



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

**VIỆN CÔNG NGHỆ THÔNG
TIN VÀ TRUYỀN THÔNG**

BÁO CÁO BÀI TẬP LỚN

XÂY DỰNG HỆ THỐNG THUÊ XE

CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

SINH VIÊN THỰC HIỆN

Họ Tên	MSSV
Đỗ Viết Trí	20173412
Trần Văn Trí	20173410
Nguyễn Ngọc Trinh	20173413
Nguyễn Mạnh Trường	20177022

Hà Nội, tháng 12/2020

Nội dung

- I. Phân công công việc
- II. Một interaction điển hình
- III. Biểu đồ lớp thiết kế
- IV. Design Consideration

PHÂN CÔNG CÔNG VIỆC

I: Phân công nhiệm vụ và đánh giá phần trăm đóng góp

1. Phân công nhiệm vụ

- + Đỗ Viết Trí: Phân tích thiết kế, code và test cho usecase trả xe, tổng hợp readme
- + Trần Văn Trí: Phân tích thiết kế, code và test cho usecase thuê xe, Tìm kiếm xe
- + Nguyễn Ngọc Trinh: Phân tích thiết kế, code và test cho usecase xem thông tin bãi xe
- + Nguyễn Mạnh Trường: Phân tích thiết kế, code và test cho usecase xem thông tin chi tiết xe

PHÂN CÔNG CÔNG VIỆC

I: Phân công nhiệm vụ và đánh giá phần trăm đóng góp

1. Đánh giá đóng góp

+ Nhóm không có nhân vật nổi trội, tuy nhiên làm việc rất tích cực

+ Đánh giá đóng góp:

+ Đỗ Viết Trí : 30%

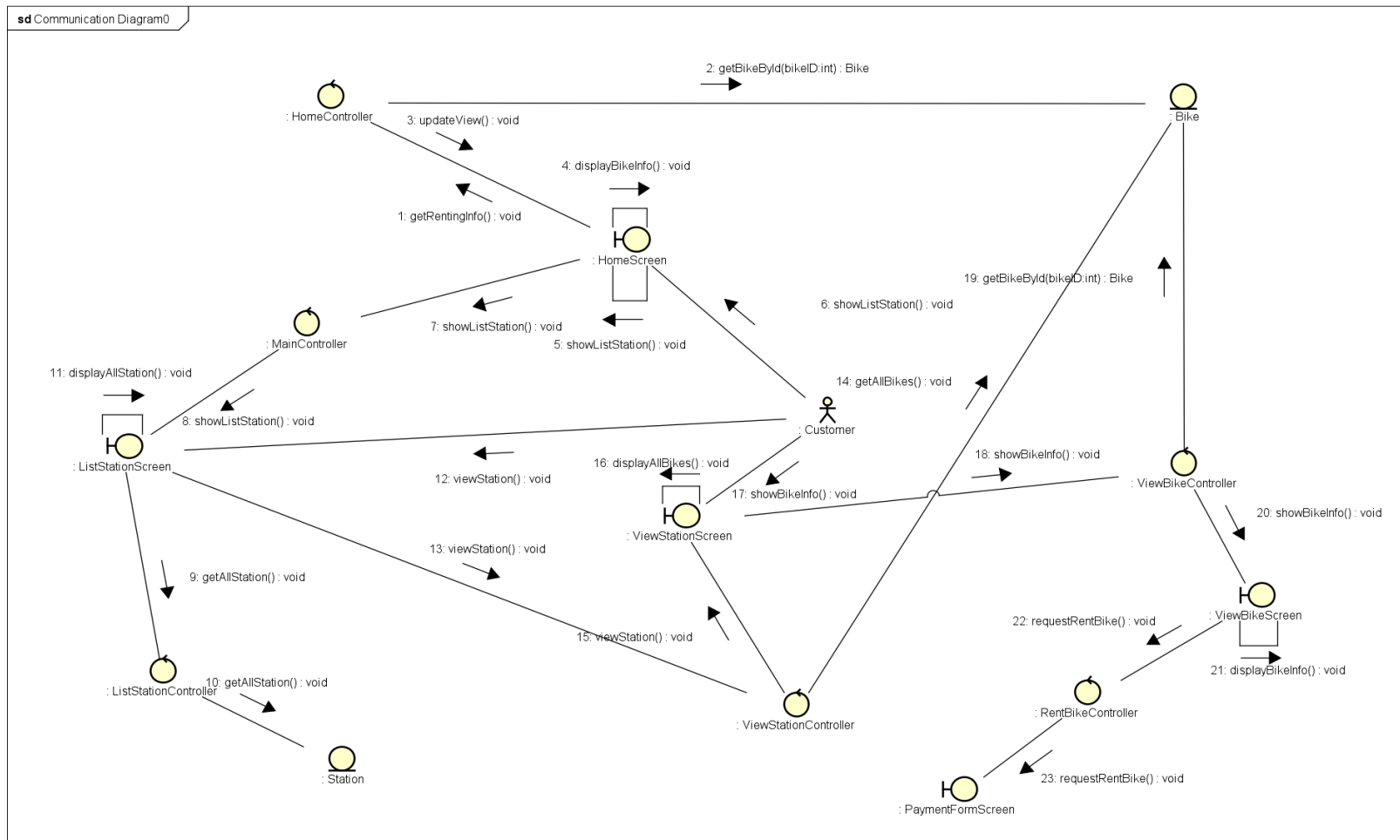
+ Trần Văn Trí : 30%

+ Nguyễn Ngọc Trinh: 20%

+ Nguyễn Mạnh Trường : 20%

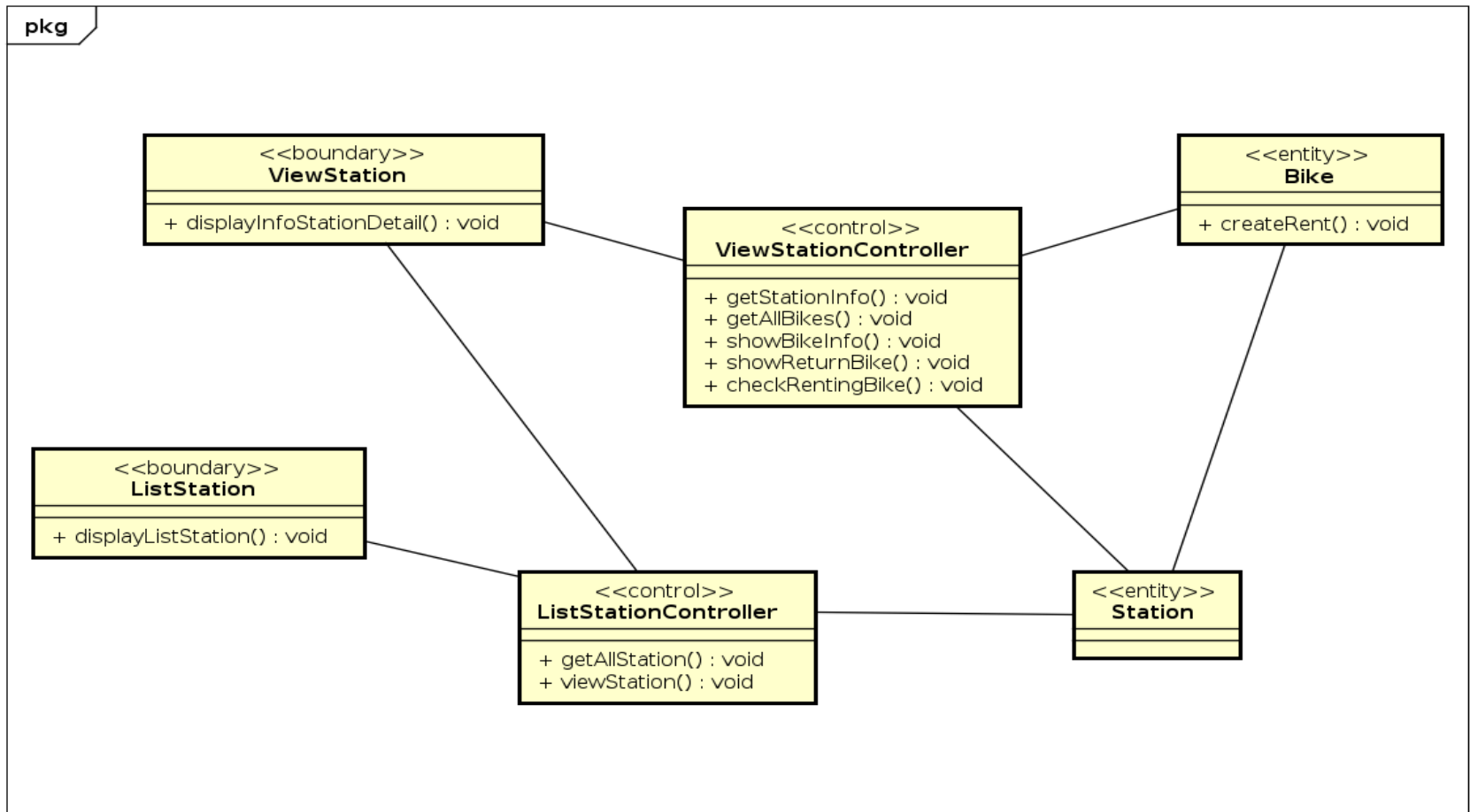
Một interaction diễn hình

II: Biểu đồ tương tác xem thông tin chi tiết xe



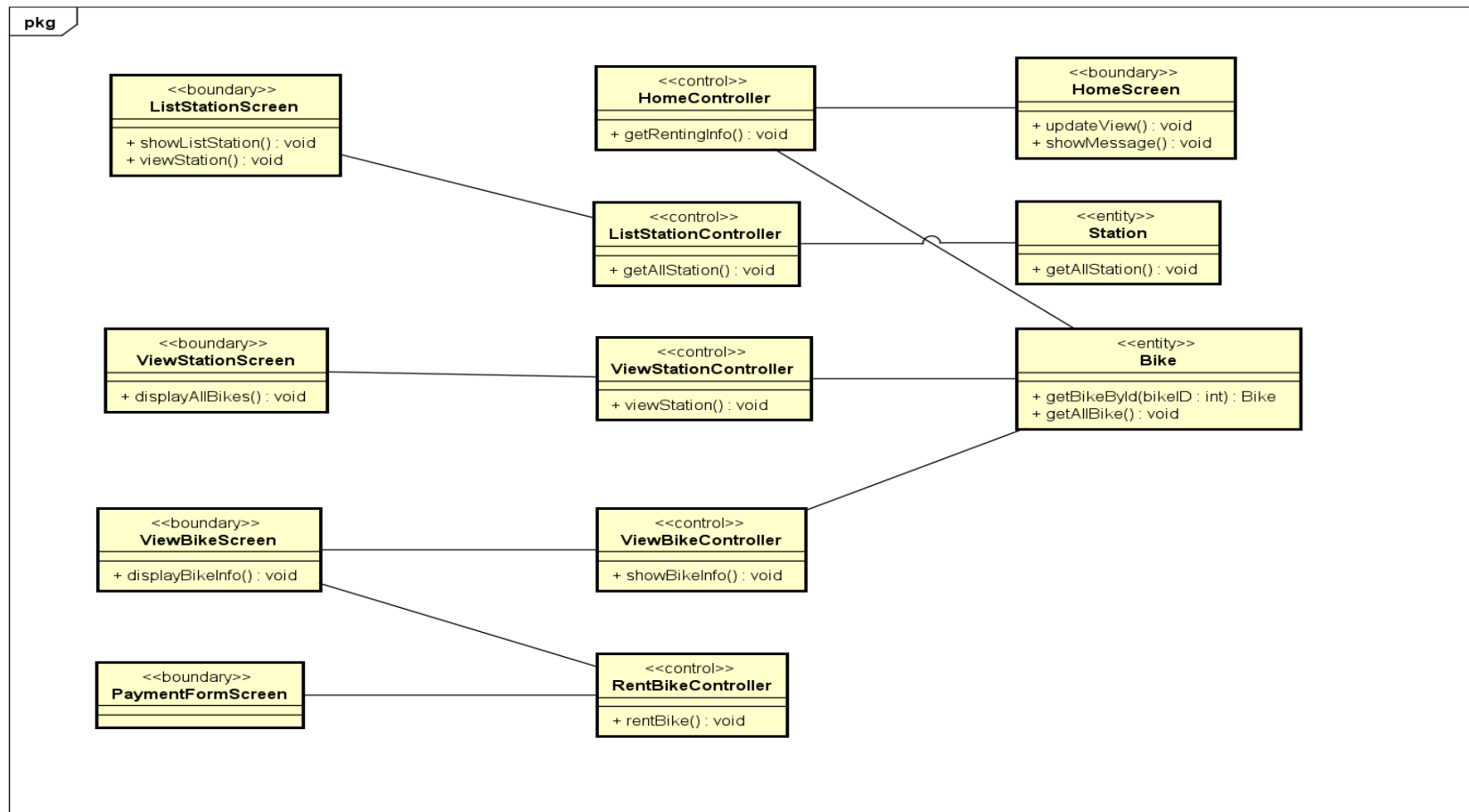
Biểu đồ lớp thiết kế

III.2.1. Biểu đồ usecase View Bike In Station



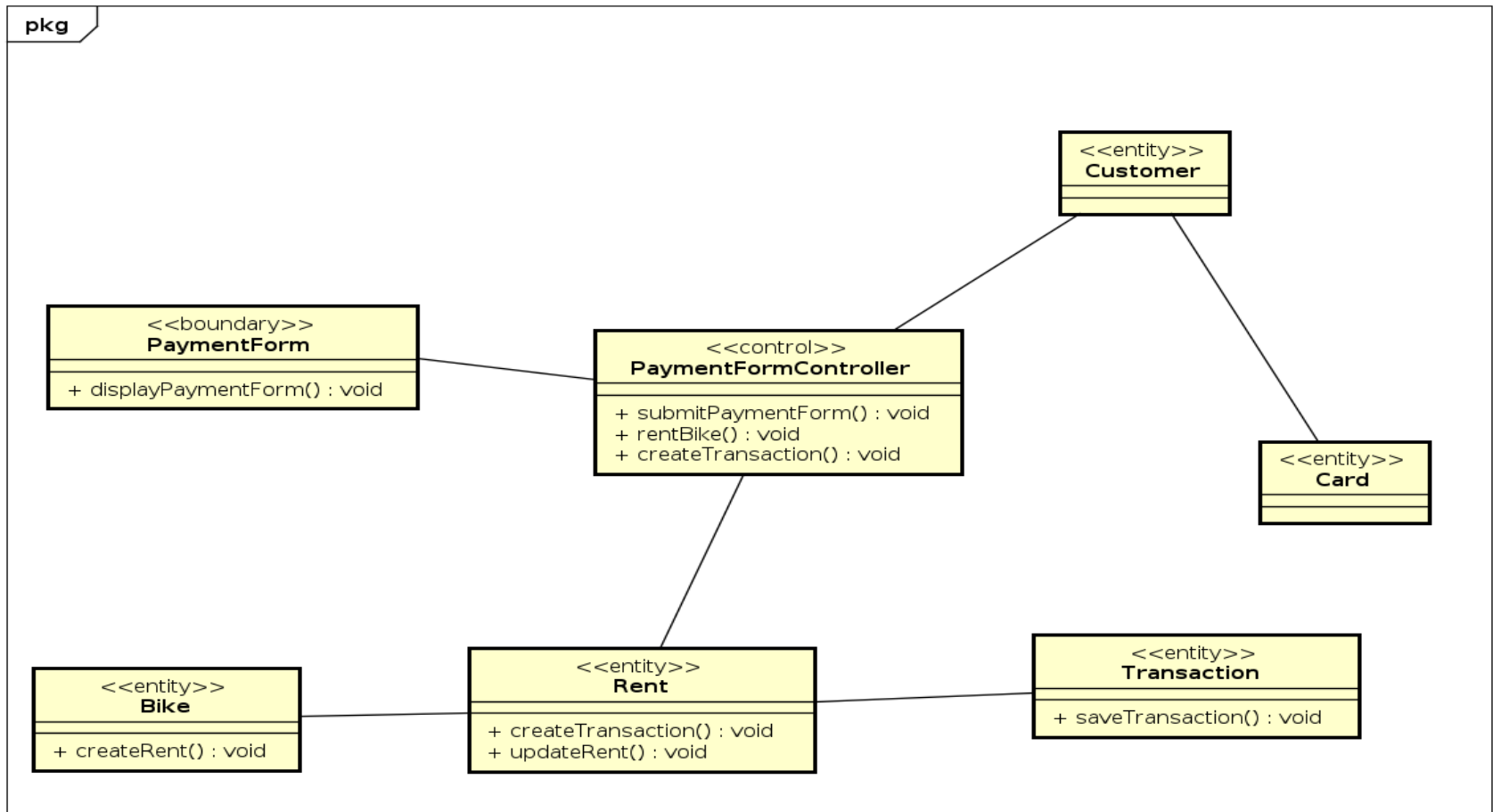
Biểu đồ lớp thiết kế

III.2.2. Biểu đồ usecase View Bike Information



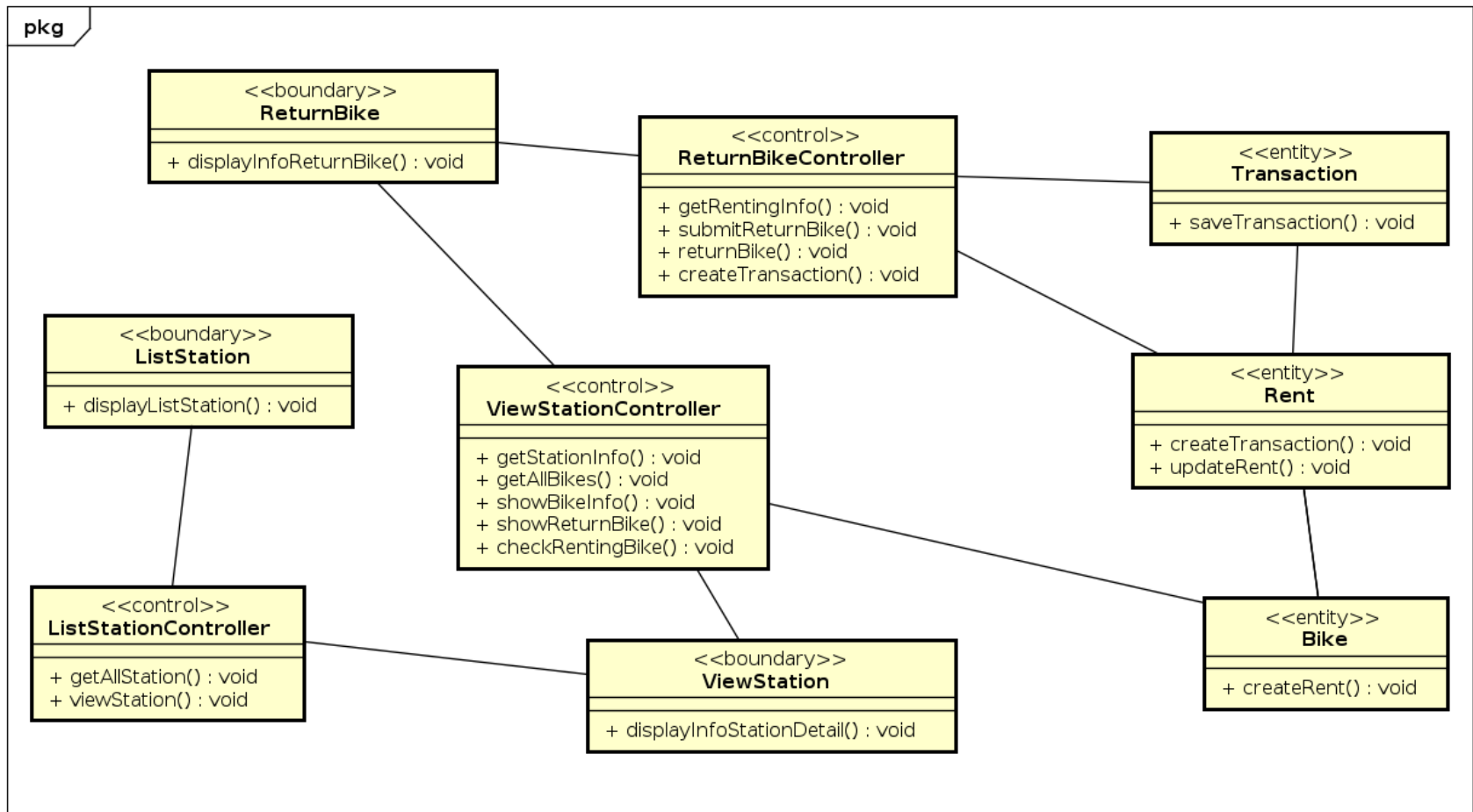
Biểu đồ lớp thiết kế

III.2.3. Biểu đồ usecase Rent Bike



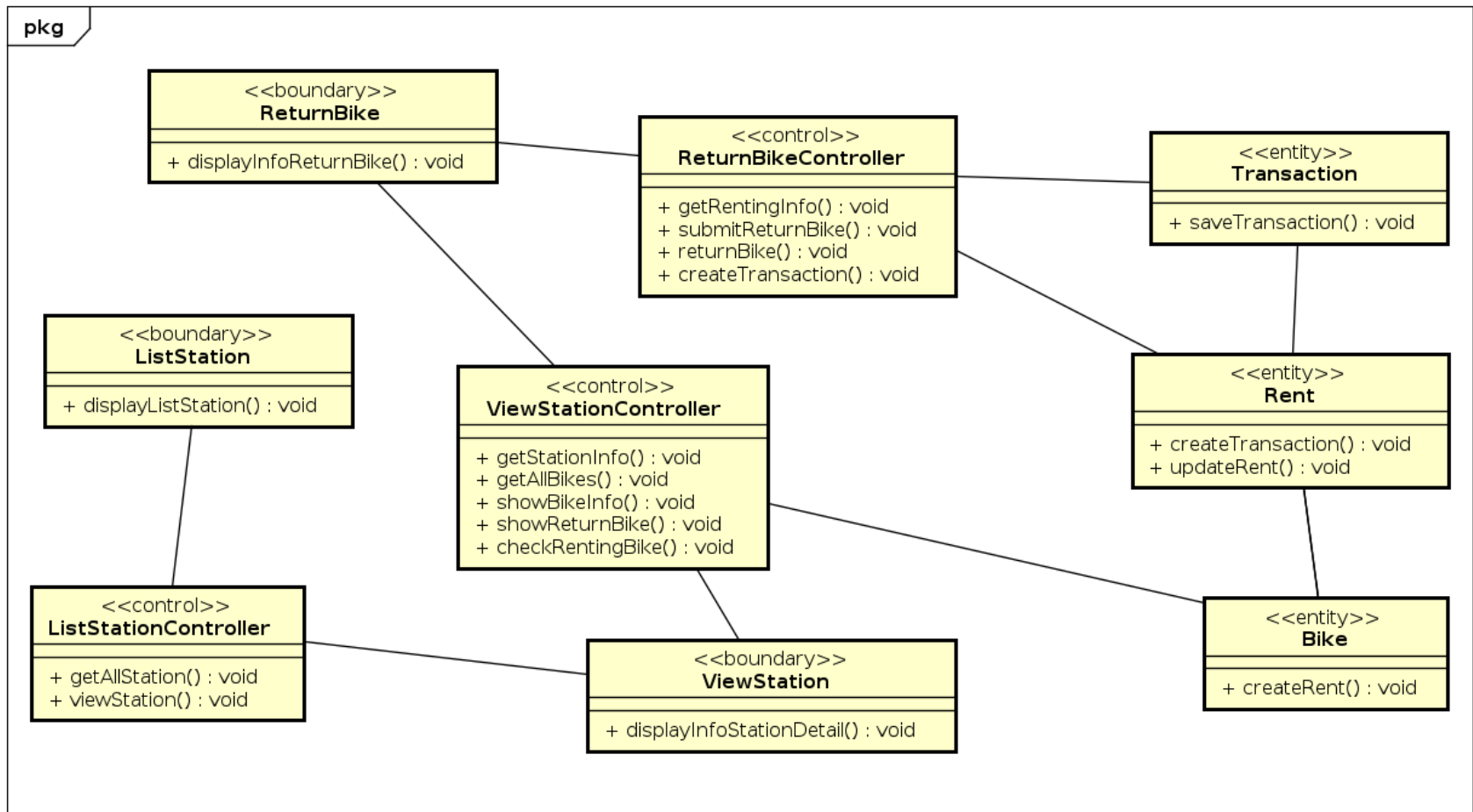
Biểu đồ lớp thiết kế

III.2.3. Biểu đồ usecase Return Bike



Design Consideration

III.2.4. Biểu đồ usecase Return Bike



Design Consideration

IV.1. Design Concepts

- Một số Common Coupling trong hệ thống:
- + Trong lớp ViewStationController, phương thức searchBike và phương thức addEvent đều sử dụng chung thuộc tính **bikeIDSelected**

```
/**
 * Tìm kiếm xe theo mã id xe
 */
public void searchBike() {
    if (txtBikeCode.getText().isEmpty()) return ;
    else {
        int id = Integer.parseInt(txtBikeCode.getText());
        int bikeID = station.getBikeByID(id).id;
        if (bikeID != 0) {
            Contants.bikeIDSelected = bikeID;
            showBikeInfo();
        }
        else {
            showMessage("Không tìm thấy xe !");
        }
    }
}
```

```
public void addEvents() {

    btnViewBike.setOnClickListener(e -> {
        if(tbvListBike.getSelectionModel().getSelectedItem() != null) {
            Contants.bikeIDSelected = tbvListBike.getSelectionModel().getSelectedItem().id;
            showBikeInfo();
        }
    });

    btnReturnBike.setOnClickListener(e -> {
        checkBikeRenting();
    });

    btnRefresh.setOnClickListener(e -> {
        getAllBikes();
    });

    btnSearch.setOnClickListener(e -> {
        searchBike();
    });

    tbvListBike.setRowFactory(tv -> {

        TableRow<Bike> row = new TableRow<>();
        row.setOnClickListener(e -> {
            if (e.getClickCount() == 2 && !(row.isEmpty())) {

                Contants.bikeIDSelected = row.getItem().id;
                showBikeInfo();
            }
        });
    });
}
```

Design Consideration

IV.1. Design Concepts

- Một số Common Coupling trong hệ thống:
- + Trong lớp ListStationController, phương thức searchBike và phương thức addEvent đều sử dụng chung thuộc tính **currentRentID**

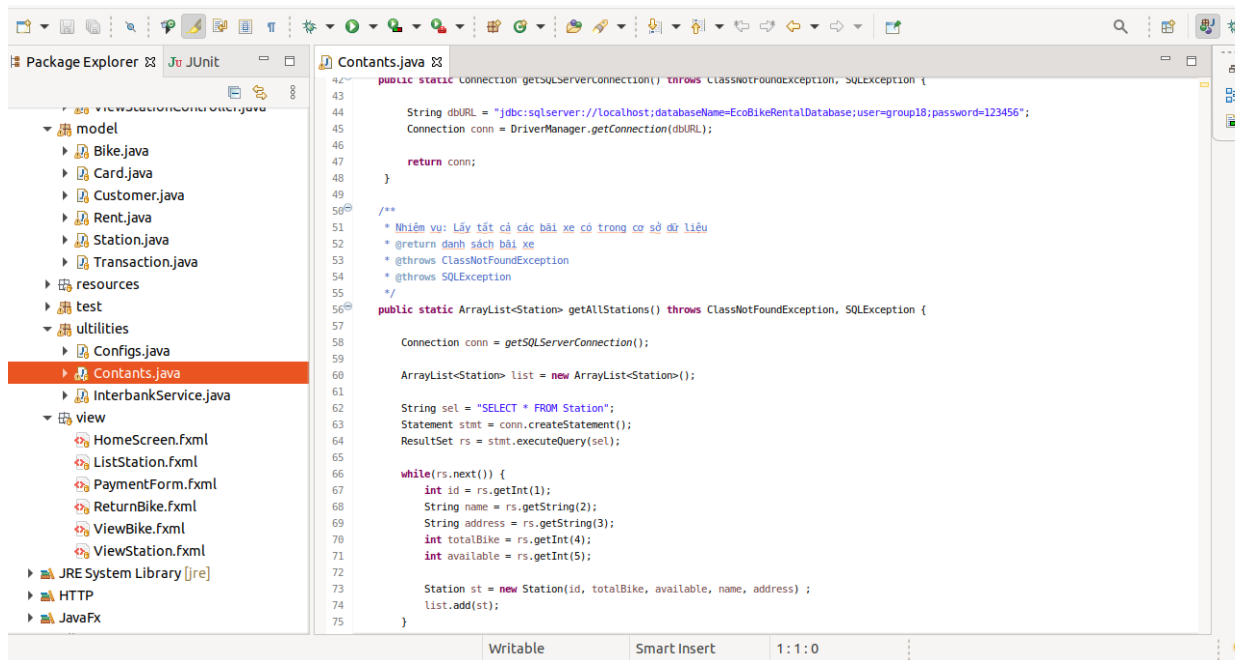
```
/**
 * Lấy thông tin thuê xe nếu người đó đang thuê xe
 */
public void getRentingInfo() {
    rent.setRentFromID(Contants.currentRentID);
    bike.setBikeFromID(rent.bikeID);
    card.setCard(HomeController.currentUser.getCustomerCard());
    // update view
    updateView();
}
```

```
/**
 * Nhiệm vụ: xử lý trả xe, gọi đến api trả xe
 * @param code: lệnh giao dịch: "pay" hay "refund"
 * @param totalMoney: tổng số tiền giao dịch
 */
public void returnBike( String code, int totalMoney) {
    String res;
    try {
        res = InterbankService.processTransaction(card, code, totalMoney);
        // System.out.print("Code: " + res);
        if("00".equals(res)) {
            bike.updateBike("available", Contants.stationIDSelected);
            rent.updateRent(current, totalTimeRent);
            Rent rent = new Rent(Contants.currentRentID);
            rent.createTransaction("return", "Return bike" , totalMoney);
            updateRentingBike();
            showMessage(Configs.MESSAGE_SUCCESS);
            Stage stage = (Stage) btnClose.getScene().getWindow();
            stage.close();
        } else {
            showMessage(Contants.response(res));
        }
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}
```

Design Consideration

IV.1. Design Concepts

- Hệ thống vi phạm Logical cohesion: cùng truy cập vào database thì nên đặt vào một package riêng nhưng trong hệ thống, package model và package utilities đều có lớp truy cập vào database
- Lớp Contants trong package utilities

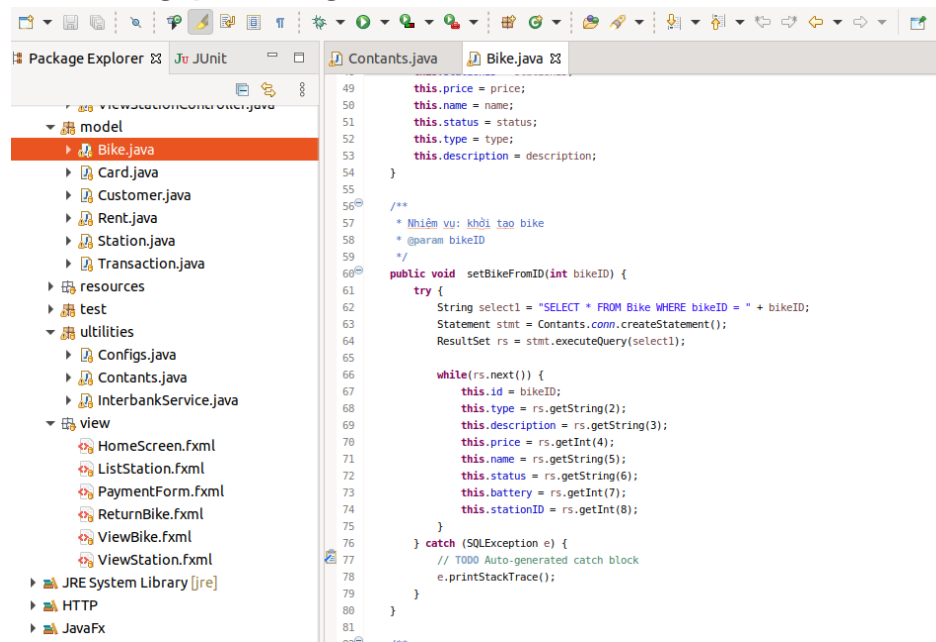


Design Consideration

IV.1. Design Concepts

-Hệ thống vi phạm Logical cohesion: cùng truy cập vào database thì nên đặt vào một package riêng nhưng trong hệ thống package model và package utilities đều có lớp có thể truy cập vào database

-Ví dụ 1 lớp Bike trong package model



Design Consideration

IV.1. Design Concepts

- Ví dụ về việc sử dụng Sequential Cohesion: đầu ra của phương thức này là thành một thành phần trong phương thức khác

- Ví dụ: Trong lớp PaymentFormController, phương thức checkBlankField() là đầu vào cơ sở của phương thức submitPaymentForm() kiểm tra xem các trường nhập vào có hợp lệ hay không

```
/**
 * Nhiệm vụ: Kiểm tra xem tất cả các trường trong form đã được nhập chưa
 * @return true hoặc false
 */
public boolean checkBlankField() {
    if (txtCardHolderName.getText().isEmpty()) return false;
    if (txtCardNumber.getText().isEmpty()) return false;
    if (txtCardBank.getText().isEmpty()) return false;
    if (txtCardExpirationDate.getText().isEmpty()) return false;
    if (txtCardCVV.getText().isEmpty()) return false;
    return true;
}
```

```
/**
 * Nhiệm vụ: xử lý khi submit form
 */
public void submitPaymentForm() {
    if (checkBlankField()) {
        card.cardHolderName = txtCardHolderName.getText();
        card.cardNumber = txtCardNumber.getText();
        card.expirationDate = txtCardExpirationDate.getText();
        card.securityCode = txtCardCVV.getText();
        card.issuingBank = txtCardBank.getText();
        current = Contants.getCurrentTime();
        rentBike(bike.getDepositMoney());
    } else {
        showMessage("Hãy nhập đầy đủ thông tin !");
    }
}
```




TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Thank you !

