	rentBike	ModelAndView	Xử lý yêu cầu chuyển đến màn hình thuê xe khi khách hàng đang xem thông tin xe đó trong bãi
3	rentBike	ModelAndView	Xử lý yêu cầu chuyển đến màn hình xe muốn thuê khi khách hàng nhập barcode tương ứng từ màn hình và click vào button rent bike
4	processRent	ModelAndView	Xử lý yêu cầu thuê xe của khách hàng

Tham số :

- bikeld: Định danh của xe
- barcode: mã vạch của xe
- cardCode: Mã thẻ của khách hàng
- owner: Chủ thẻ
-

Exception:

- Không

Method

Không

State

Không

4.4.3.5 Lớp DockController

DockController
+ showListDocks() : ModelAndView + showListDocksByKeyword(keyword : String) : ModelAndView + showDockDetail(dockId : Integer) : ModelAndView

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	showListDock	ModelAndView	Xử lý yêu cầu hiện ra danh sách các bãi xe đã có
2	showListDockByKeyWord	ModelAndView	Xử lý yêu cầu hiện ra danh sách các xe theo từ khoá mà người dùng đã nhập
3	showDockDetail	ModelAndView	Xử lý yêu cầu hiển thị chi tiết thông tin một bãi xe mà khách hàng đã click đúp vào.

Tham số :

- keyword: Từ khoá mà khách hàng nhập vào
- dockId: Định danh của bãi xe

Exception:

-

State: Không

4.4.3.6 Lớp PaymentController

PaymentController
+ addCardInfoDeposit(bikeId : Integer) : ModelAndView
+ addCardInfoPayment(bikeId : Integer, dockId : Integer, point : Integer, refundAmount : Long) : ModelAndView

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	addCardInfoDeposit	ModelAndView	Xử lý yêu cầu hiển thị màn hình thêm thông tin thẻ để đặt cọc của khách hàng
2	addCardInfoPayment	ModelAndView	Xử lý yêu cầu hiển thị màn hình thêm thông tin thẻ để thanh toán của khách hàng

Tham số :

- bikeId: Định danh của xe
- dockId: Định danh của bãi xe
- point: Điểm còn trống trong bãi mà khách hàng muốn trả xe
- refundAmount: Là số tiền hoàn lại

Exception:

- Không

State

Không

4.4.3.7 Lớp BikeRentService

BikeRentService
<ul style="list-style-type: none"> - bikeRentDao : BikeRentDao - bikeService : BikeService - interbankConnection : InterbankInterface - depositTransactionDao : DepositTransactionDao
+ processRent(bikeId : Integer, cardCode : String, owner : String) : String

Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	bikeRentDao	BikeRentDao	Null	Đối tượng tương tác với Database
2	bikeService	BikeService	Null	Lớp dịch vụ xe
3	interbankConnection	InterbankInterface	Null	Giao diện của interbank
4	depositTransactionDao	DepositTransactionDao	Null	Đối tượng tương tác với Database
5				

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	processRent	String	Xử lý thuê xe và trả về thông báo thành công hay thất bại
2			

Tham số :

- bikeID: Định danh xe
- cardCode: Mã thẻ
- owner: Chủ thẻ

Exception: không

State: Không

4.4.3.8 Lớp BikeReturnService

BikeReturnService
+ getPaymentAmount() : Long + processReturn(bikeId : Integer, dockId : Integer, point : Integer, refundAmount : Long, cardCode : String, owner : String) : String + calculateTime(rentTime : String) : Long

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getPaymentAmount	Long	Xử lý yêu cầu tính ra số tiền một lần giao dịch thuê
2	processReturn	String	Xử lý yêu cầu trả xe và trả về thành công hay thất bại

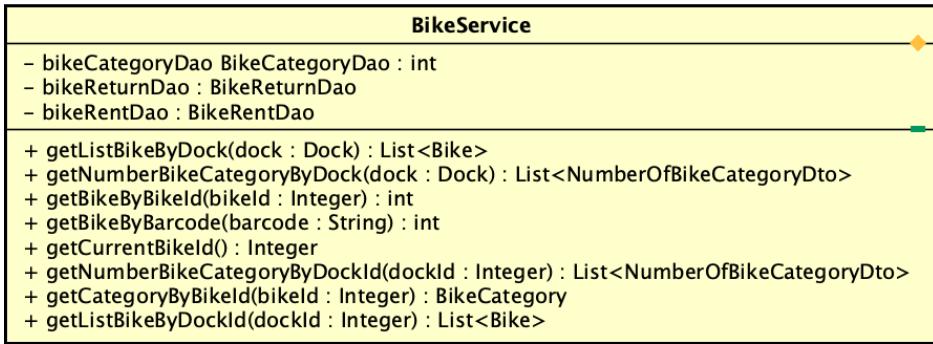
Tham số :

- bikeId: Định danh xe mà khách hàng muốn trả xe
- dockId: Định danh bãi xe mà khách hàng muốn trả xe
- point: Điểm đỗ xe
- cardCode: Mã thẻ
- owner: Chủ thẻ

Exception: không

State: Không

4.4.3.9 Lớp BikeService



Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	bikeCategory Dao	BikeCategory Dao	NULL	Lớp tương tác với database
2	bikeReturnDa o	BikeRetur nDao	NULL	Lớp tương tác với database
3	bikeRentDao	BikeRent Dao	NULL	Lớp tương tác với database

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getListBikeByDock Id	List<DockHasBike>	Xử lý yêu cầu lấy danh sách các xe trong bãi
2	getNumberBikeCat egoryByDockId	List<NumberOfBike CategoryDto>	Xử lý yêu cầu lấy số các loại xe trong bãi
3	getBikeByBikeId	DockHasBike	Xử lý yêu cầu lấy thông tin một xe bởi định danh của xe đó
4	getCategoryByBike Id	BikeCategory	Xử lý yêu cầu lấy thông tin loại xe
5	getBikeByBarcode	DockHasBike	Xử lý yêu cầu lấy thông tin xe tương ứng với Barcode mà khách hàng nhập vào

6	getCurrentBikeId	Integer	Xử lý yêu cầu lấy định danh của xe
---	------------------	---------	------------------------------------

Tham số :

- dockId: Định danh của bãi xe
- bikeId: Định danh của xe
- barcode: Mã vạch tương ứng của một xe
- dock: đối tượng bãi xe

Exception:

- Không

State: Không

4.4.3.10 Lớp DockService

DockService	
- dockDao : DockDao	
+ getListDockProvince() : List<String>	
+ getAllDock() : List<Dock>	
+ getListDockByKeyword(keyword : String) : List<Dock>	
+ getDockByDockId(dockId : Integer) : Dock	

Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	dockDao	DockDao	NULL	Lớp tương tác với database
2				

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích

1	getListDockProvince	List<String>	Xử lý yêu cầu lấy thông tin địa chỉ của bãi xe
2	getAllDock	List<Dock>	Xử lý yêu cầu lấy thông tin danh sách tất cả đối tượng bãi xe
3	getListDockByKeyword	List<Dock>	Xử lý yêu cầu lấy thông tin danh sách các đối tượng bãi xe theo từ khoá
4	getDockById	Dock	Xử lý yêu cầu lấy thông tin đối tượng bãi xe bởi định danh của bãi đó

Tham số :

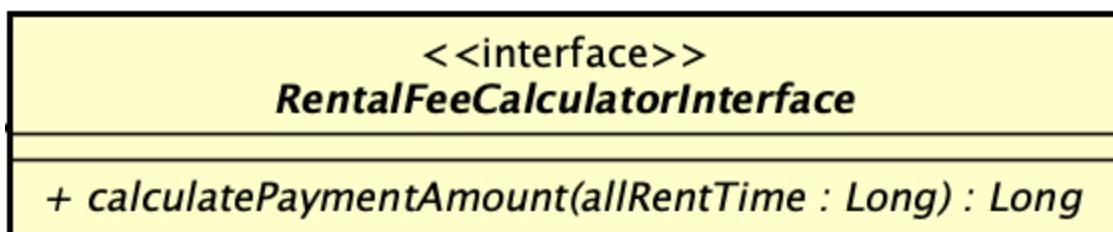
- keyword: Giá trị mặc định là Null, là từ khoá mà khách hàng nhập
- dockId: Định danh của bãi xe

Exception:

- Không

State: Không

4.4.3.11 Lớp RentalFeeCalculatorInterface



Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	calculatePaymentAmount	Long	Xử lý việc tính số tiền cần thanh toán
2			

Tham số :

- allRentTime: Thời gian thuê xe

Exception: không

State: không

4.4.3.12 Lớp BikeCategoryDao

BikeCategoryDao
+ getAllBikeCategory() : List<BikeCategory>
+ getCategoryById(categoryId : Integer) : BikeCategory

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getAllBikeCategory	List<BikeCategory>	Thực hiện việc truy vấn tất cả các dữ liệu BikeCategory trong Database và trả về đối tượng tương ứng
2	getCategoryById	BikeCategory	Thực hiện việc truy vấn dữ liệu BikeCategory khi cung cấp định danh và trả về đối tượng tương ứng.

Tham số :

- categoryId: Định danh của đối tượng BikeCategory

Exception:

- Không

State: không

4.4.3.13 Lớp DockDao

DockDao
+ getAllDock() : List<Dock> + getListDockProvince() : List<Dock> + getListDockByKeyword(keyword : String) : List<Dock> + getDockByDockId(dockId : Integer) : Dock

<i>Phương thức</i>			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getAllDock	List<Dock>	Xử lý truy vấn tất cả dữ liệu Dock trong database và trả lại danh sách đối tượng tương ứng.
2	getListDockProvince	List<Dock>	Xử lý truy vấn tất cả dữ liệu địa chỉ của bãi xe trong cơ sở dữ liệu
3	getListDockByKeyword	List<Dock>	Xử lý truy vấn danh sách các dữ liệu dock theo keyword trong database và trả lại danh sách các đối tượng tương ứng
4	getDockByDockId	Dock	Xử lý truy vấn dữ liệu bãi xe trong database được định danh bởi DockId và trả về đối tượng tương ứng.

Tham số:

- keyword: Từ khoá
- dockId: Định danh của bãi xe

4.4.3.14 Lớp BikeRentDao

BikeRentDao
+ getAllBikeRent() : List<BikeRent>
+ saveBikeRent(bikeRent : BikeRent) : BikeRent
+ getLastBikeRent() : List<BikeRent>

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getAllBikeRent	List<BikeRent>	Xử lý truy vấn dữ liệu tất cả các xe đã thuê và trả về 1 danh sách tất cả các đối tượng tương ứng
2	saveBikeRent	BikeRent	Xử lý yêu cầu lưu trữ thông tin xe đã thuê vào cơ sở dữ liệu
3	getLastBikeRent	List<BikeRent>	Xử lý truy vấn dữ liệu của xe cuối cùng đã thuê và trả về đối tượng tương ứng

Tham số :

- bikeRent: Đối tượng xe đã được thuê

Exception:

- Không

State

Không

4.4.3.15 Lớp BikeReturnDao

BikeReturnDao
+ getAllBikeReturn() : List<BikeReturn> + getBikeReturnByRentId(rentId : Integer) : List<BikeReturn> + saveBikeReturn(bikeReturn : BikeReturn) : BikeReturn

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getAllBikeReturn	List<BikeReturn>	Xử lý truy vấn dữ liệu danh sách thông tin lần trả xe về
2	getBikeReturnByRentId	List<BikeReturn>	Xử lý truy vấn dữ liệu thông tin lần trả xe về trong cơ sở dữ liệu và trả về đối tượng tương ứng
3	saveBikeReturn	BikeReturn	Xử lý việc lưu thông tin lần trả xe về trong cơ sở dữ liệu.

Tham số :

- rentId: Định danh xe đã thuê
- bikeReturn: Thông tin một lần thuê xe

Exception:

- Không

State

Không

4.4.3.16 Lớp DepositTransactionDao

DepositTransactionDao
+ getAllDepositTransaction() : List<DepositTransaction> + getDepositTransactionByRentId(rentId : Integer) : DepositTransaction + saveDepositTransaction(depositTransaction : DepositTransaction) : DepositTransaction + getDepositTransactionById(id : Integer) : DepositTransaction

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getAllDepositTransaction	List<DepositTransaction>	Xử lý truy vấn tất cả các thông tin giao dịch đặt cọc và trả về đối tượng tương ứng
2	getDepositTransactionByRentId	DepositTransaction	Xử lý truy vấn thông tin giao dịch đặt cọc trong cơ sở dữ liệu khi truyền vào RentID
3	saveDepositTransaction	DepositTransaction	Lưu trữ giao dịch đặt cọc vào trong cơ sở dữ liệu

Tham số :

- rentId: Định danh lần thuê xe
- depositTransaction: Đối tượng entity giao dịch đặt cọc

Exception:

- Không

State: không

4.4.3.17 Lớp PaymentTransactionDao

PaymentTransactionDao
+ getAllPaymentTransaction() : List<PaymentTransaction> + savePaymentTransaction(paymentTransaction : PaymentTransaction) : PaymentTransaction

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getAllPaymentTransactions	List<PaymentTransaction>	Xử lý truy vấn thông tin tất cả giao dịch thanh toán và trả về 1 danh sách các đối tượng tương ứng
2	savePaymentTransaction	PaymentTransaction	Lưu trữ giao dịch thanh toán vào trong cơ sở dữ liệu

Tham số :

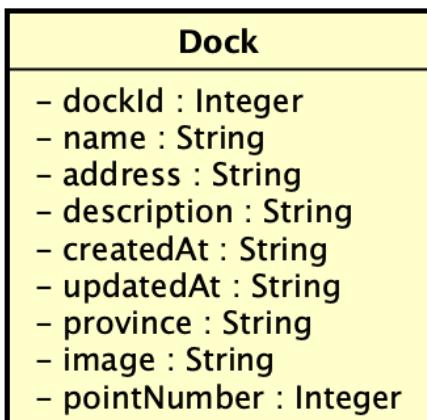
- paymentTransaction: Là Entity giao dịch thanh toán.

Exception:

- Không

State: Không

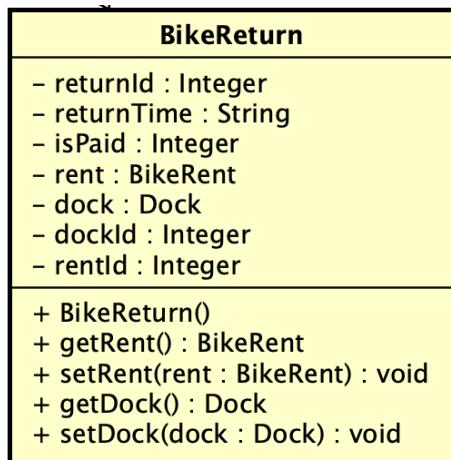
4.4.3.18 Lớp Dock



Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	dockId	Integer	NULL	Định danh của bãi xe
2	name	String	NULL	Tên của bãi xe

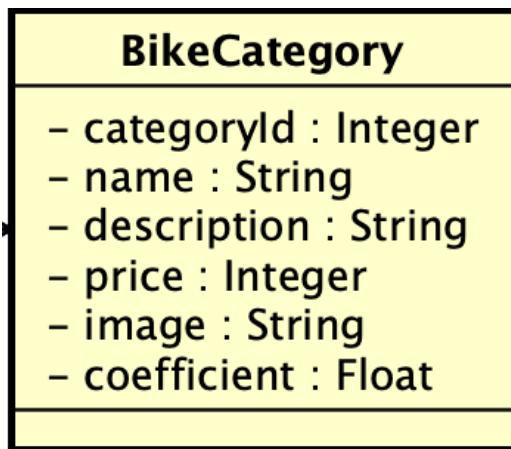
3	address	String	NULL	Địa chỉ của bãi xe
4	description	String	NULL	Mô tả về bãi xe
5	createdAt	String	NULL	Thời gian tạo
6	updateAt	String	NULL	Thời gian cập nhật
7	province	String	NULL	Khu vực của bãi xe
8	image	String	NULL	Đường dẫn đến hình ảnh bãi xe
9	pointNumber	Integer	NULL	Số lượng điểm đỗ xe trong bãi

4.4.3.19 Lớp BikeReturn



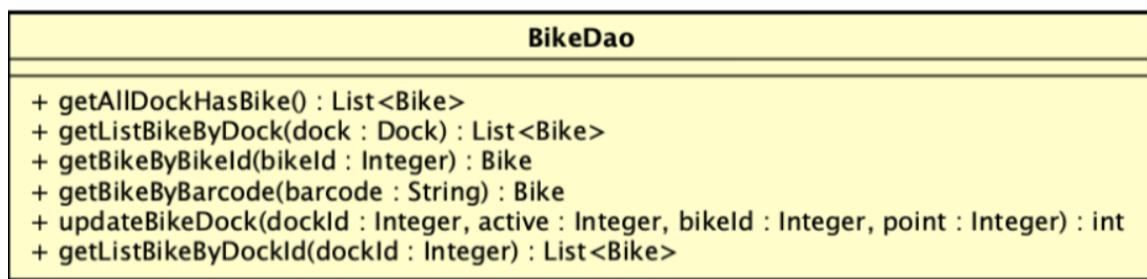
Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	returnId	Integer	NULL	Định danh của 1 lần trả xe
2	rentId	Integer	NULL	Định danh của lần thuê xe
3	returnTime	String	NULL	Thời gian trả xe
4	dockId	Integer	NULL	Định danh của bãi xe đã trả xe
5	isPaid	Integer	NULL	Đã thanh toán hay chưa

4.4.3.20 Lớp BikeCategory



Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	categoryId	Integer	NULL	Định danh của loại xe
2	name	String	NULL	Tên của loại xe
3	description	String	NULL	Mô tả về loại xe
4	price	Integer	NULL	Giá xe
5	image	String	NULL	Đường dẫn đến hình ảnh xe

4.4.3.21 Lớp BikeDao



Phương thức

#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getAllBike	List<Bike>	Lấy danh sách tất cả các bike
2	getListBikeByDock	List<Bike>	Lấy danh sách các Bike theo từng bến xe
3	getBikeById	Bike	Lấy danh sách các Bike theo định danh của bike
4	getBikeByBarcode	Bike	Xử lý truy vấn lấy Bike theo mã barcode
5	updateBikeDock	int	Cập nhật BikeDock
6	getListBikeByDock Id	List<Bike>	Lấy danh sách các Bike theo định danh của bến xe

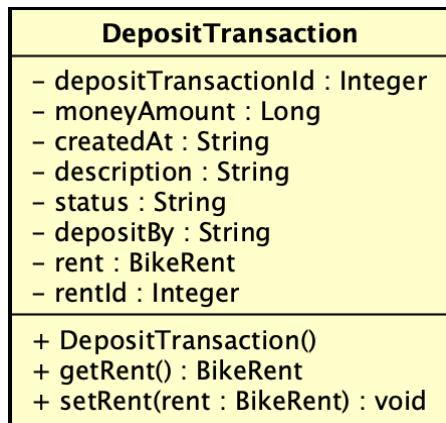
4.4.3.22 Lớp BikeRent



Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	rentId	Integer	NULL	Định danh của lần thuê xe
2	userId	Integer	NULL	Định danh của người thuê xe
3	bikeId	Integer	NULL	Định danh của xe đã được thuê
4	rentTime	String	NULL	Thời gian thuê

5	isDeposited	Integer	NULL	Số tiền đặt
6	dockId	Integer	NULL	Định danh của bãi thuê xe

4.4.3.23 Lớp DepositTransaction



Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	depositTransactionId	Integer	NULL	Định danh của giao dịch đặt cọc
2	rentId	Integer	NULL	Định danh của lần thuê xe
3	moneyAmount	Long	NULL	Số tiền đã trả
4	createdAt	String	NULL	Thời gian tạo
5	description	String	NULL	Mô tả
6	status	String	NULL	Trạng thái của xe
7	depositBy	String	NULL	Đặt cọc bởi ai

4.4.3.24 Lớp PaymentTransaction

PaymentTransaction	
- paymentTransactionId : Integer	
- time : Long	
- payment : Long	
- returnedMoney : Long	
- createdTime : String	
- depositTransaction : DepositTransaction	
- rent : BikeRent	
- rentId : Integer	
- depositTransactionId : Integer	
+ PaymentTransaction()	
+ getRent() : BikeRent	
+ setRent(rent : BikeRent) : void	
+ getDepositTransaction() : DepositTransaction	
+ setDepositTransaction(depositTransaction : DepositTransaction) : void	

Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	paymentTransactionId	Integer	NULL	Định danh của giao dịch thanh toán
2	rentId	Integer	NULL	Định danh của lần thuê xe
3	time	Long	NULL	Thời gian thuê
4	payment	Long	NULL	Số tiền phải trả
5	depositTransactionId	Integer	NULL	Định danh của lần đặt cọc
6	returnedMoney	Long	NULL	Số tiền trả lại
7	createdTime	String	NULL	Thời gian tạo giao dịch

4.4.3.25 Lớp BodyToConvertMD5Dto

BodyToConvertMD5Dto	
- secretKey : String	
+ getTransaction() : TransactionToConvertMD5Dto	
+ setTransaction(transaction : TransactionToConvertMD5Dto) : void	

Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	secretKey	String	NULL	Khoá bí mật
2				

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getTransaction	TransactionToConvertMD5DTO	Xử lý yêu cầu lấy giao dịch
2	setTransaction	void	Xử lý yêu cầu thiết lập giao dịch.

Tham số :

- transaction: Giao dịch

Exception:

- Không

State

Không

4.4.3.26 Lớp RequestProcessTransactionDto

RequestProcessTransactionDto	
- version : String	
- appCode : String	
- hashCode : String	
+ getTransaction() : RequestTransactionDto	
+ setTransaction(transaction : RequestTransactionDto) : void	

Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	version	String	NULL	Phiên bản
2	appCode	String	NULL	Mã app
3	hashCode	String	NULL	Mã băm

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getTransaction	RequestTransactionDto	Xử lý yêu cầu lấy giao dịch
2	setTransaction	void	Xử lý yêu cầu thiết lập giao dịch.

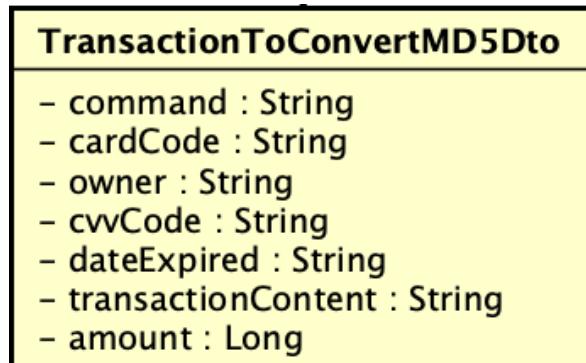
Tham số :

- transaction: Giao dịch

Exception: không

State: không

4.4.3.27 Lớp TransactionToConvertMD5Dto



Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	command	String	NULL	Lệnh yêu cầu
2	cardCode	String	NULL	Mã thẻ
3	owner	String	NULL	Chủ thẻ
4	cvvCode	String	NULL	Mã dùng để xác minh thẻ
5	dateExpired	String	NULL	Ngày hết hạn
6	transaction Content	String	NULL	Nội dung giao dịch
7	amount	Long	NULL	Số tiền giao dịch

4.4.3.28 Lớp RequestTransactionDto

RequestTransactionDto	
-	command : String
-	cardCode : String
-	owner : String
-	cvvCode : String
-	dateExpired : String
-	transactionContent : String
-	amount : Long
-	createdAt : String

Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	command	String	NULL	Lệnh yêu cầu
2	cardCode	String	NULL	Mã thẻ
3	owner	String	NULL	Chủ thẻ
4	cvvCode	String	NULL	Mã dùng để xác minh thẻ
5	dataExpire d	String	NULL	Ngày hết hạn
6	transaction Content	String	NULL	Nội dung giao dịch
7	amount	String	NULL	Số tiền giao dịch
8	createdAt	String	NULL	Thời gian tạo

4.4.3.29 Lớp RequestResetBalanceDto

RequestResetBalanceDto				
<ul style="list-style-type: none"> - cardCode : String - owner : String - cvvCode : String - dateExpired : String 				

Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	cardCode	String	NULL	Mã thẻ
2	owner	String	NULL	Chủ thẻ
3	cvvCode	String	NULL	Mã dùng để xác minh thẻ
4	dataExpire d	String	NULL	Ngày hết hạn

4.4.3.30 Lớp NumberOfBikeCategoryDto

NumberOfBikeCategoryDto				
<ul style="list-style-type: none"> - numbers : Integer 				
<ul style="list-style-type: none"> + getBikeCategory() : BikeCategory + setBikeCategory(bikeCategory : BikeCategory) : void 				

Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	numbers	Integer	NULL	Số lượng của loại xe
2				

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	getBikeCategory	BikeCategory	Xử lý yêu cầu lấy ra loại xe
2	setBikeCateger	void	Xử lý yêu cầu thiết lập 1 loại xe

Tham số :

- bikeCategory: Là đối tượng loại xe
-

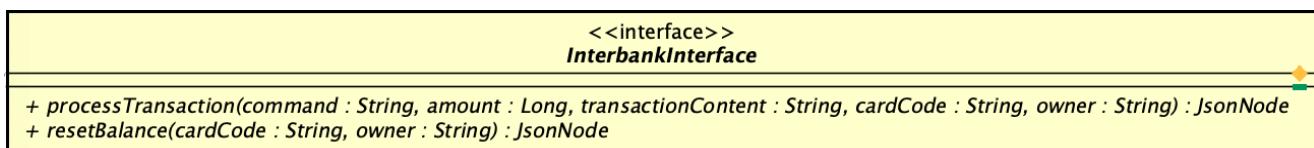
Exception:

- Không

State

Không

4.4.3.31 InterbankInterface



Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	processTransaction	JsonNode	Xử lý giao dịch
2	resetBalance	JsonNode	Thiết lập lại số dư

Tham số :

- command: Lệnh truyền vào để thực thi, có thể là “Pay” hoặc “refund”
- amount: Số tiền trong lần giao dịch này
- transactionContent: Nội dung giao dịch
- cardCode: Mã thẻ

- owner: Chủ thẻ

Exception:

- Không

State

Không

4.4.3.32 Lớp InterbankController

InterbankController
<ul style="list-style-type: none"> - URL_PROCESS_TRANSACTION : String = "https://e... - URL_RESET_BALANCE : String = "https://e... - VERSION : String = "1.0.1" - CVV_CODE : String = "936" - DATE_EXPIRED : String = "1125" - APP_CODE : String = "Cn2iBu0Jxlc=" - SECRET_KEY : String = "BTrW4tn1vwY=" <ul style="list-style-type: none"> + processTransaction(command : String, amount : Long, transactionContent : String, cardCode : String, owner : String) : JsonNode + resetBalance(cardCode : String, owner : String) : JsonNode + MD5String(input : String) : String

Thuộc tính				
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	URL_PROCESS_TRANSACTION	Integer	NULL	Đường dẫn URL dùng để xử lý giao dịch
2	URL_RESET_BALANCE	String	NULL	Đường dẫn URL dùng để reset lại số dư
3	VERSION	String	NULL	Phiên bản
4	CVV_CODE	String	NULL	Mã dùng để valid card
5	DATE_EXPIRED	String	NULL	Ngày hết hạn
6	APP_CODE	String	NULL	Mã app sử dụng hệ thống thanh toán
7	SECRET_KEY	String	NULL	Khoá bí mật

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	processTransaction	JsonNode	Xử lý yêu cầu lấy ra loại xe
2	resetBalance	void	Xử lý yêu cầu thiết lập 1 loại xe
3	MD5String	String	Xử lý băm một xâu bằng MD5

Tham số :

- command: Yêu cầu giao dịch
- Amount: Số tiền
- transactionContent: Nội dung giao dịch
- cardCode: Mã thẻ thanh toán
- owner : Chủ tài khoản
- String: Một chuỗi kí tự

Exception:

- Không

State

Không

4.4.3.33 InterbankBoundary

InterbankBoundary
+ post(url : String, body : String) : JsonNode

Phương thức			
#	Tên	Kiểu dữ liệu trả về	Mô tả mục đích
1	post	JsonNode	Phương thức giúp gọi các API dạng Post

Tham số :

- url: Đường dẫn
- body: Nội dung request gửi đi

Exception:

- Không

State

Không

5 Design Considerations

5.1 Mục tiêu và định hướng

Những dòng code đầu tiên trong bản thiết kế luôn là một bản nháp và quá trình phát triển để hoàn thiện, biến nó trở thành kiệt tác luôn cần nhiều trải qua nhiều giai đoạn hay các bước như phân tích, thiết kế, xây dựng, kiểm thử, bảo trì v...v. Do đó đòi hỏi các nhà quản lý, nhà phát triển phần mềm phải có mục tiêu hay đường lối chính sách đúng đắn.

Mục tiêu:

- Tạo ra sản phẩm tốt, chất lượng, đáp ứng được nhu cầu của người dùng thuê xe trong bối cảnh các phương tiện khác ảnh hưởng tiêu cực đến môi trường.
- Sử dụng các nguyên lý cơ bản của thiết kế hướng đối tượng và thiết kế theo module

Định hướng:

- Tạo một bản thiết kế làm cho các nhà phát triển dễ **tái sử dụng**, dễ **đọc**, dễ **hiểu**, dễ **truyền thông**, dễ **mở rộng** và dễ **bảo trì**.

Nhóm đã chia module một cách đủ nhỏ để có thể thiết kế, lập trình, kiểm thử một cách độc lập.

Tuân thủ Java Coding Convention để khiêm cho bản thiết kế dễ hiểu và trao đổi giữa các thành viên dễ dàng hơn.

Thực hiện nguyên lý Open/Closed Principle khiêm cho bản thiết kế dễ mở rộng và dễ bảo trì.

Nhóm đã giải thích code tỉ mỉ, chi tiết, viết java doc cho các module để tăng tính tái sử dụng và dễ bảo trì sau này.

Chính sách thiết kế và lập trình

- Cố gắng tạo ra bản thiết kế low coupling, high cohesion và tuân thủ các nguyên lý design principle
- Thiết kế mang tính module hoá.
- Sử dụng phần mềm như Eclipse để lập trình, là công cụ hiệu quả nhất để gỡ lỗi, ngoài ra Eclipse gồm có các tính năng tuyệt vời với một số hỗ trợ Plugin và Tích hợp. Trên hết Eclipse là mã nguồn mở và hoàn toàn miễn phí.
- Sử dụng phần mềm Screen Builder để giúp tăng tốc quá trình xây dựng giao diện bản mẫu trước một cách nhanh chóng mà không cần phải coding.

- Tái sử dụng các thư viện có sẵn như Maven, Tomcat.

5.2 Architectural Strategies

Các quyết định ảnh hưởng đến toàn bộ hệ thống:

1. Sử dụng mô hình thác nước là mô hình để phát triển phần mềm do quy trình làm việc của toàn bộ học kỳ này
2. Sử dụng ngôn ngữ lập trình Java.
3. Sử dụng mô hình Spring MVC (đã giới thiệu ở phần hai).
4. Sử dụng MySQL là công cụ quản lý cơ sở dữ liệu.
5. Sử dụng Interface Paradigm là Technology Paradigm.
6. Sử dụng github:
 - a. Là công cụ giúp quản lý source code tổ chức theo dạng dữ liệu phân tán.
 - b. Giúp đồng bộ source code của team lên 1 server.
 - c. Hỗ trợ các thao tác kiểm tra source code trong quá trình làm việc (diff, check modifications, show history, merge source, ...)
7. Nhóm sẽ mở rộng phần mềm không chỉ ở Thái Bình hay Hà Nội mà còn nhiều tỉnh thành lân cận khác để mọi người dễ có thể tiếp cận được ứng dụng. .

Trong bản thiết kế, toàn bộ các luồng hoạt động có thay đổi đến dữ liệu của hệ thống đều được cập nhật vào trong cơ sở dữ liệu. Đây là một quyết định với mục đích ưu tiên tính đúng đắn và chắc chắn của dữ liệu, khi mà dữ liệu không được lưu trên memory mà luôn được ghi vào một persistent storage. Tuy nhiên, sự chắc chắn này đánh đổi lại việc hiệu năng của hệ thống sẽ bị phụ thuộc vào hiệu năng đọc/ghi của cơ sở dữ liệu.

5.3 Coupling and Cohesion

5.3.1 Phân tích tính Coupling giữa các lớp và giữa các gói với nhau.

1. Content Coupling

Không có

2. Common Coupling

Không có module nào chứa Global data.

3. Control Coupling

Method ProcessTransaction của InterbankInterface chứa tham số điều khiển command

```

    /**
     * Description: method process return a bike
     * @param bikeId: id of the bike returning
     * @param dockId: id of the dock want to renting
     * @param point: point want to renting
     * @param refundAmount: the refund money system give to user
     * @return String: status of process: success or fail
     */
    public String processReturn(Integer bikeId, Integer dockId, Integer point, Long refundAmount, String cardCode, String owner) {
        try {
            JsonNode result;
            //refund or pay more
            if (refundAmount > 0) {
                result = interbankConnection.processTransaction("refund", refundAmount, "Hoan tien thue xe", cardCode, owner);
            } else {
                result = interbankConnection.processTransaction("pay", -refundAmount, "Tra tien thue xe", cardCode, owner);
                if (result.get("errorCode").asText().equals("02")) {
                    interbankConnection.resetBalance(cardCode, owner);
                    result = interbankConnection.processTransaction("pay", -refundAmount, "Tra tien thue xe", cardCode, owner);
                }
            }
        }
    }

```

Ở trên ta thấy module bikeRentService phải biết luồng hoạt động bên trong của module Interbank. Ta sẽ loại bỏ coupling này bằng cách tách ra thành 2 module processPayTransaction và module processRefundTransaction

4. Stamp Coupling

Không có

5. Data Coupling

Không có

5.3.2 Phân tích tính cohesion của các lớp

1. Coincidental Cohesion

Không có

2. Logical cohesion

Không có

3. Temporal cohesion

Không có

4. Procedural cohesion

Không có

5. Communicational cohesion

Các phương thức trong class InterbankSubsystemController có thể được coi là đạt mức communicational cohesion. Các method này đều hoạt động

dựa trên cùng một thông tin chung và cùng nằm trong một luồng xử lý giao dịch.

6. Sequential cohesion

Không có

7. Informational cohesion

Các lớp service mà nhóm tạo ra đều có thể thoả mãn Informational Cohesion. Chúng có có thể khác nhau ở input output nhưng cùng thao tác trên một tập dữ liệu chung là một số các DAO mà nó cần tiếp cận đến.

8. Functional cohesion

Ta thấy các lớp thuộc gói DAO thoả mãn tính functional Cohesion. Xét một ví dụ là lớp BikeRentDao. Lớp này có 3 phương thức là getAllBikeRent, saveBikeRent, getLastBikeRent. Các phương thức này đều sử dụng chung một thuộc tính jdbcTemplate để truy vấn đến cơ sở dữ liệu và đều thực hiện chung mục đích liên quan đến dữ liệu thuê xe.

5.4 Design Principles

Thiết kế của nhóm đã nhắm đến việc tuân thủ 5 nguyên tắc SOLID. Tuy nhiên, do điều kiện về thời gian, kiến thức và kinh nghiệm còn thiếu, nên thiết kế chưa thể hoàn thiện như mong muốn. Dưới đây, nhóm sẽ bàn luận về thiết kế dưới 5 nguyên tắc SOLID.

5.4.1. Single responsibility

Nhóm luôn cố gắng thiết kế các module thực hiện một nhiệm vụ duy nhất mà tiêu biểu là các lớp thuộc gói Service. Lấy một ví dụ là lớp DockService gồm các hàm đều sử dụng chung một attribute dockDao và chỉ thực hiện một mục đích chung là lấy danh sách các bãi xe

```

① /**
 * Description: Class Service execute the logic action of dock
 *
 */
② @Service
public class DockService {
    ③     @Autowired
    ④     DockDao dockDao;

    ⑤     public List<String> getListDockProvince() {
        ⑥         try {
            ⑦             List<String> listProvince = new ArrayList<String>();
            ⑧             List<Dock> listDockProvince = dockDao.getListDockProvince();

            // get list province of system
            ⑨             for (Dock dock : listDockProvince) {
                ⑩                 listProvince.add(dock.getProvince());
            }
            ⑪             return listProvince;
        } catch(Exception e) {
            ⑫                 return null;
        }
    }

    ⑬     public List<Dock> getAllDock() {
        ⑭         try {
            ⑮             List<Dock> listDock = dockDao.getAllDock();
            ⑯             return listDock;
        } catch(Exception e) {
            ⑰                 return null;
        }
    }

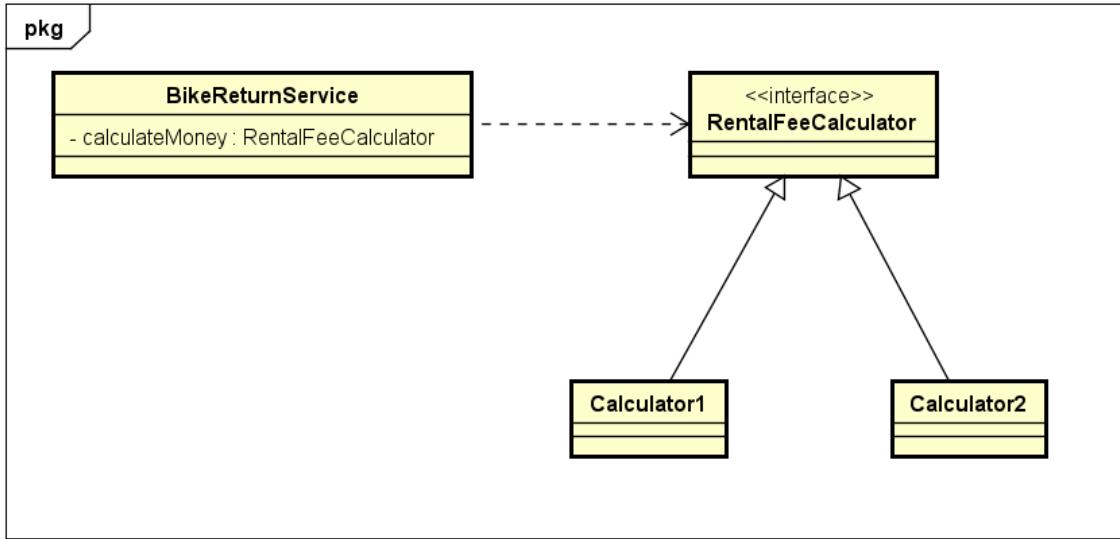
    ⑱     public List<Dock> getListDockByKeyword(String keyword) {
        ⑲         try {
            ⑳             List<Dock> listDock = dockDao.getListDockByKeyword(keyword);
            ⑳             return listDock;
        } catch(Exception e) {
            ⑳                 return null;
        }
    }

    ⑳     public Dock getDockByDockId(Integer dockId) {
        ⑳         try {
            ⑳             Dock dock = dockDao.getDockByDockId(dockId);
            ⑳             return dock;
        } catch (Exception e) {
            ⑳                 return null;
        }
    }
}

```

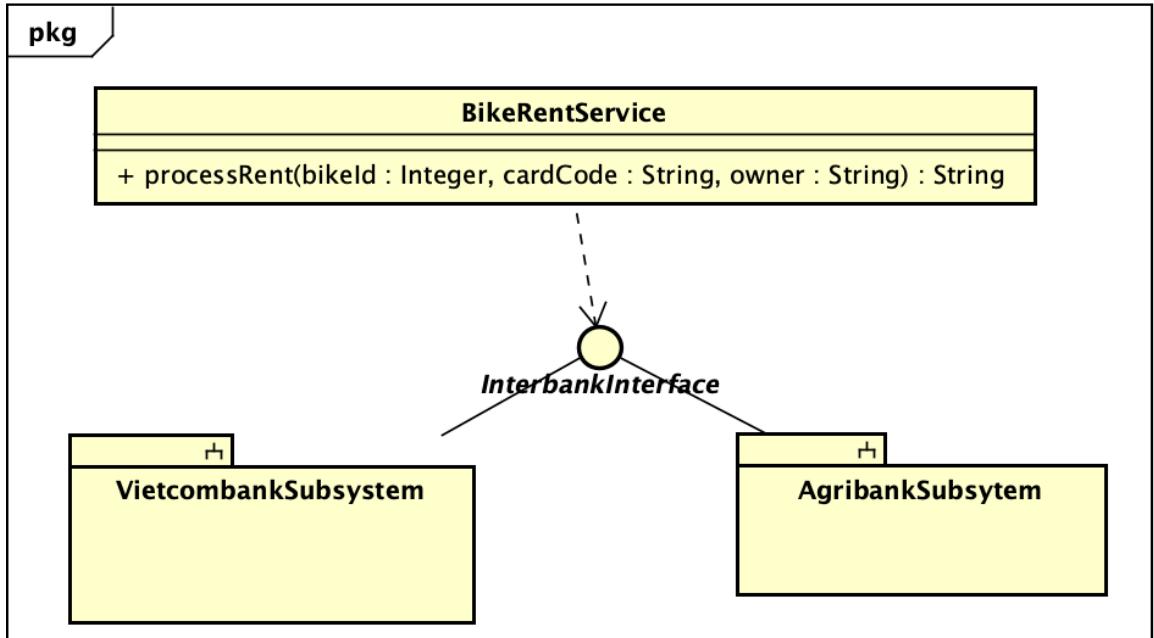
5.4.2. Open/Closed principle

Module tính tiền thuê xe:



Ta có thể mở rộng thêm nhiều cách tính tiền khác hay thay đổi lớp Calculator mà không ảnh hưởng đến lớp Client. Nói một cách khác Interface RentalFeeCalculator close để có thể sửa đổi và lớp thực thi Calculator open để có thể mở rộng.

Module InterbankSubsystem:



Thay vì để một lớp phụ thuộc chặt chẽ vào một ngân hàng khi gọi đến một nghiệp vụ thanh toán nào đó thì ta sẽ đóng gói các hàm xử lý giao dịch với ngân hàng bằng một hệ thống con và các lớp bên ngoài có thể gọi đến hệ thống con đó

thông qua giao diện. Từ đó ta sẽ có thể mở rộng chức năng thanh toán với nhiều ngân hàng khác mà không làm thay đổi bên trong Client.

5.4.3. Liskov substitution

Không có

5.4.4. Interface segregation

Ở nguyên lý này, thay vì sử dụng một interface quá lớn, quá nhiều phương thức, ta nên tách nhỏ interface này ra làm nhiều interface con cụ thể hơn. Các interface con sẽ trở nên dễ sử dụng hơn, đồng thời không bị vướng phải tình trạng phải cài đặt các phương thức không cần thiết.

Thiết kế hiện tại đảm bảo khá tốt nguyên lý này. Tuy nhiên, có một điểm cần cân nhắc, đó là InterbankInterface hiện tại đang yêu cầu phương thức xử lý giao dịch cho cả giao dịch thanh toán và hoàn tiền. Điều này khiến giao diện trở nên không còn hợp lý khi một ngân hàng nào đó chỉ thanh toán mà không hoàn tiền, trong khi interface của Subsystem lại mong chờ tính năng này.

5.4.5. Dependency inversion

Nguyên lí này cho rằng: thành phần cụ thể của hệ thống chỉ nên phụ thuộc vào thành phần trừu tượng hơn nó, không phải chiều ngược lại.

Hiện tại, toàn bộ subsystem interbank đang phụ thuộc vào một phương thức thanh toán, thể hiện qua tham số cardCode và owner được quy định bởi InterbankInterface. Thiết kế này chịu rủi ro khi một phương thức thanh toán khác xuất hiện mà không tương thích với cơ chế xác thực sử dụng cardCode và owner. Vì vậy, thiết kế cho hệ thống con Interbank cần được xem xét lại.

5.5 Design Patterns

5.5.1. DAO - Data Access Object pattern

DAO pattern là một mẫu thiết kế thuộc nhóm cấu trúc (Structural pattern). DAO tạo ra một giao diện trong suốt giữa ứng dụng với một loại cơ sở dữ liệu hoặc bất cứ cơ chế lưu trữ dữ liệu cố định nào. Theo cách này, các service được che dấu khỏi các hoạt động truy xuất cơ sở dữ liệu ở mức thấp. Nhờ có pattern này, toàn bộ DB schema hoàn toàn trở nên trong suốt đối với ứng dụng, hay nói cách khác, ứng dụng không cần quan tâm đến nguồn dữ liệu.

Pattern này cung cấp cho phần mềm EcoBikeRental khả năng thay đổi cơ sở dữ liệu một cách dễ dàng, miễn là cơ sở dữ liệu mới chứa đầy đủ thông tin mà phần

mềm cần (bao gồm các thông tin về bãi xe, xe đạp, xe đạp được đỗ trong các bãi xe, người dùng, thông tin thuê & trả xe, thông tin giao dịch).

Trong mã nguồn của EcoBikeRental, các lớp trong module DAO cài đặt các câu truy vấn SQL để truy cập đến cơ sở dữ liệu. Module này cũng là module duy nhất thao tác trực tiếp với cơ sở dữ liệu.

5.5.2. DTO - Data transfer object pattern

DTO pattern là một mẫu thiết kế thuộc nhóm kiến trúc (Architectural pattern). DTO được sử dụng khi chúng ta muốn truyền dữ liệu qua lại giữa các tầng của một ứng dụng. Transfer object ở đây đơn giản là một POJO (plain old java object), chỉ chứa các getter/setter và có thể cài đặt serialize để truyền tải dữ liệu qua mạng.

Trong phần mềm EcoBikeRental, các lớp DTO được sử dụng để trao đổi dữ liệu giữa ứng dụng với hệ thống ngân hàng. DTO giúp việc tách biệt giữa logic và dữ liệu một cách rõ ràng: transfer object chỉ chứa dữ liệu, còn logic được cài đặt ở các lớp xử lý khác. DTO cũng hay được sử dụng cùng với DAO.

5.5.3. Strategy pattern

Strategy pattern là một mẫu thiết kế thuộc nhóm thể hiện (Behavioral pattern) cho phép chương trình có khả năng lựa chọn giữa nhiều cách cài đặt khác nhau của cùng một service tại runtime. Thay vì cài đặt duy nhất một giải thuật, chương trình tự lựa chọn một trong nhiều lựa chọn giải thuật có thể sử dụng, tùy vào sự kiện diễn ra tại thời điểm chạy. Các giải thuật (strategies) khác nhau được đóng gói riêng rẽ, không phụ thuộc vào phần sử dụng đến nó, và có thể được tái sử dụng ở nhiều nơi mà không bị lặp lại.

Trong phần mềm EcoBikeRental, strategy pattern được sử dụng nhằm đáp ứng được sự thay đổi trong cách tính tiền thuê xe. Trong BikeReturnService, phần tính tiền thuê xe được trừu tượng hóa bằng một interface phục vụ việc tính tiền. Để thêm các cách tính tiền khác nhau, việc cần làm chỉ là cài đặt theo interface trên.

