TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

**BÁO CÁO**

*Thiết kế và xây dựng phần mềm- IT4490*

*EcoBike Rental*

Họ và tên: Phương Trung Đức

Mã số sinh viên: 20173030

Lớp: KHMT.04-K62

Nhóm : TKXDPM-KHMT-K62C-01

Giảng viên hướng dẫn : TS.Trịnh Tuấn Đạt

***Hà Nội, 22 tháng 12 năm 2020***

Báo cáo cá nhân:

Use – case phụ trách: Thuê xe

Contents

[1, Use case tổng quan hệ thống 3](#_heading=h.30j0zll)

[2, Đặc tả use-case thuê xe 3](#_heading=h.1fob9te)

[3, Thiết kế kiến trúc/phân tích use case 4](#_heading=h.3znysh7)

[3.1, Biểu đồ trình tự use-case thuê xe 4](#_heading=h.2et92p0)

[3.2: Biểu đồ lớp thiết kế use – case thuê xe 5](#_heading=h.tyjcwt)

[4. Thiết kế giao diện 5](#_heading=h.3dy6vkm)

[4.1: Thiết kế màn hình 5](#_heading=h.1t3h5sf)

[4.2: Đặc tả màn hình 6](#_heading=h.4d34og8)

[5. Thiết kế chi tiết lớp 6](#_heading=h.2s8eyo1)

[5.1: Biểu đồ trình tự use-case thuê xe 6](#_heading=h.17dp8vu)

[5.2: Biểu đồ chi tiết lớp cho use-case thuê xe 8](#_heading=h.3rdcrjn)

[6. Lập trình và kiểm thử đơn vị. 8](#_heading=h.26in1rg)

[6.1: Kiểm thử hộp đen 8](#_heading=h.lnxbz9)

[6.2: Kiểm thử hộp trắng 9](#_heading=h.35nkun2)

[6.3: Kiểm thử tích hợp 9](#_heading=h.1ksv4uv)

[7. Chỉnh sửa lại thiết kế 10](#_heading=h.44sinio)

# 1, Use case tổng quan hệ thống

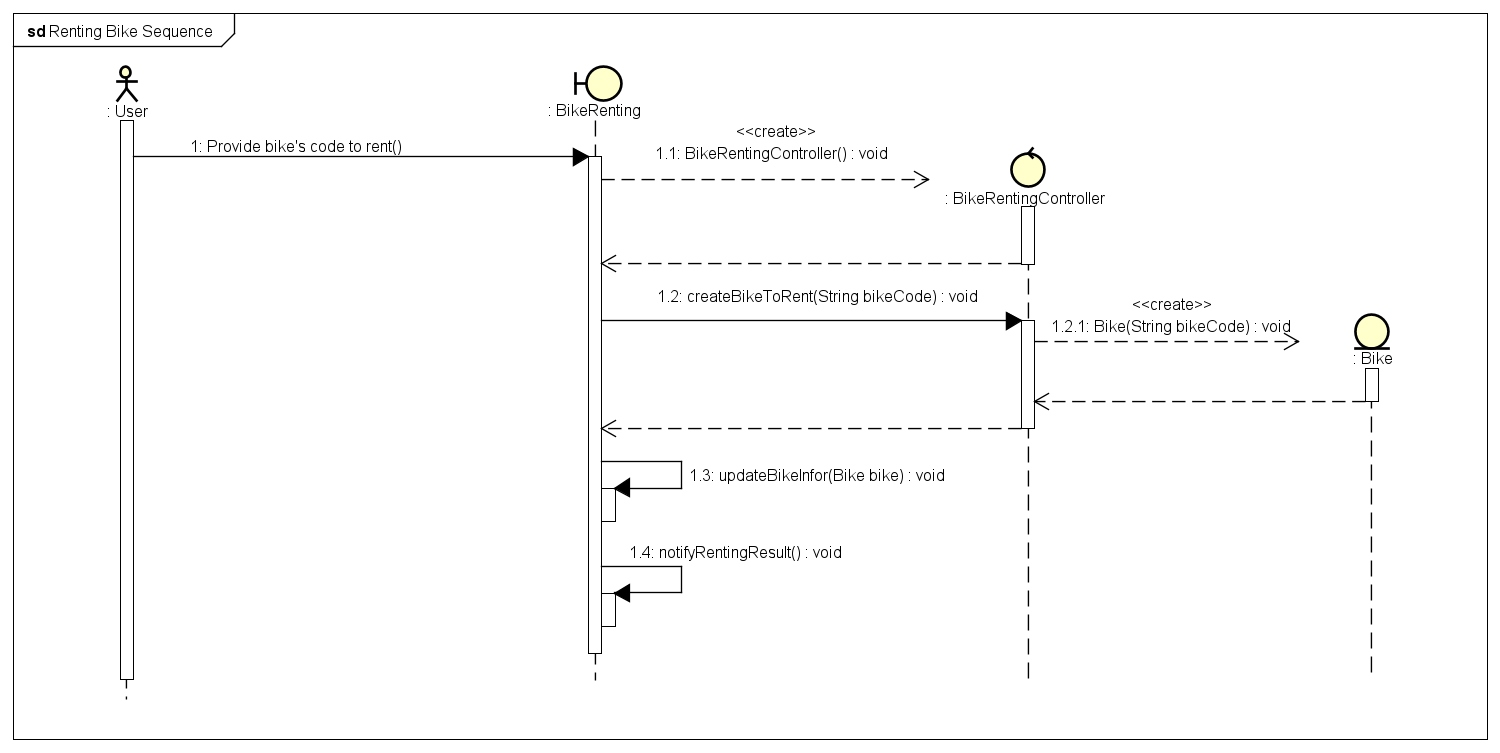


# 2, Đặc tả use-case thuê xe

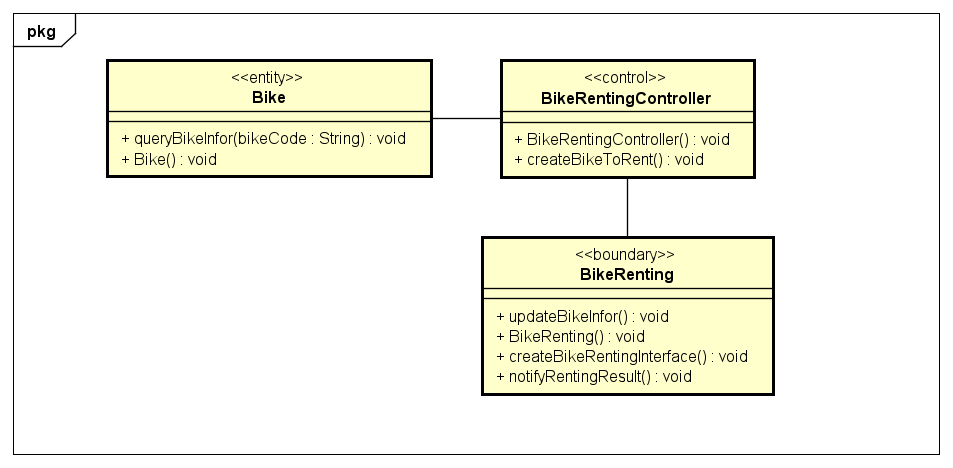
|  |  |  |  |
| --- | --- | --- | --- |
| **UC Code** | UC001 | **Use case name** | Thuê xe |
| **Actor** | Người dùng , EcoBikeApi | | |
| **Precondition** | None | | |
| **Main flow of event (success)** | |  |  |  | | --- | --- | --- | | **#** | **Doer** | **Action** | |  | Người dùng | Nhập mã xe muốn thuê | |  | Hệ thống | Kiếm tra mã xe khách hàng nhập | |  | Hệ thống | Hiển thị thông tin xe | |  | Người dùng | Ấn nút “Thuê xe” để thực hiện thuê xe có mã đã nhập | |  | Hệ thống | Yêu cầu người dùng thanh toán cho việc thuê xe bằng việc hiện form nhập mã thẻ | |  | Người dùng | Nhập thông tin thẻ tín dụng | |  | EcoBikeApi | Kiểm tra thông tin thẻ tín dụng | |  | EcoBikeApi | Gửi thông báo cho hệ thống | |  | Hệ thống | Thông báo kết quả thuê xe cho khách hàng | | | |
| **Alternative flow of event** | |  |  |  | | --- | --- | --- | | **#** | **Doer** | **Action** | | 2a. | Hệ thống | Thông báo “Mã xe không chính xác” khi mã đã nhập không trùng với bất kì xe nào | | 2b. | Hệ thống | Thông báo “xe đã được thuê” nếu mã xe có trạng thái “ĐÃ THUÊ” | | 4a. | Hệ thống | Thông báo “Chưa có xe được chọn” nếu chưa có thông tin của bất kì xe nào | | 9a. | Hệ thống | Thông báo “Thông tin thẻ tín dụng không chính xác” nếu không thể xác nhận thẻ | | 9b. | Hệ thống | Thông báo ”Không đủ tiền trong tài khoản” nếu số dư < tiền thuê xe + tiền cọc | | 9c. | Hệ thống | Thông báo ”Thẻ đã được sử dụng” nếu thẻ đã được dùng để thuê một xe khác | | | |
| **Post condition** | None | | |

# 3, Thiết kế kiến trúc/phân tích use case

## 3.1, Biểu đồ trình tự use-case thuê xe



## 3.2: Biểu đồ lớp thiết kế use – case thuê xe



# 4. Thiết kế giao diện

## 4.1: Thiết kế màn hình



# 4.2: Đặc tả màn hình

- Đặc tả màn hình

|  |  |  |  |
| --- | --- | --- | --- |
|  | Control | Operation | Function |
| Ô nhập liệu để nhập mã xe | Khởi tạo | - Người dùng nhập mã code vào ô nhập liệu |
| Các ô thông tin chi tiết xe | Sau khi người dùng nhập đúng code xe và focus out khỏi ô mã xe | Hiển thị các trường thông tin tương ứng của xe ứng với mã code |
| Nút thuê xe | Click | Xác nhận việc thuê xe có mã code tương ứng đã nhập |

- Khai báo các trường thuộc tính

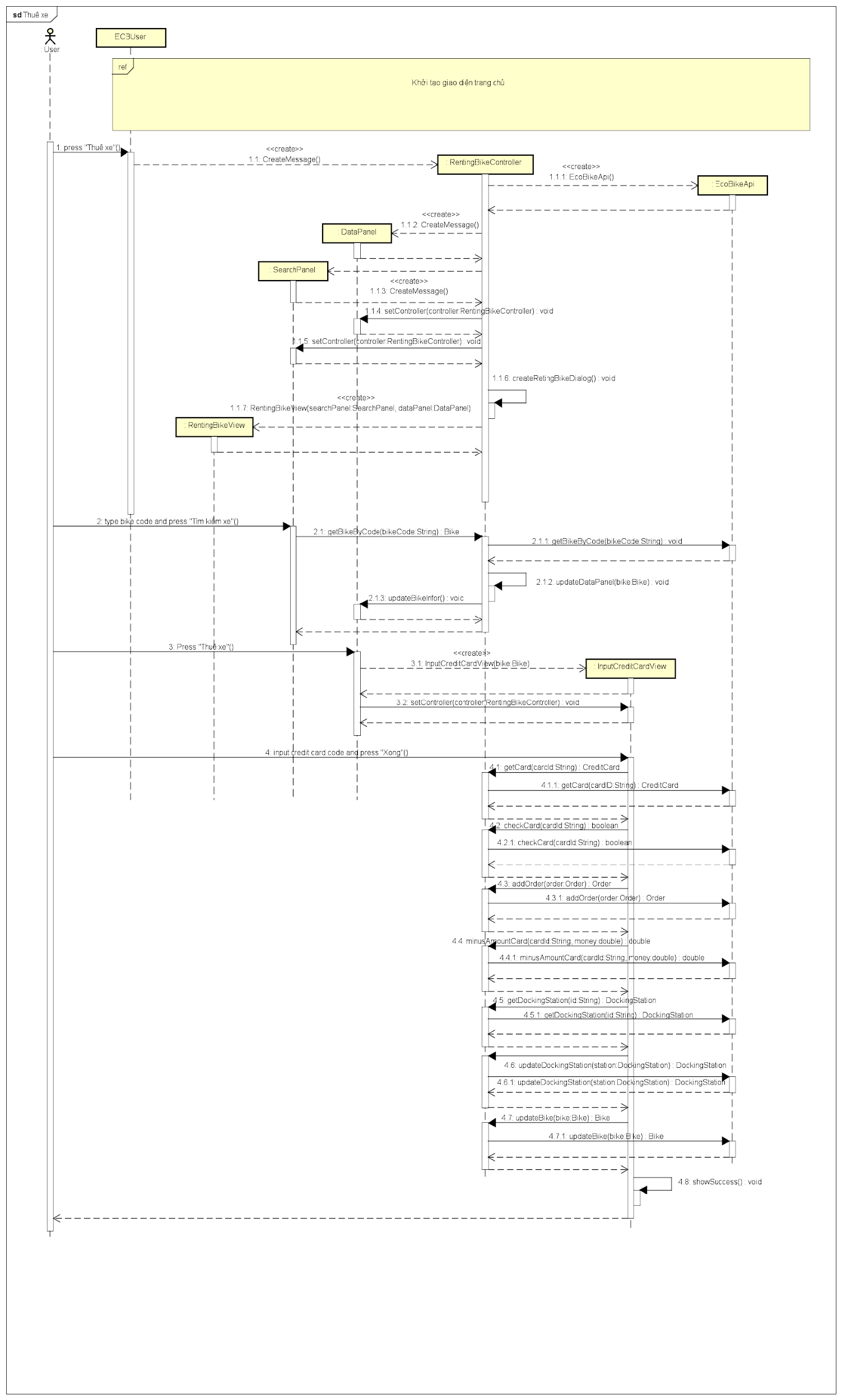
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên thuộc tính | Số lượng kí tự (bytes) | Loại | Trường thuộc tính | Nhận xét |
| Mã xe | 8 | String | Đen |  |
| Tên xe | 8 | String | Đen |  |
| Trọng lượng | 4 | Numerial | Đen |  |
| Biển số xe | 11 | String | Đen |  |
| Ngày sản xuất | 4 | Numerial | Đen |  |
| Thời gian sử dụng dự kiến | 4 | Numerial | Đen |  |

# 5. Thiết kế chi tiết lớp

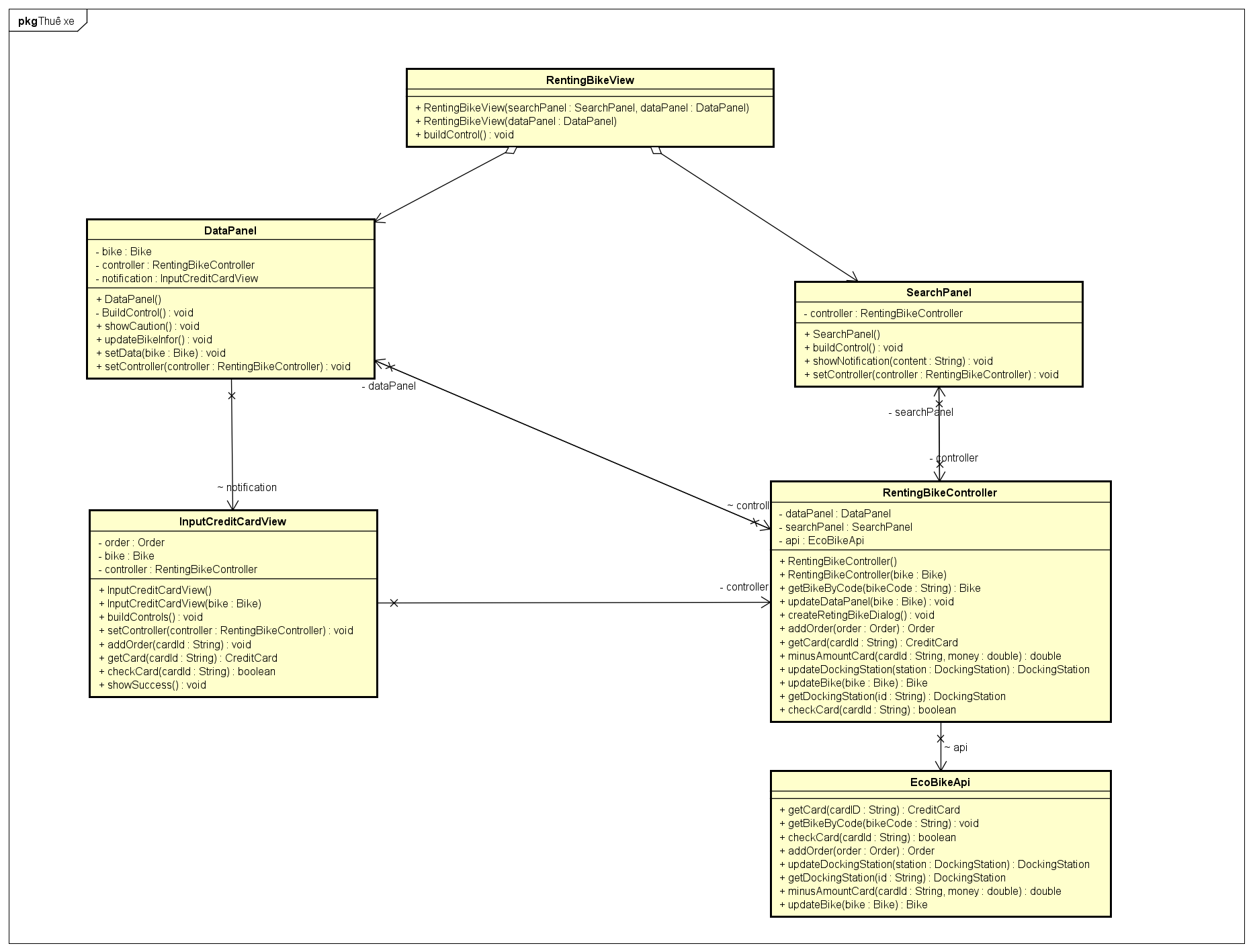
## 5.1: Biểu đồ trình tự use-case thuê xe

- Trong biểu đồ ở lớp thiết kế, lớp RentingBike sẽ được đổi tên thành RentingBikeView. Trong đó, riêng phần thanh tìm kiếm sẽ được tách ra thành lớp SearchPanel, phần các ô textfield hiển thị thông tin xe được tách thành DataPanel để đảm bảo lớp RentingBike không thực hiện quá nhiều công việc và đảm bảo nguyên lý đóng mở sẽ phân tích kĩ hơn ở phần 7.Chỉnh sửa thiết kế

- Đồng thời, trong biểu đồ thiết kế xuất hiện lớp: InputCreditCardView, đây là lớp phụ trách hiển thị form nhập mã thẻ và thông báo một số kết quả khi ấn thanh toán sau khi nhập mã thẻ



## 5.2: Biểu đồ chi tiết lớp cho use-case thuê xe



# 6. Lập trình và kiểm thử đơn vị.

Kiểm thử phương thức: **getBikeByCode(String bikeCode): Bike** trong class RentingBikeController. Khi người dùng muốn thuê xe, cần nhập mã xe vào ô tìm kiếm, khi ấn tìm kiếm sẽ hiện ra thông tin của xe. Phương thức được getBikeByCode sẽ được sử dụng trong tình huống này với:

* Input: mã xe cần thuê
* Output: thông tin xe đã nhập mã
* Nếu mã xe không tồn tại hoặc đã được thuê, cần thông báo cho người dùng

## 6.1: Kiểm thử hộp đen

Lựa chọn kĩ thuật kiểm thử: Bảng quyết định

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rule |  | R1 | R2 | R3 |
| Condition | Bike not null | F | T | T |
|  | Bike is renting | - | T | F |
| Action | output | null | null | Bike has the same bikeCode |

Đưa ra testcase:

TC1: input : “bikebike”

Expected result: null (do mã xe không tồn tại trên hệ thống)

TC2: input : “bike1”

Expected result: null (do xe đang được thuê)

TC1: input : “bike2”

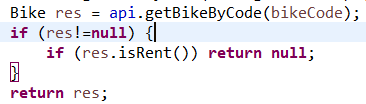
Expected result: Bike with bikeId is bike2 (xe hợp lệ)

- Các test case được cài đặt trong lớp: RentingBikeBlackBoxTest

## 6.2: Kiểm thử hộp trắng

Lựa chọn kĩ thuật kiểm thử: All path testing

Trong mã nguồn của hàm, có hai biểu thức điều kiện:



Trong đó, câu lệnh if thứ 2 nằm trong câu lệnh thứ 1, như vậy ta cần tối thiểu 3 test case để đi đủ các path của hàm

TC1: 1False

TC2: 1True-2True

TC3: 1True-2False

Từ đó ta đưa ra được cái test case tương ứng:

TC1: input : “bikebike”

Expected result: null (do mã xe không tồn tại trên hệ thống)

TC2: input : “bike1”

Expected result: null (do xe đang được thuê)

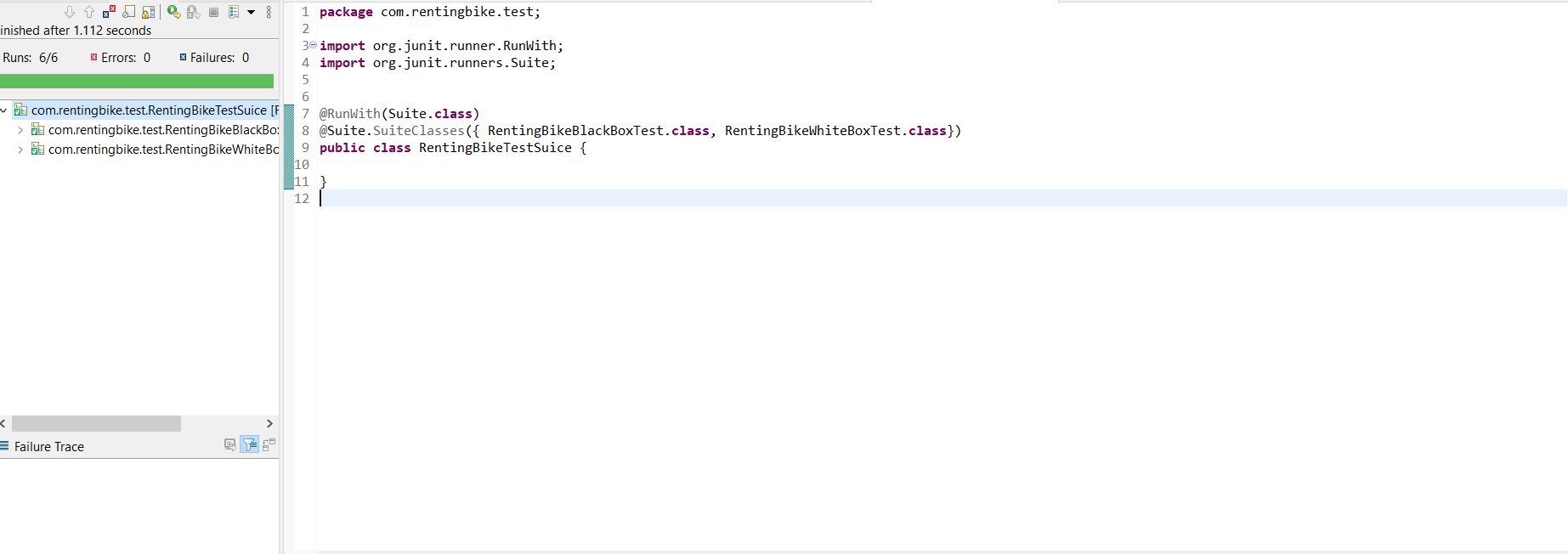
TC1: input : “bike2”

Expected result: Bike with bikeId is bike2 (xe hợp lệ)

* Các test case được cài đặt trong lớp: RentingBikeWhiteBoxTest

## 6.3: Kiểm thử tích hợp

- Các test case được cài đặt trong hai lớp **RentingBikeWhiteBoxTest** và **RentingBikeBlackBoxTest** được tích hợp vào lớp **RentingBikeTestSuice**. Chỉ cần khởi chạy lớp này để test với cả hai kiểu hộp trắng và hộp đen



# 7. Chỉnh sửa lại thiết kế

- Như đã phân tích ở phần ở phần 5.1, lớp RentingBike được tách thành hai lớp SearchPanel và DataPanel, đảm bảo “Nguyên lý một nhiệm vụ”. Lớp RentingBike cũ làm quá nhiều nhiệm vụ, khởi tạo thanh search và khởi tạo phần hiển thị thông tin của xe, đồng thời cũng phải xử lí các sự kiện khi nhập liệu vào ô tìm kiếm và hiển thị chúng lên các ô thông tin. Trong khi lớp này chỉ nên đóng vai trò hiển thị, nên ta tách thành hai lớp con đồng thời đổi tên thành RentingBikeView để thể hiện đúng vai trò của nó.

- Thêm phương thức **createRetingBikeDialog** vào lớp **RentingBikeController** với nhiệm vụ tạo ra giao diện RentingBikeView. Phần nhiệm vụ này vốn được trong hàm khởi tạo của lớp **RentingBikeController**, nhưng làm như vậy thì vi phạm “nguyên tắc đóng mở.