

Hệ thống thuê xe điện tử

Môn thiết kế và xây dựng phần mềm

Nhóm 12

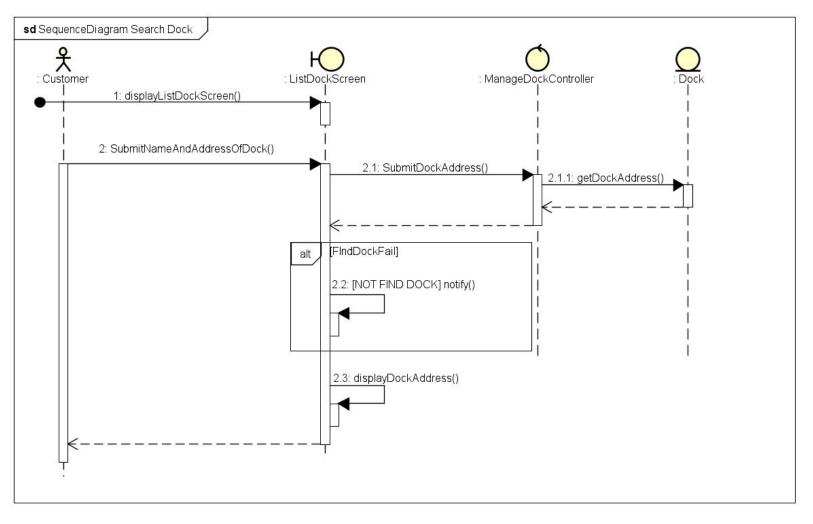
Giáo viên hướng dẫn: TS. Nguyễn Thị Thu Trang

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

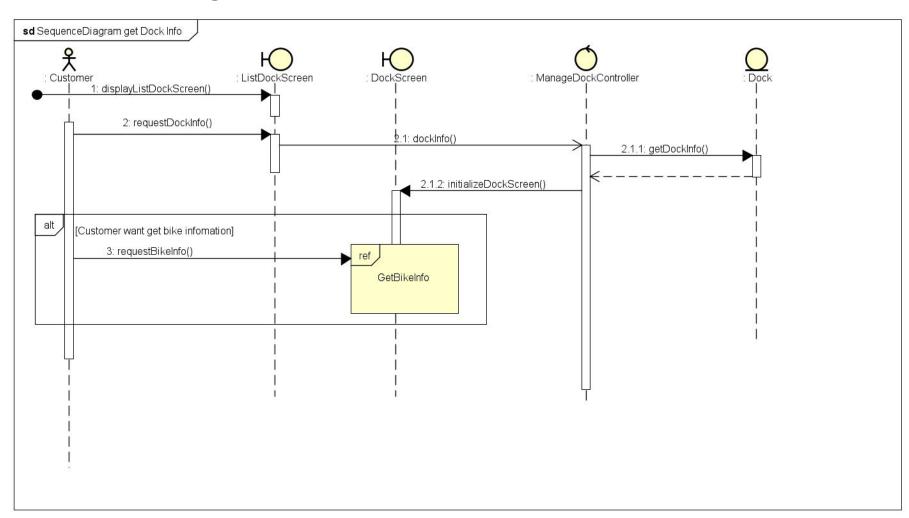
Nội dung

- 1. Biểu đồ tương tác use case.
- 2. Biểu đồ lớp.
- 3. Mô tả source code.
- 4. Coupling/Cohesion.
- 5. Design principles.
- 6. Design Pattern.
- 7. Phân công công việc.

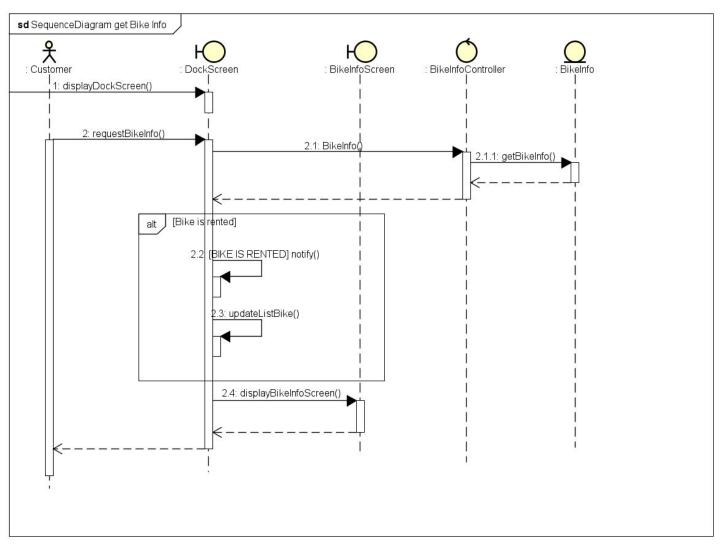
UC001: Tìm kiếm bãi xe



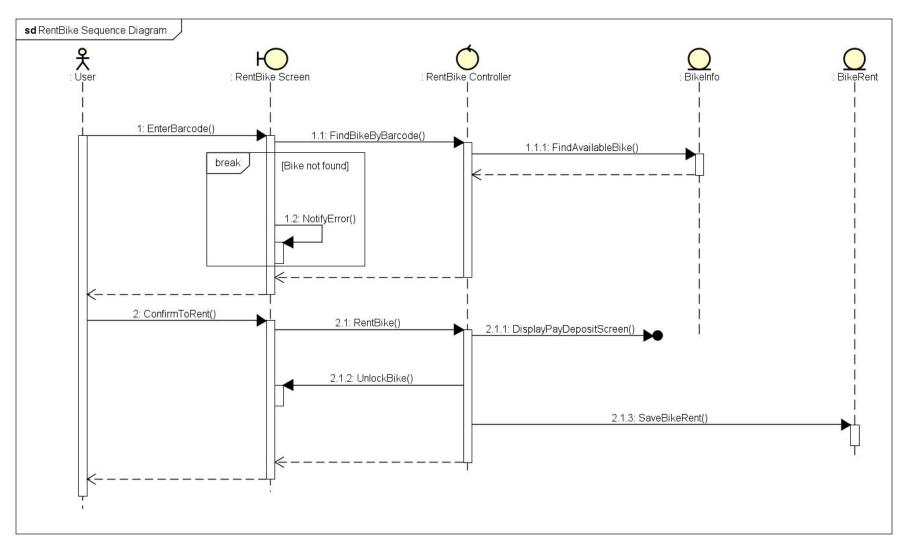
UC002: Xem thông tin bãi xe



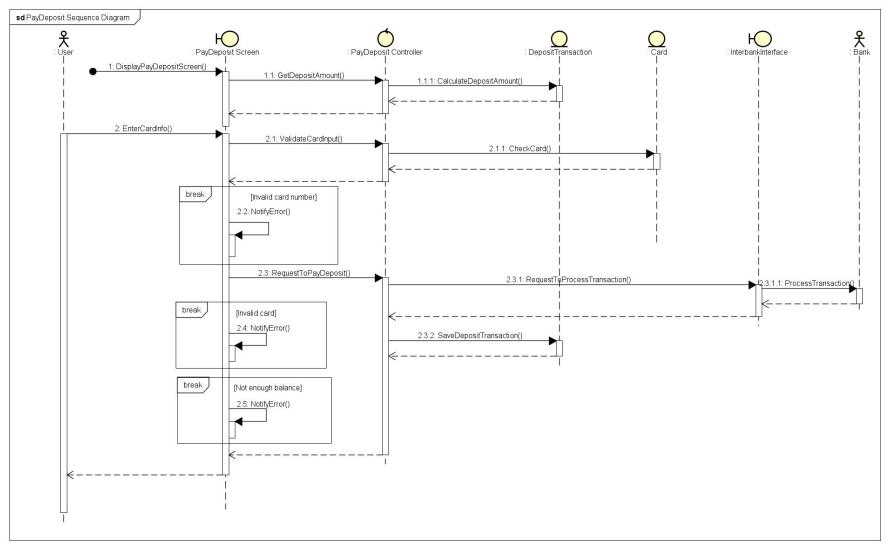
UC003: Xem thông tin xe trong bãi xe



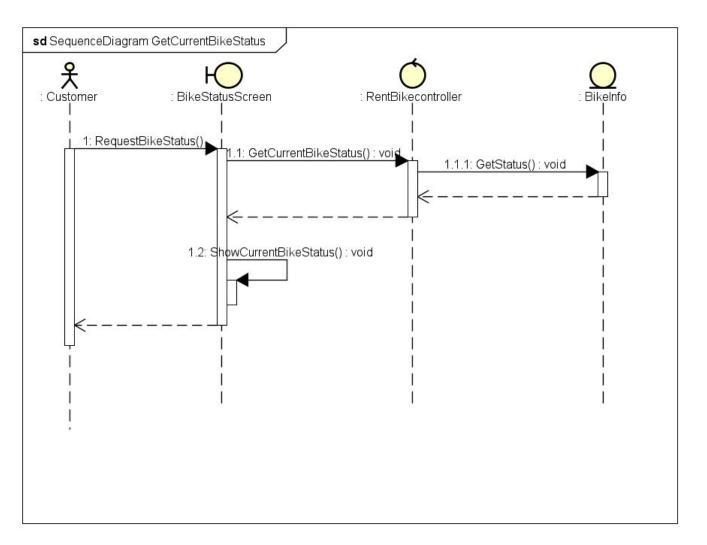
UC004: Thuê xe



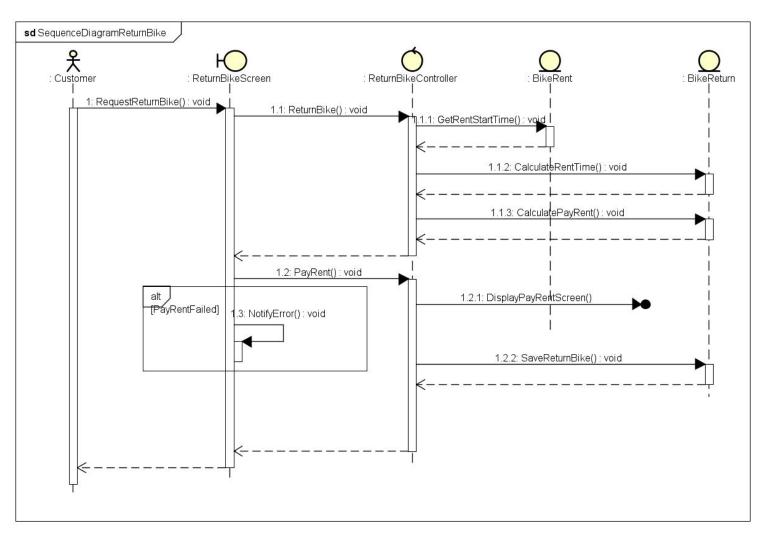
UC005: Trả tiền đặt cọc



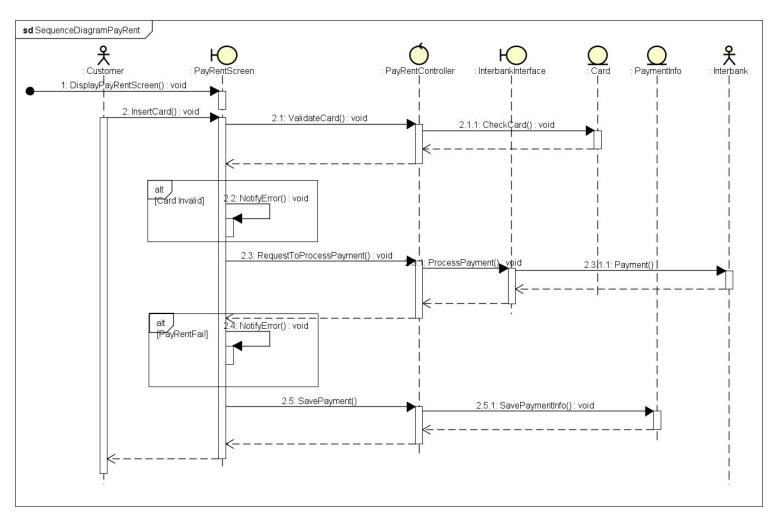
UC006: Tình trạng xe đang thuê



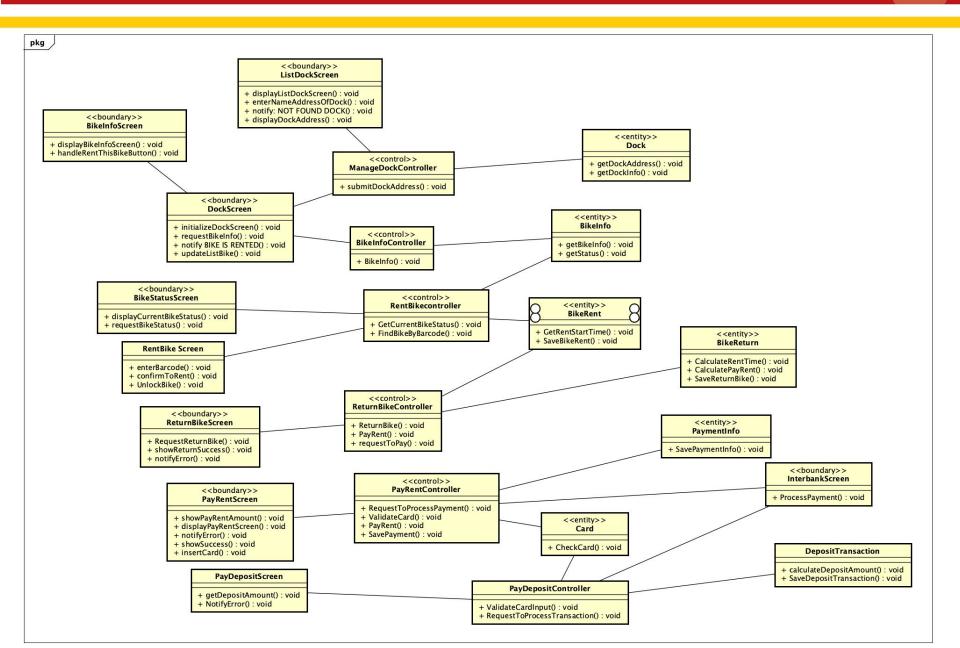
UC007: Trả xe



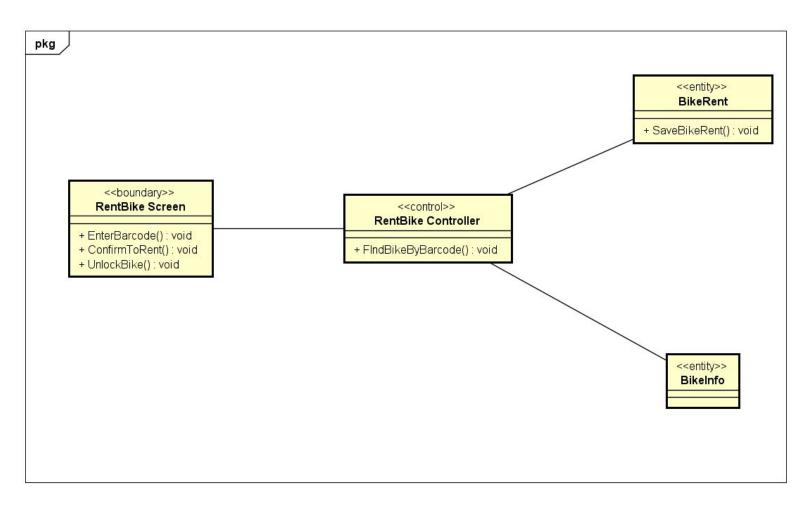
UC008: Trả tiền thuê xe



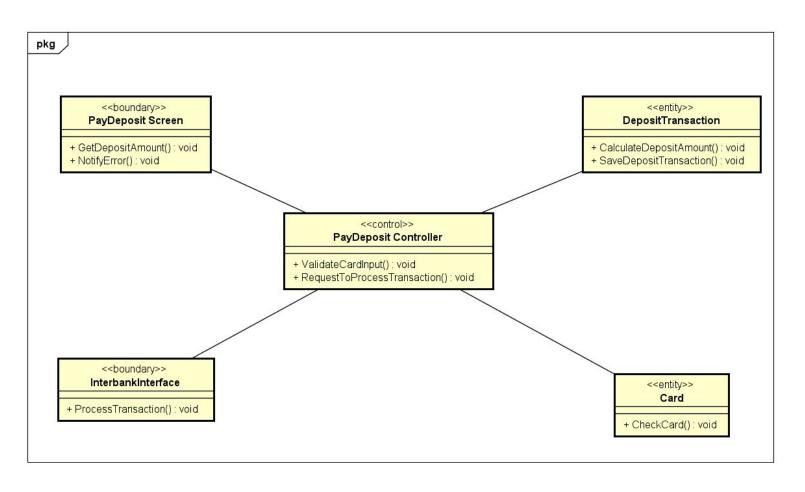
Biểu đồ lớp phân tích tổng quan



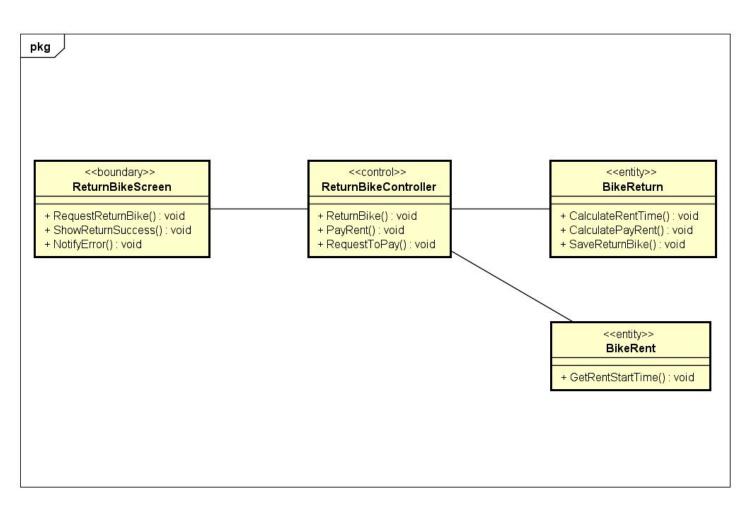
Biểu đồ lớp UC004: Thuê xe



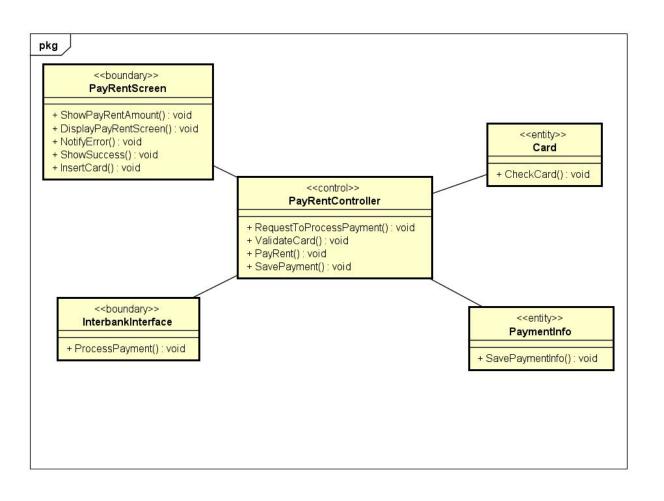
Biểu đồ lớp UC005: Trả tiền đặt cọc



Biểu đồ lớp UC007: Trả xe



Biểu đồ lớp UC008: Thanh toán tiền thuê xe



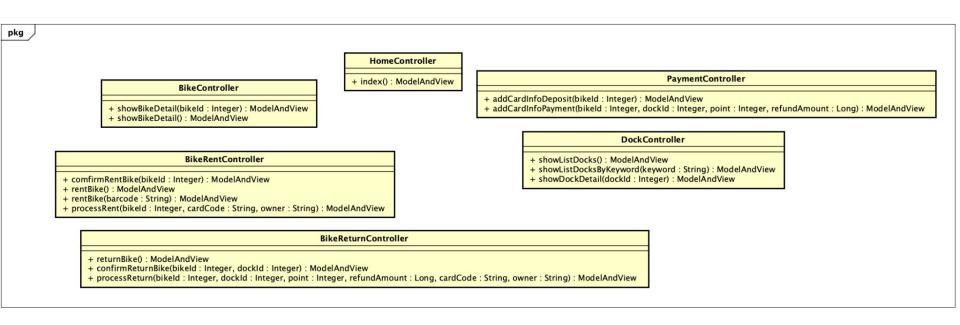
Các package:

- ▼ # EcoBikeRental.Controller
 - BikeController.java
 - BikeRentController.java
 - BikeReturnController.java
 - DockController.java
 - HomeController.java
 - PaymentController.java
- ▼ # EcoBikeRental.Entity
 - BikeCategory.java
 - BikeRent.java
 - BikeReturn.java
 - DepositTransaction.java
 - Dock.java
 - DockHasBike.java
 - MapperBikeCategory.java
 - MapperBikeRent.java
 - MapperBikeReturn.java
 - MapperDepositTransaction.java
 - MapperDock.java
 - MapperDockHasBike.java
 - MapperPaymentTransaction.java
 - PaymentTransaction.java

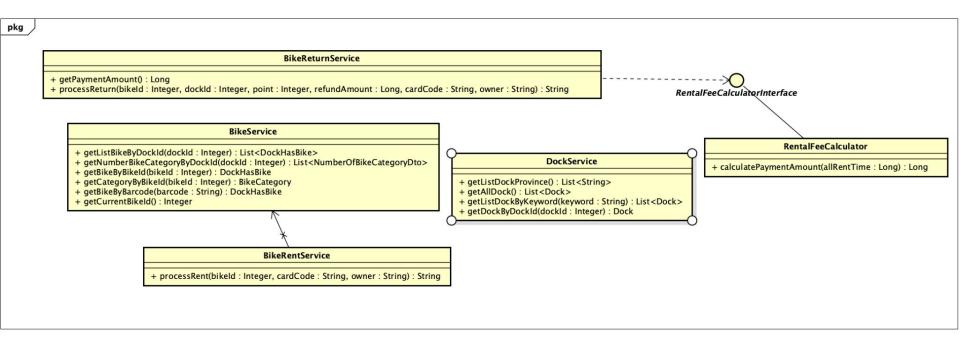
- ▼ # EcoBikeRental.Dao
 - BikeCategoryDao.java
 - BikeRentDao.java
 - BikeReturnDao.java
 - DepositTransactionDao.java
 - DockDao.java
 - DockHasBikeDao.java
 - PaymentTransactionDao.java
 - ▼ # EcoBikeRental.Service
 - BikeRentService.java
 - BikeReturnService.java
 - BikeService.java
 - DockService.java
 - RentalFeeCalculator.java
 - RentalFeeCalculatorInterface.java

- - BodyToConvertMD5Dto.java
 - NumberOfBikeCategoryDto.java
 - RequestProcessTransactionDto.java
 - RequestResetBalanceDto.java
 - ▶ ☐ RequestTransactionDto.java
 - TransactionToConvertMD5Dto.java
 - ▼ 冊 > EcoBikeRental.Subsystem
 - InterbankBoundary.java
 - InterbankController.java
 - InterbankInterface.java

Package Controller.



Package Service.



Package Dao.

pkg

BikeCategoryDao

- + getAllBikeCategory(): List<BikeCategory>
- + getCategoryByld(categoryld : Integer) : BikeCategory

BikeRentDao

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

DockDao

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

BikeReturnDao

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

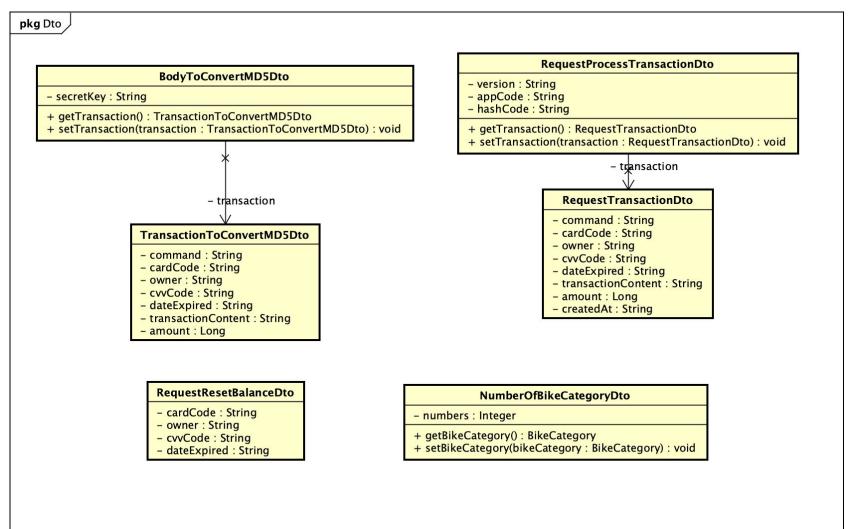
DepositTransactionDao

- + getAllDepositTransaction() : List < DepositTransaction >
- + getDepositTransactionByRentId(rentId : Integer) : DepositTransaction
- + save Deposit Transaction (deposit Transaction: Deposit Transaction): Deposit Transaction

PaymentTransactionDao

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

Package Dto.



Package Entity.

pkg Entity

Dock

dockld : Integer

- name : String

address : Stringdescription : String

- createdAt : String

- updatedAt : String

- province : String

image : StringpointNumber : Integer

DockHasBike

- id : Integer

dockld : Integer

- categoryld : Integer

- bikeld : Integer

- status : Integer

- description : String

- active : Integer

point : Integer

- barcode : String

BikeReturn

- returnId : Integer

- rentld : Integer

- returnTime : String

- dockld : Integer

- isPaid : Integer

BikeCategory

- categoryld : Integer

- name : String

description : String

- price : Integer

- image : String

BikeRent

- rentld : Integer

- userld : Integer

- bikeld : Integer

- rentTime : String

isDeposited : Integer

- dockld : Integer

PaymentTransaction

- paymentTransactionId : Integer

- rentld : Integer

- time : Long

- payment : Long

- depositTransactionId : Integer

- returnedMoney: Long

- createdTime : String

DepositTransaction

- depositTransactionId : Integer

- rentld : Integer

- moneyAmount : Long

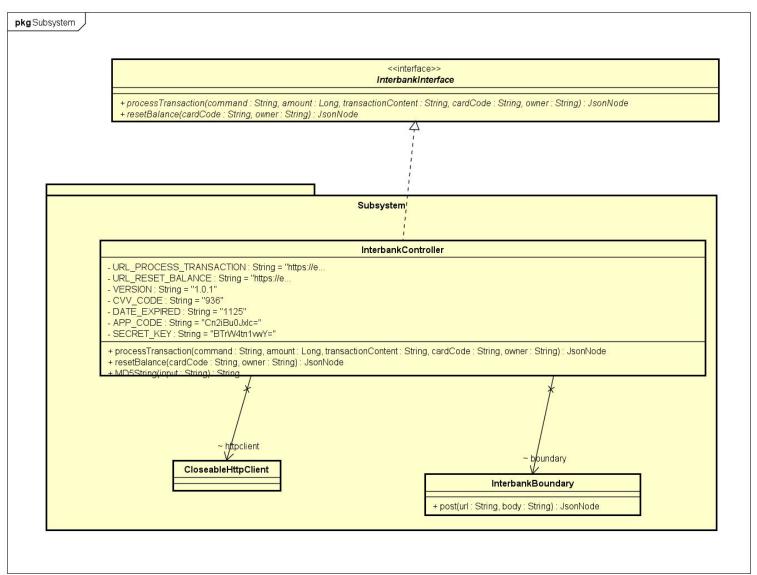
- createdAt : String

- description : String

- status : String

- depositBy : String

Package Subsystem.



Coupling/Cohesion.

- Control Coupling
 - InterbankInterface.processTransaction: có chứa tham số điều khiển
 - O Cần xem xét lại thiết kế của Interbank Interface
- Cohesion: đều ở mức chấp nhận được

```
* 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) {
        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);
```

Design principles.

- Interface Segregation:
 - Phương thức processTransaction của Interbank Interface đang đảm nhận hai việc
 - Thanh toán
 - Hoàn tiền
- Dependency inversion
 - Interbank subsystem ~ card

Design patterns.

- Dao (Data Access Object): Là một pattern trong đó các object được dành riêng cho giao tiếp với Data Layer. Trong chương trình của nhóm có sử dụng các lớp Dao: DockDao, BikeCategoryDao, DepositTransactionDao,... nhằm thực hiện mục đích truy vấn vào cơ sở dữ liệu.
- Dto (Data Transfer Object): Là một pattern được tạo ra nhằm giảm số lần gọi các method giữa các tiến trình xử lý. Thay vì truyền vào nhiều thuộc tính ta sẽ truyền vào một lớp chứa các thuộc tính đó.

VD: RequestProcessTransactionDto.java là lớp chứa các thuộc tính để truyền request body sang Web Service Interbank. Như vậy thay vì truyền vào rất nhiều thuộc tính để gửi sang Interbank, ta chỉ cần truyền một lớp Dto là đủ.

Design patterns.

- Strategy pattern: là pattern nhằm cho phép việc lựa chọn một trong nhiều chiến lược (strategy) khác nhau cho cùng một service tại runtime.
- Thay vì gắn chặt service với một strategy, chương trình sẽ lựa chọn giữa nhiều strategy tùy thuộc vào quyết định tại runtime.
- Các strategy được đóng gói một cách riêng rẽ, không phụ thuộc vào service sử dụng nó -> dễ tái sử dụng hơn
- VD: trong lớp BikeReturnService, các cách tính phí thuê xe được trừu tượng hóa thông qua interface RentalFeeCalculator

Phân công công việc.

MSSV	Tên sinh viên	SRS	SDD	Programming	Testing	Summary
20183577	Trần Xuân Lộc	Thuê xe; Trả tiền đặt cọc	Design Consideration; Phân tích lớp	DAO, DTO, Entity, Mapper	Đặc tả test case	33%
20183930	Vũ Quang Huy	Xem tình trạng xe; Trả xe; Trả tiền thuê xe	Data modeling; Thiết kế interbank subsystem	Interbank Subsystem; Bike rent & Bike return service	Cài đặt test case	33%
20180078	Trần Hữu Hiếu	Tìm kiếm bãi xe; Xem thông tin bãi; Xem thông tin xe	Thiết kế giao diện người dùng, thiết kế lớp chi tiết	Dock & Bike service		33%