ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
****

BÁO CÁO CAPSTONE PROJECT

Giảng viên hướng dẫn: **Nguyễn Thị Thu Trang**

Sinh viên thực hiện: **Nguyễn Bá Thái - 20194666**
Đào Văn Thắng - 20194670
Lê Văn Toàn - 20194689
Trần Minh Trường - 20184211

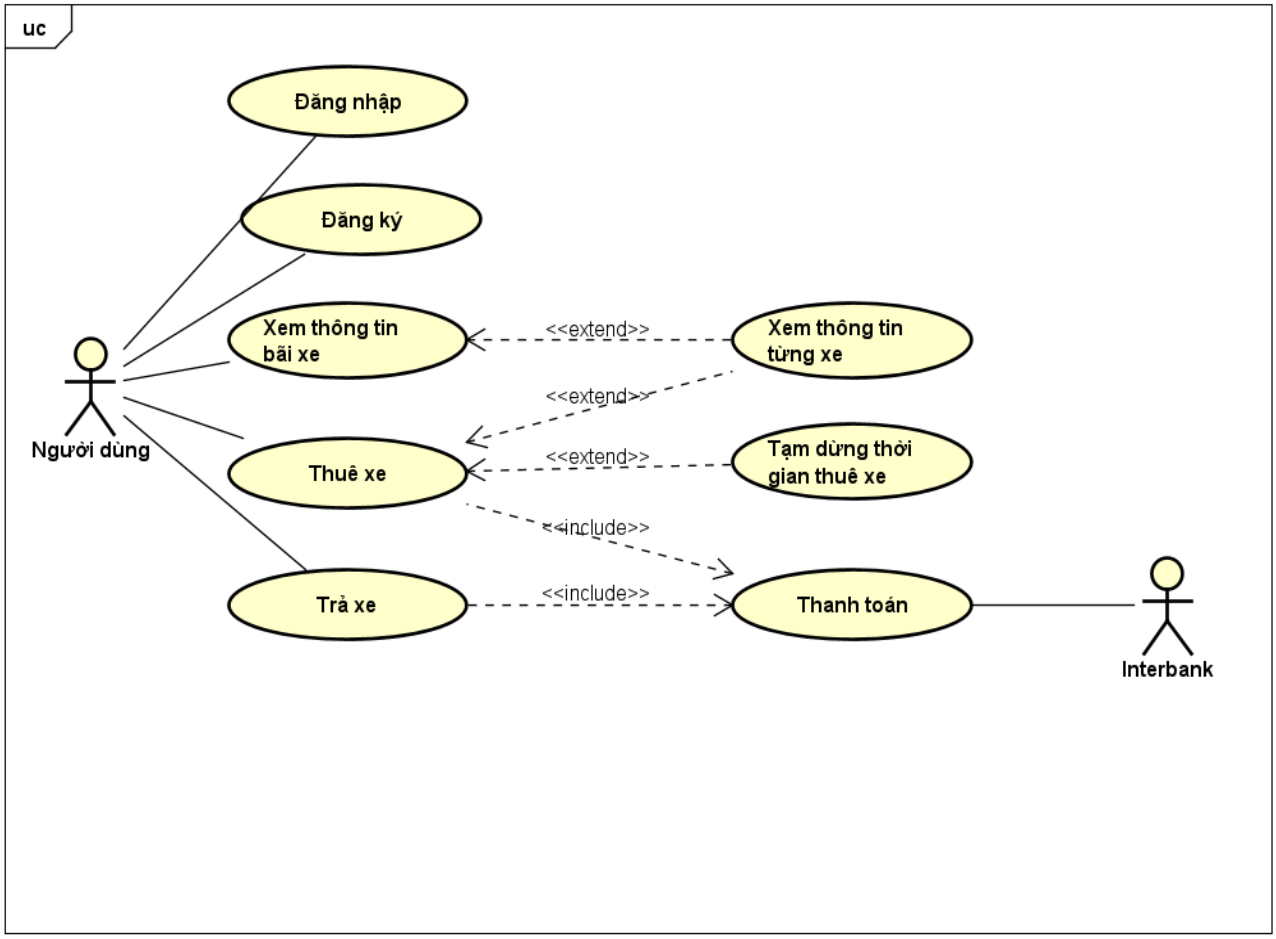
Báo cáo Capstone

Nhóm :12

Thành viên	MSSV	Công việc	Tỉ lệ đóng góp
Nguyễn Bá Thái	20194666	Phân tích, thiết kế phần “payment”	25%
Đào Văn Thắng	20194670	Phân tích, thiết kế phần “rentbike”	25%
Lê Văn Toàn	20194689	Phân tích, thiết kế phần “givebikeback”	30%
Trần Minh Trường	20184211	Phân tích, thiết kế phần “parkinginterface”	20%

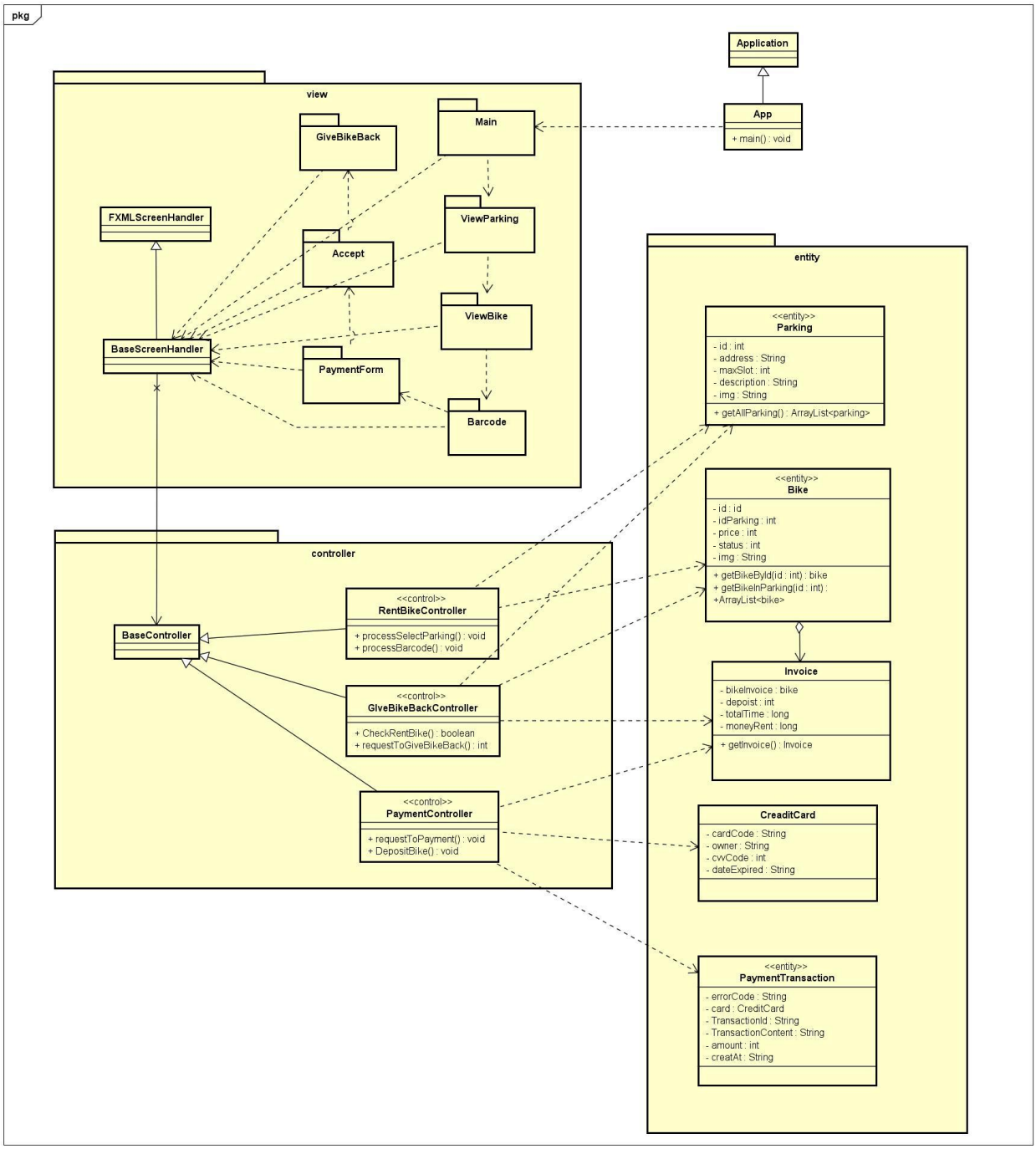
1. Phân tích yêu cầu

Biểu đồ use case

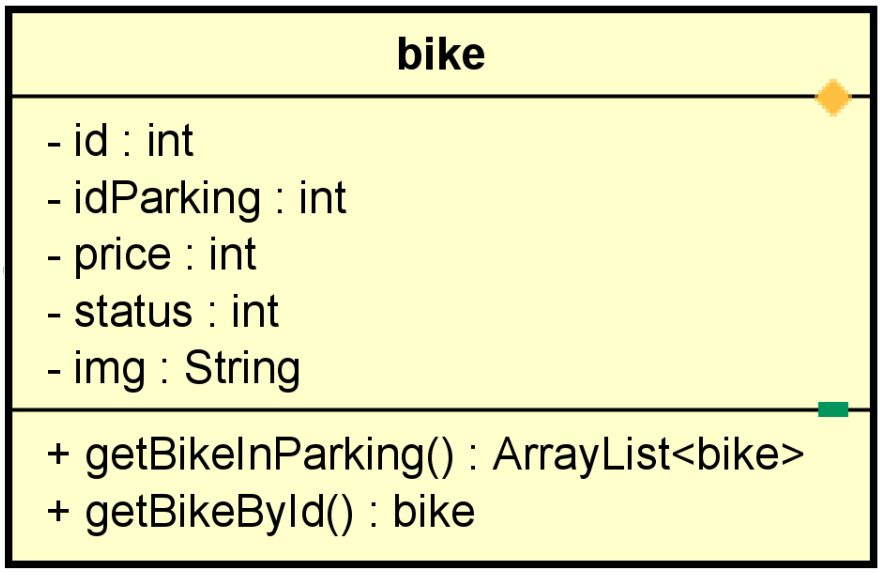


2. Thiết kế

2.1. Biểu đồ lớp tổng quan



2.2. Biểu đồ lớp chi tiết



Attribute

#	Name	Type	Default value	Description
1	id	int	NULL	Mã số xe
2	idParking	int	NULL	Mã số bãi đỗ xe
3	price	int	NULL	Giá tiền xe
4	status	Int	NULL	Trạng thái xe
5	Img	String	NULL	Hình ảnh của xe

Operation

none

Parameter

getBikeInParking:

int id: mã của bãi đỗ xe,

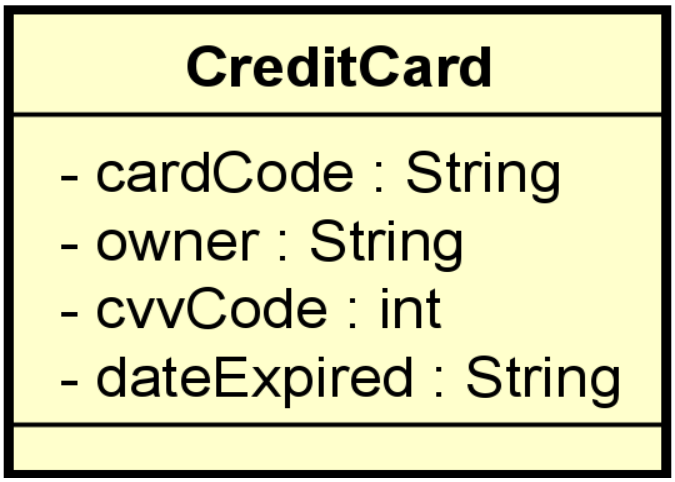
getBikeById:

int id: mã của xe

Exception

SQLException: Lỗi truy cập database

Method



Attribute

#	Name	Type	Default value	Description
1	cardCode	String	NULL	Mã thẻ thanh toán
2	owner	String	NULL	Người sở hữu thẻ
3	cvvCode	int	NULL	Mã an toàn thẻ
4	dateExpired	Int	NULL	Ngày hết hạn

Operation

none

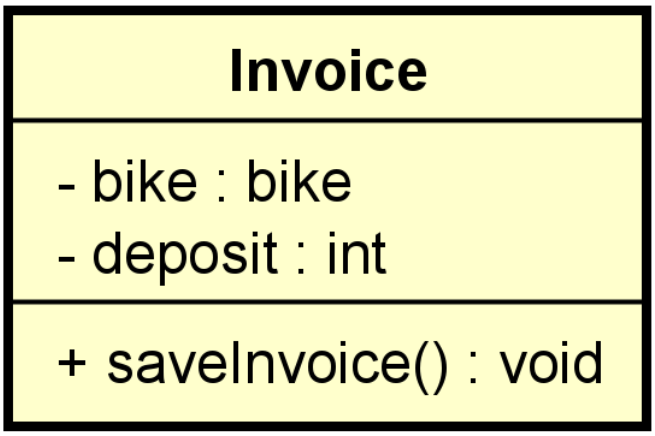
Parameter

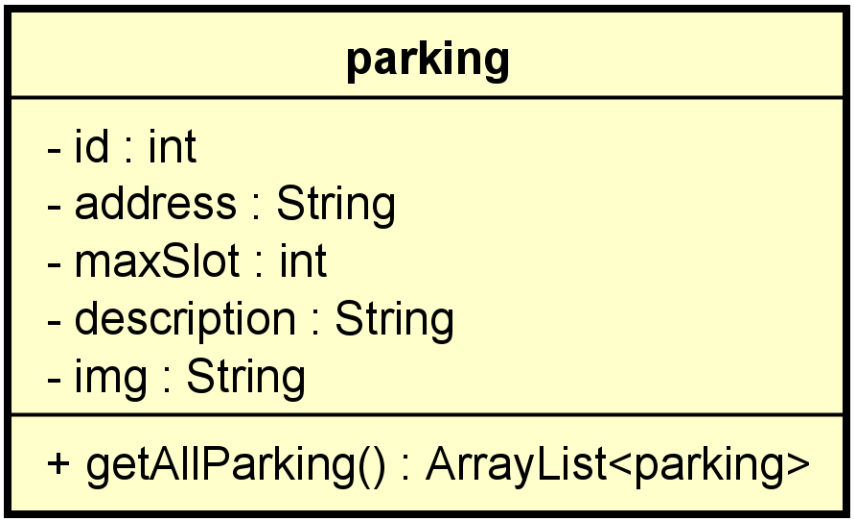
None

Exception

SQLException: Lỗi truy cập database

Method





Attribute

#	Name	Type	Default value	Description
1	id	int	NULL	Mã số bãi đỗ xe
2	address	String	NULL	Địa chỉ bãi đỗ xe
3	maxSlot	int	NULL	Số xe tối đa
4	description	String	NULL	Mô tả
5	img	String	NULL	Hình ảnh của bãi đỗ xe

Operation

none

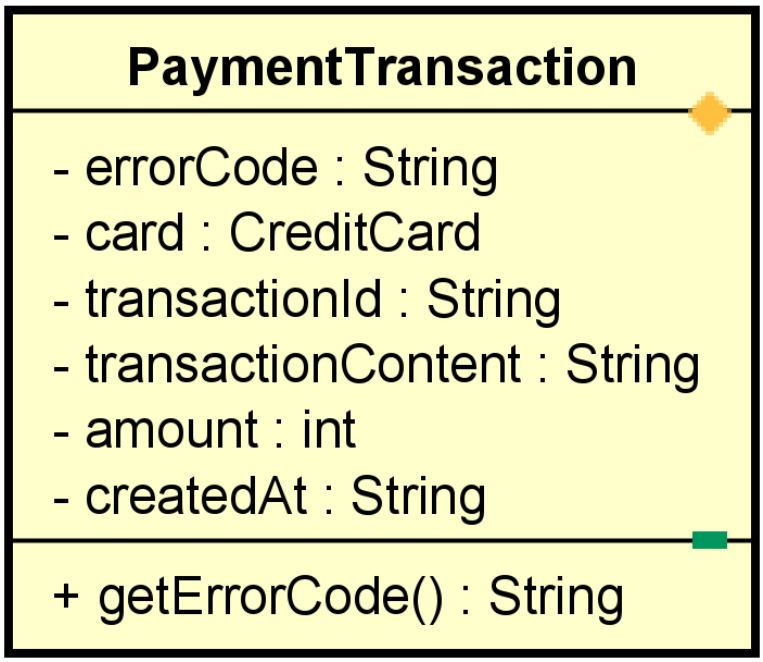
Parameter

none

Exception

SQLException: Lỗi truy cập database

Method



Attribute

#	Name	Type	Default value	Description
1	errorCode	String	NULL	Mã lỗi
2	card	CreditCard	NULL	Thẻ thanh toán
3	transactionId	String	NULL	Mã giao dich
4	transactionContent	String	NULL	Nội dung giao dịch
5	amount	int	NULL	Số tiền
6	createdAt	String	NULL	Thời gian tạo

Operation

none

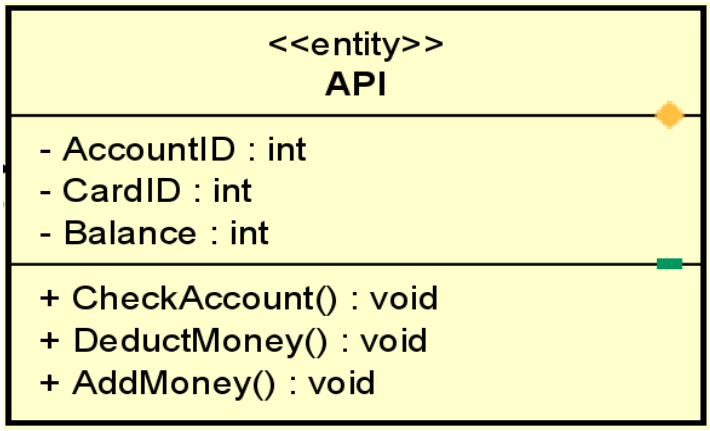
Parameter

none

Exception

SQLException: Lỗi truy cập database

Method



Attribute

#	Name	Date Type	Default value	Description
1	AccountID	Int	NULL	Mã tài khoản
2	CardID	int	NULL	Mã thẻ
3	Balance	Int	1000000	Số tiền trong thẻ

Operation

#	Name	Return Type	Description
1	CheckAccount	Boolean	Kiểm tra tài khoản
2	DeductMoney	Void	Trừ tiền trong tài khoản (để đặt cọc)
3	AddMoney	Void	Cộng tiền trong tài khoản (để trả xe)

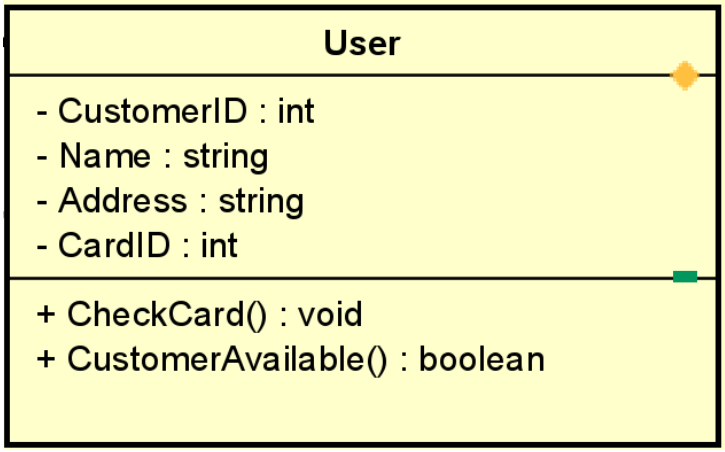
Parameter

none

Exception

SQLException: Lỗi truy cập database

Method



Attribute

#	Name	Date Type	Default value	Description
1	CustomerID	Int	NULL	Mã khách hàng
2	Name	String	NULL	Tên khách hàng
3	Address	String	NULL	Địa chỉ khách hàng
4	CardID	Int	NULL	Mã thẻ

Operation

#	Name	Return Type	Description
1	CheckCard	Boolean	Kiểm tra thẻ
2	CustomerAvailable	Boolean	Kiểm tra có tồn tại khách hàng không

Parameter

None

Exception

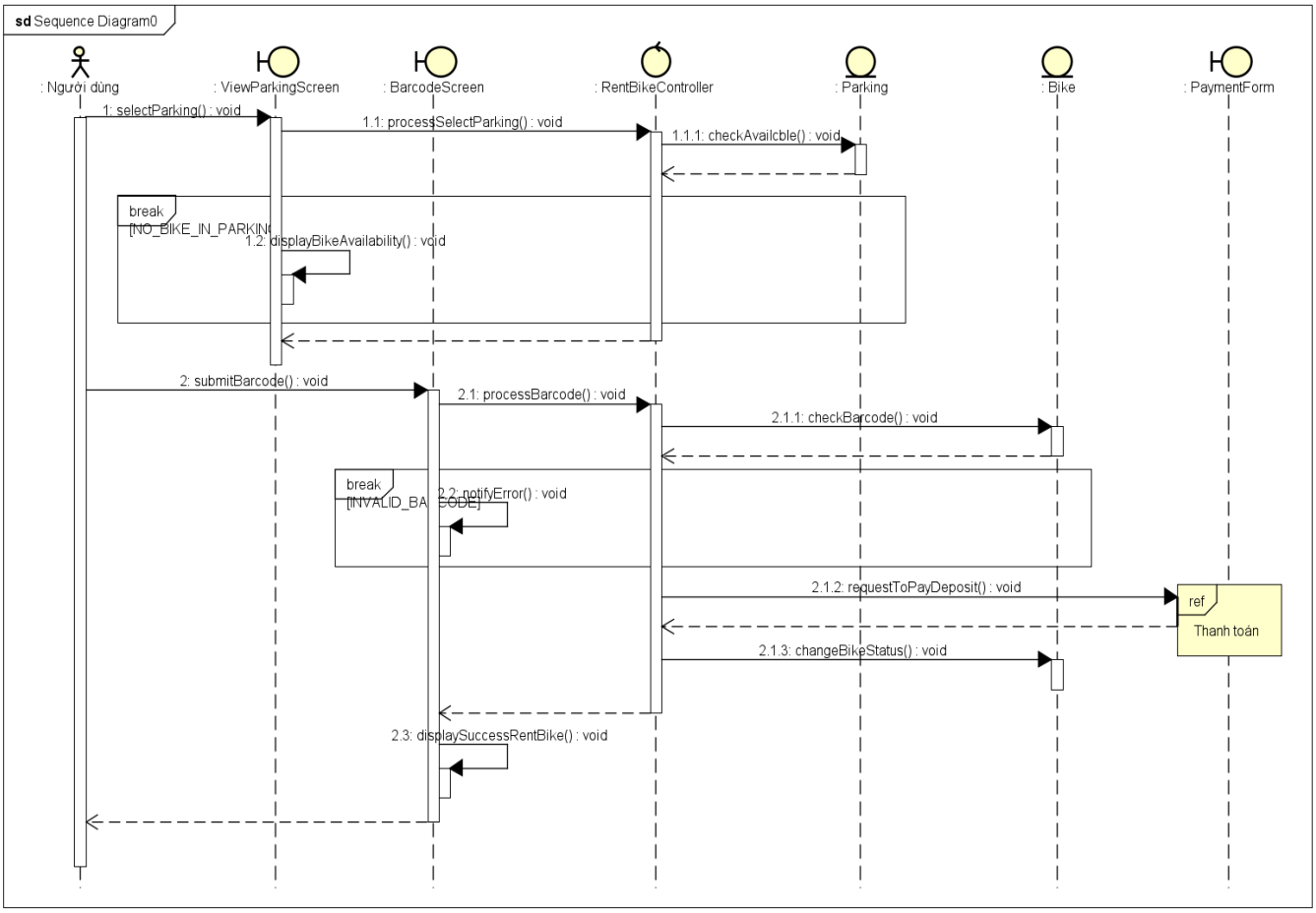
None

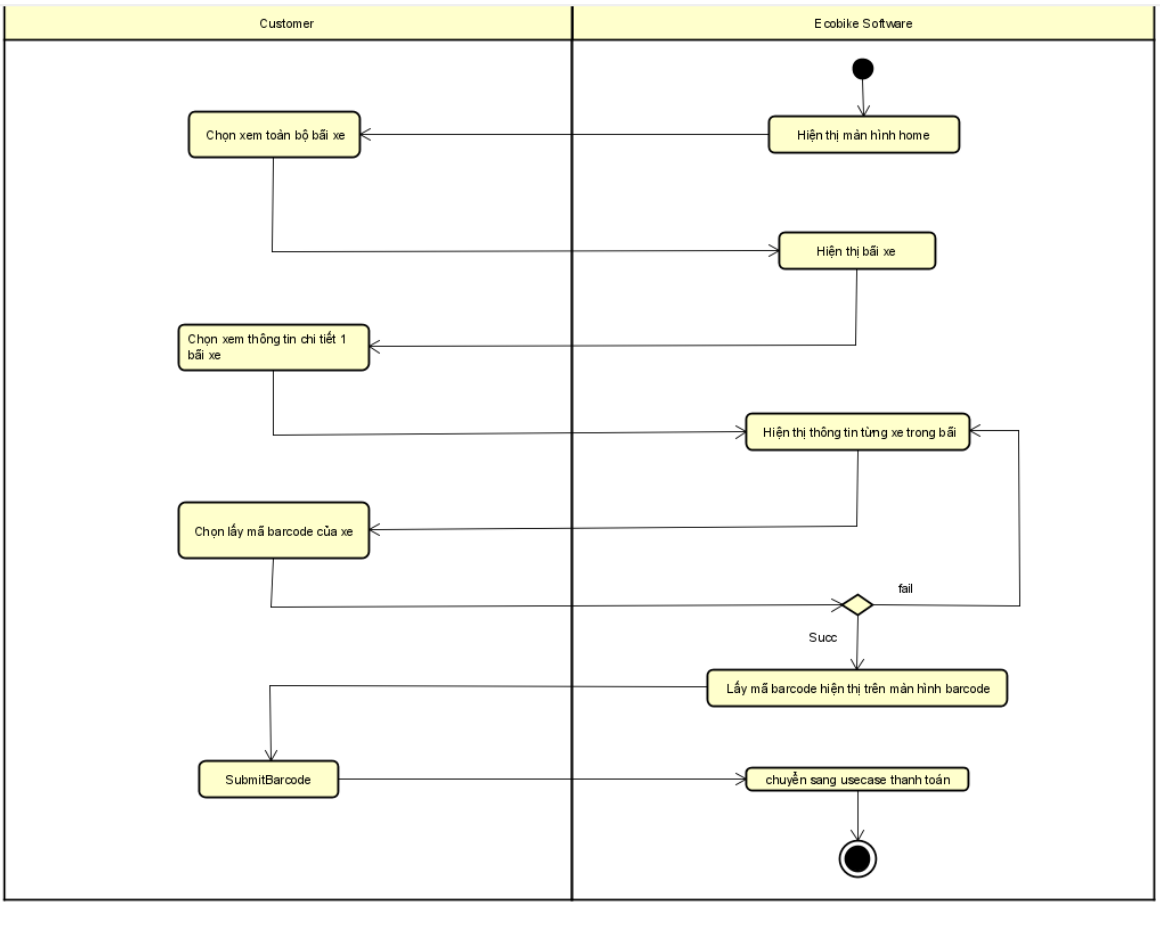
Method

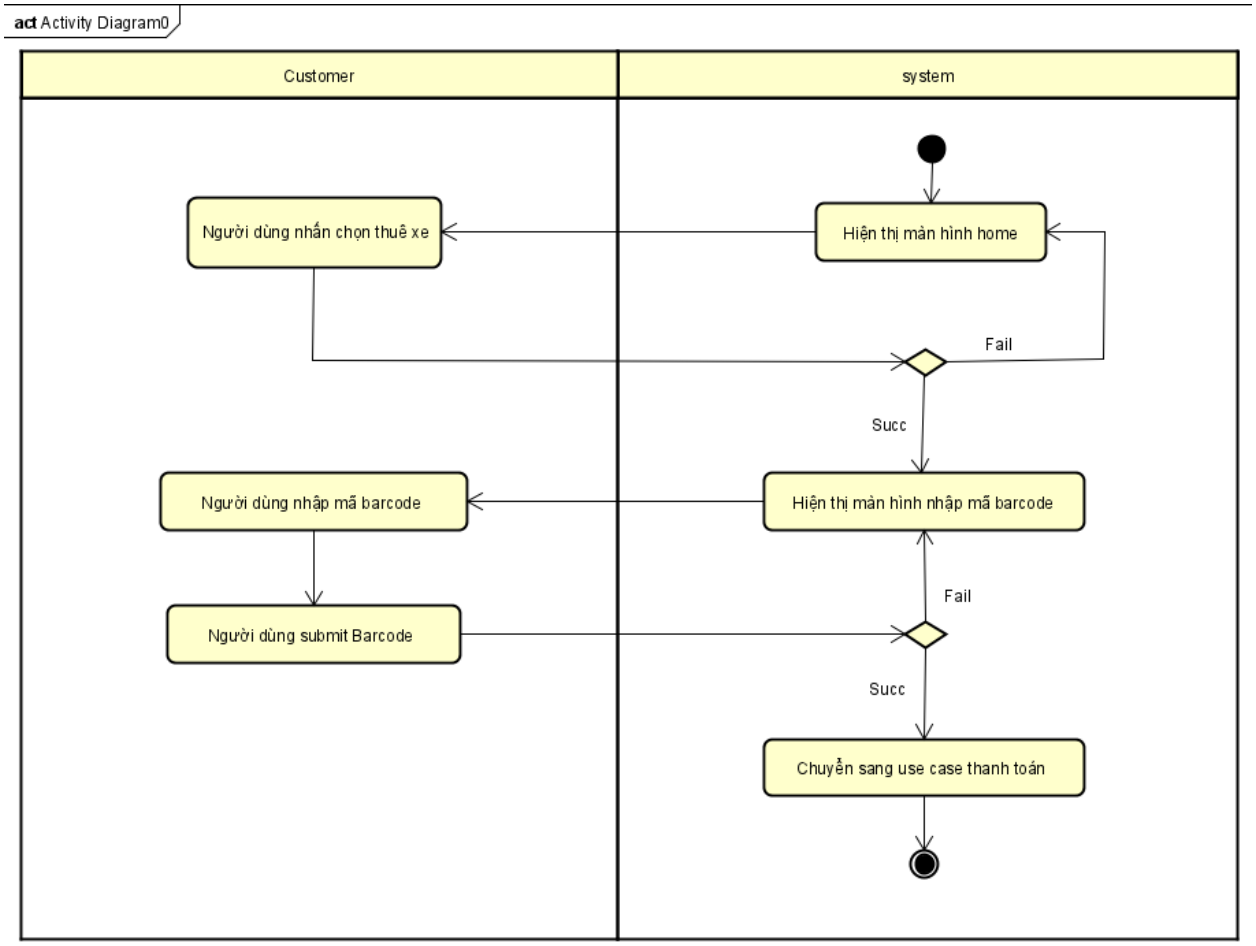
None

2.3. Biểu đồ tương tác phần quan trọng

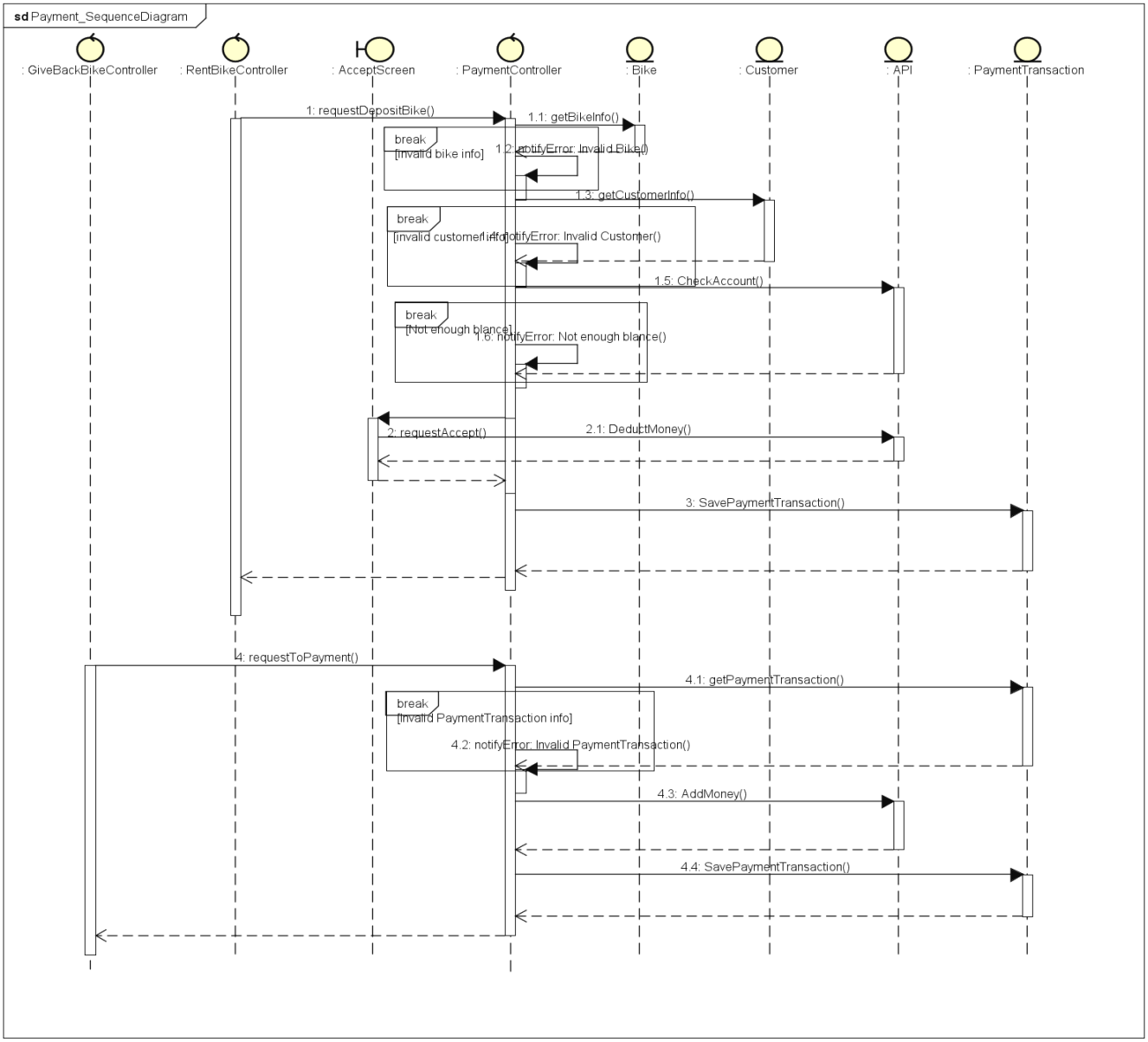
Thuê xe



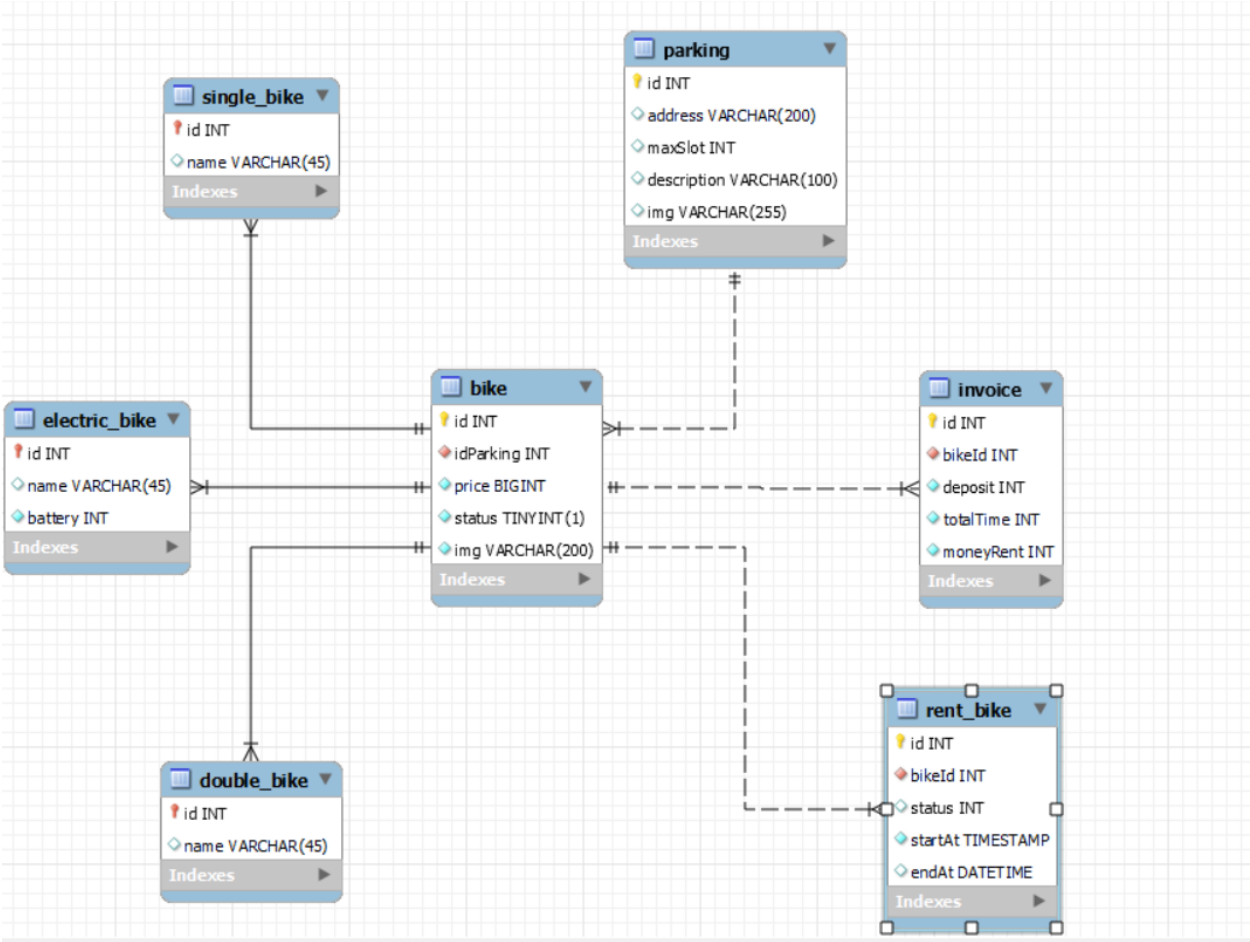

Customer
Ecobike Software
Chọn xem toàn bộ bãi xe
Hiện thị màn hình home
Hiện thị bãi xe
Chọn xem thông tin chi tiết 1 bãi xe
Hiện thị thông tin từng xe trong bãi
Chọn lấy mã barcode của xe
fail
Succ
Lấy mã barcode hiện thị trên màn hình barcode
SubmitBarcode
chuyển sang usecase thanh toán


act Activity Diagram0
Customer
system
Người dùng nhấn chọn thuê xe
Hiện thị màn hình home
Fail
Succ
Người dùng nhập mã barcode
Hiện thị màn hình nhập mã barcode
Fail
Người dùng submit Barcode
Succ
Chuyển sang use case thanh toán

Thanh toán



Thiết kế cơ sở dữ liệu:



3. Đánh giá thiết kế

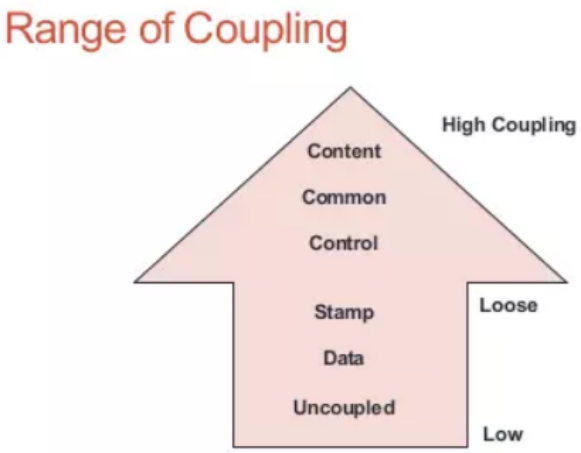
3.1 Coupling và cohesion ở mức nào tại sao :

Đầu tiên muốn tìm hiểu coupling và cohesion ở mức độ nào ta cần tìm hiểu coupling và cohesion là gì :

Trong thiết kế lập trình, mục tiêu của chúng ta là đạt được 2 vấn đề chung cốt lõi đó là loose coupling và high cohesion. Những thứ mà có lẽ chúng ta đều cảm thấy quen thuộc vì đã được dạy ở trường đại học. Coupling nói một cách đơn giản nó là sự phụ thuộc lẫn nhau giữa các modules (có thể hiểu là class, thành phần của phần mềm), là tính liên kết giữa các modules. Thiết kế code tốt thì phải theo hướng loose coupling tức là sự liên kết giữa các class, các thành phần trong phần mềm càng ít chặt chẽ càng tốt. Bởi vì nếu các thành phần quá phụ thuộc lẫn nhau, khi chúng ta thay đổi 1 thành phần sẽ ảnh hưởng tới rất nhiều thành phần liên quan khác.

Cohesion cũng là chỉ mức độ về sự liên kết như coupling nhưng nó là sự liên kết ở mức chi tiết. Sự liên kết giữa các thành phần bên trong 1 module càng chặt chẽ thì thiết kế đó càng tốt tức là high cohesion. Lý do cho high cohesion là tất cả các yếu tổ liên quan đến nhau được đặt trong 1 module để hướng tới 1 nhiệm vụ nhất định. Ví dụ như trong class Invoice chúng ta không lưu trường customer_name trường này được lưu trong class Customer hay như class Payment chúng ta không cần đặt vào đó function tính toán số tiền mà nhiệm vụ đó nằm ở class Invoice (tạo hoá đơn).

3.1.1: coupling :



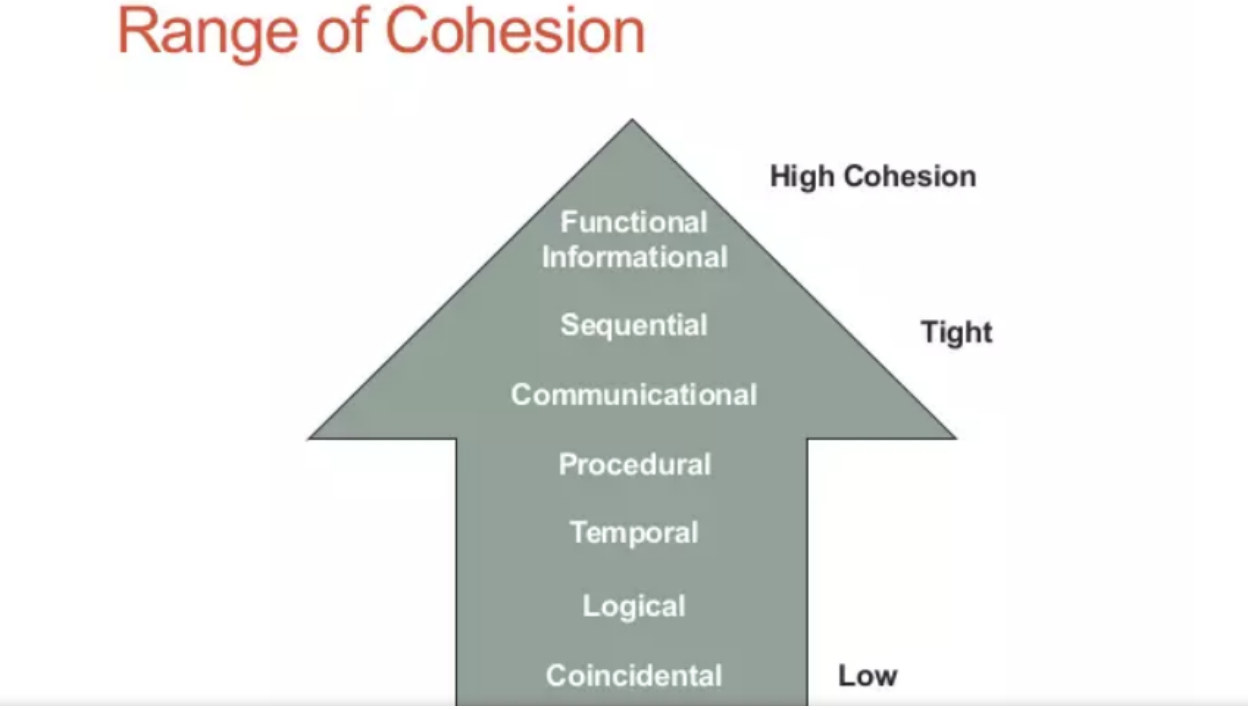
-Nhìn từ bảng trên thì ta có thể thấy được là phần mềm của nhóm ở mức high coupling khi vẫn mắc những lỗi sau :

Common coupling: Ví dụ trong module thuê xe và module trả xe thì đều chỉnh sửa dữ liệu trong bảng rent_bike của database

Data coupling: Các module tương tác với nhau thông qua tham số truyền vào. Vd : module view bike tương tác với module thuê xe thông qua tham số đầu ra của module view bike

-> thiết kế vẫn còn ở mức high coupling

3.1.2 Cohesion:



Nhìn vào bảng trên thì ta có thể thấy là cohesion của phần mềm đạt ở mức tight cohesion bởi vì :

sequential cohesion: Khi output của một element trở thành input của một element khác

4. Xây dựng phần mềm:

4.1Công nghệ :

Ngôn ngữ lập trình: Java

Hệ quản trị cơ sở dữ liệu:Mysql