Bài tập lớn môn thiết kế xây dựng hệ thống

Nhóm 5

Ngày 24 tháng 12 năm 2020

Phân công công việc

Nguyễn Văn Huy

- Thiết kế và cài đặt data model trong database
- Triển khai datalayer
- Viết test cho datalayer

Đánh giá: 25%

Trần Đình Hùng

- Thiết kế và cài đặt Interbank Subsystem
- Viết tài liệu SDD
- Viết tài liệu SRS

Đánh giá: 25%

Trương Quang Khánh

- Thiết kế và cài đặt các lớp controller và test
- Thiết kế lại hệ thống theo nguyên lý solid cùng với các design pattern
- Review, đánh giá và cải thiện bản thiết kế

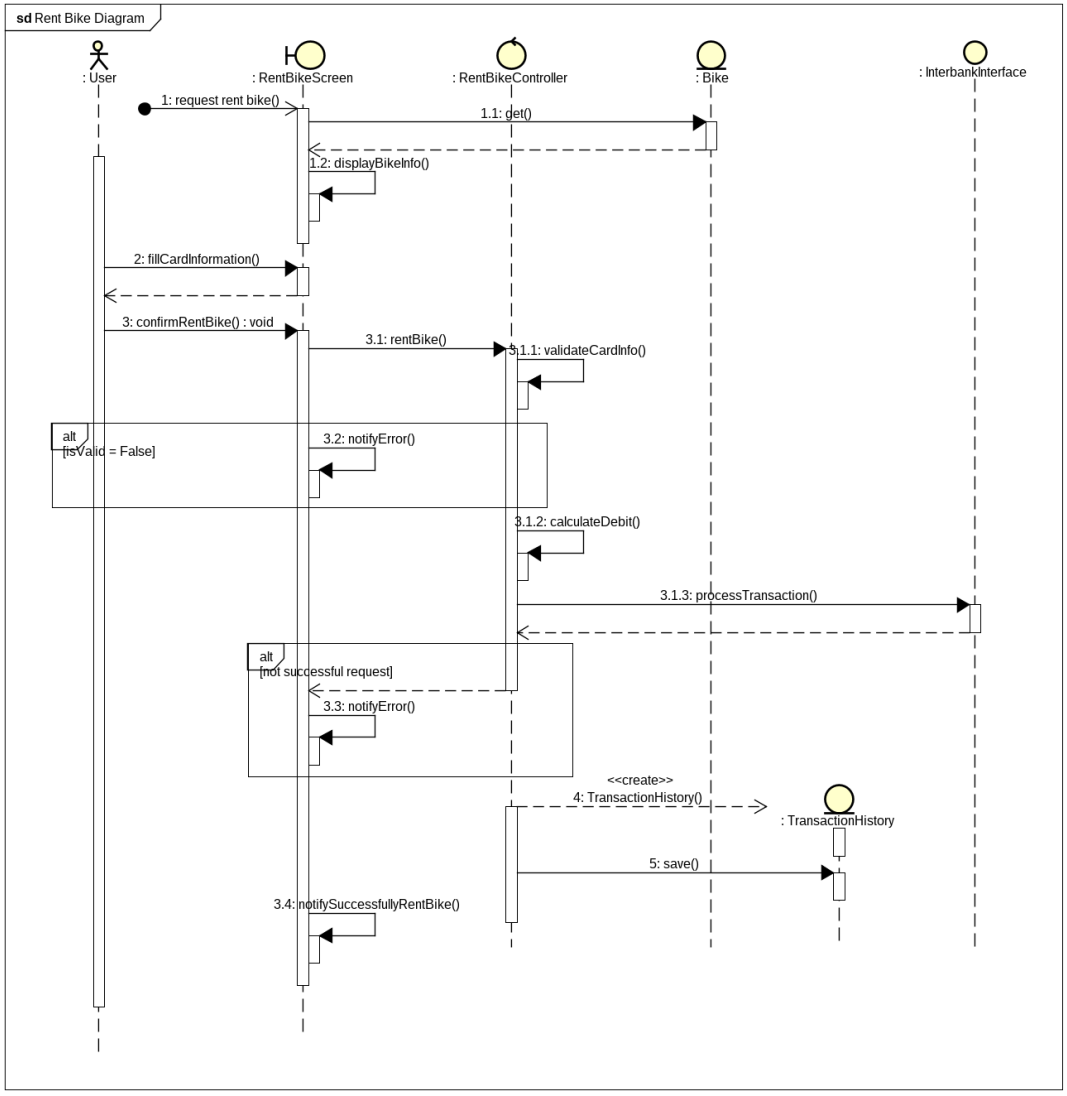
Đánh giá: 25%

Nguyễn Hồng Quốc Khánh

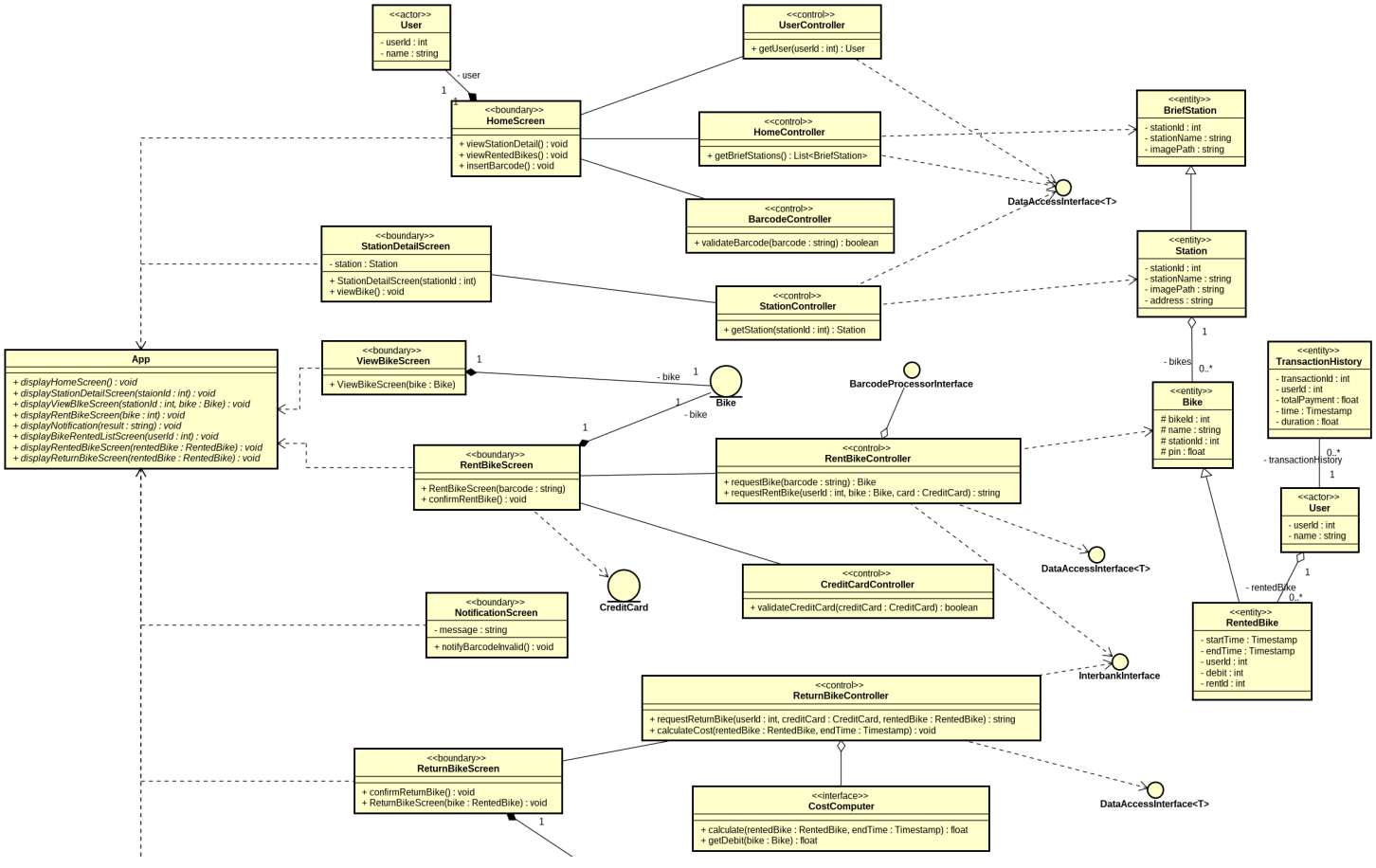
- Thiết kế và cài đặt presentationlayer với javafx
- Viết tài liệu RDD và SRS
- Review đánh giá hệ thống

Đánh giá: 25%

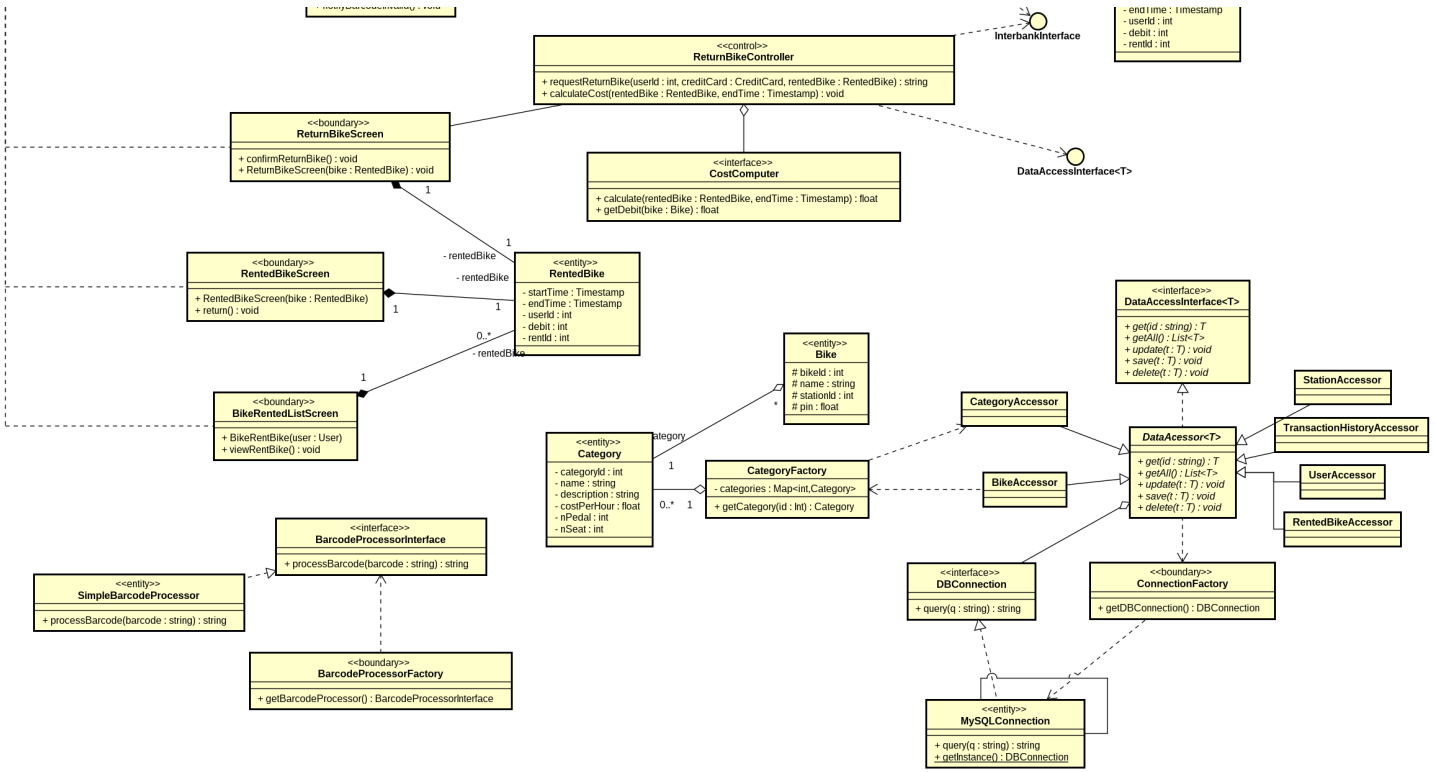
Rent Bike Sequence Diagram



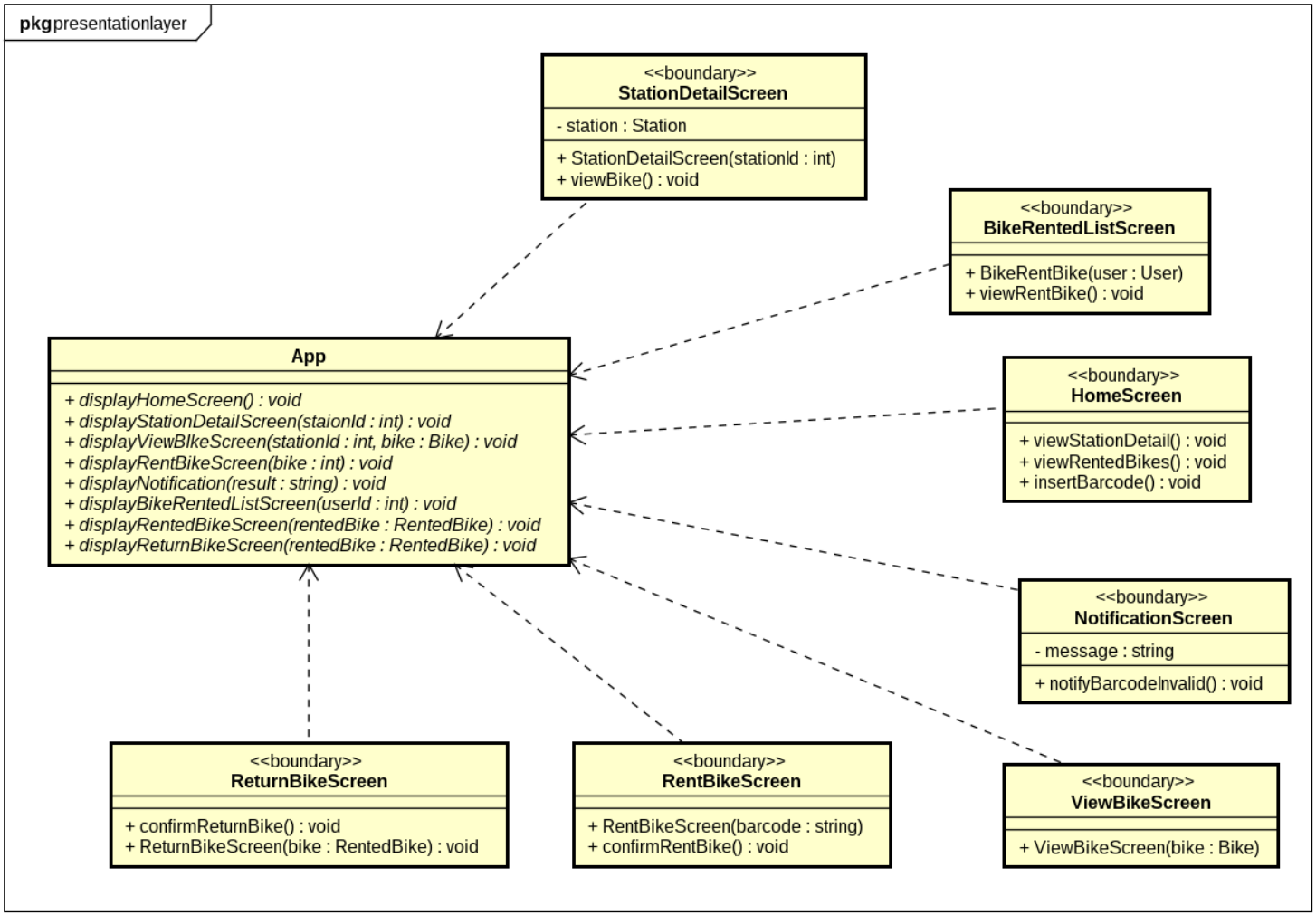
General Class Diagram



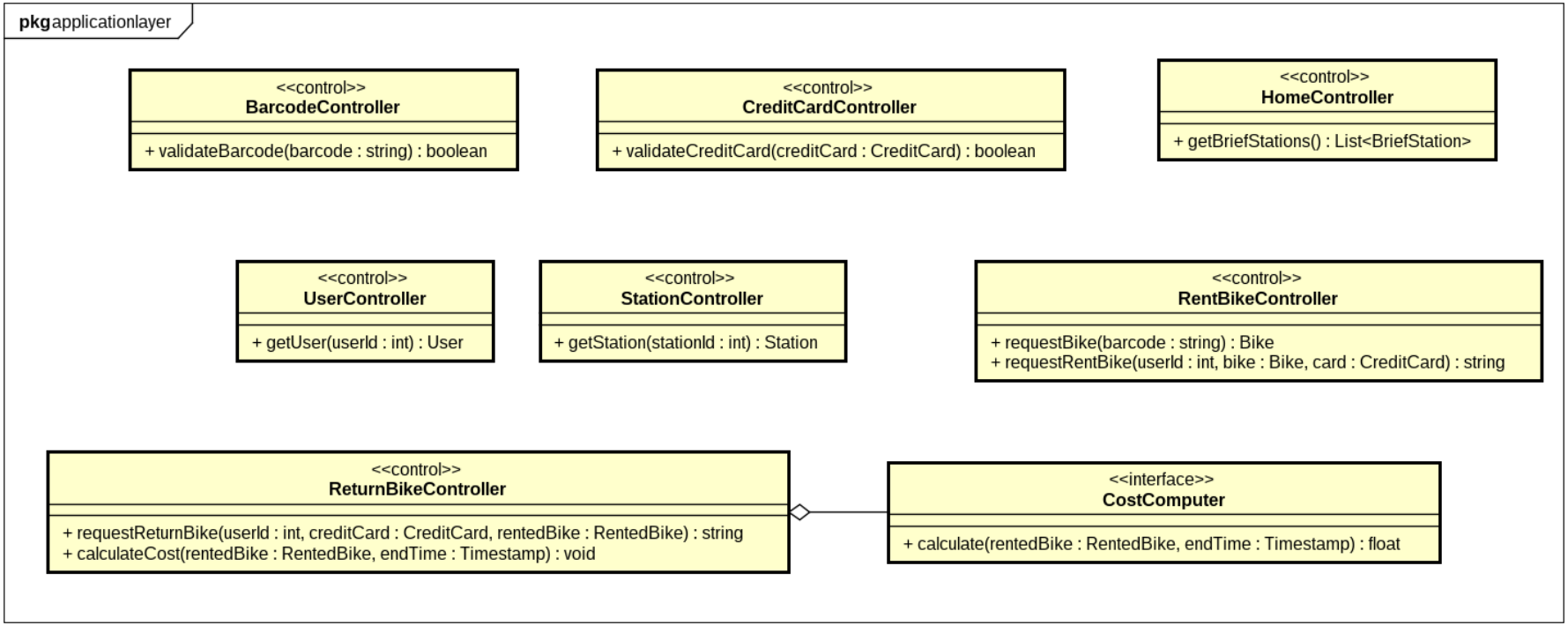
General Class Diagram



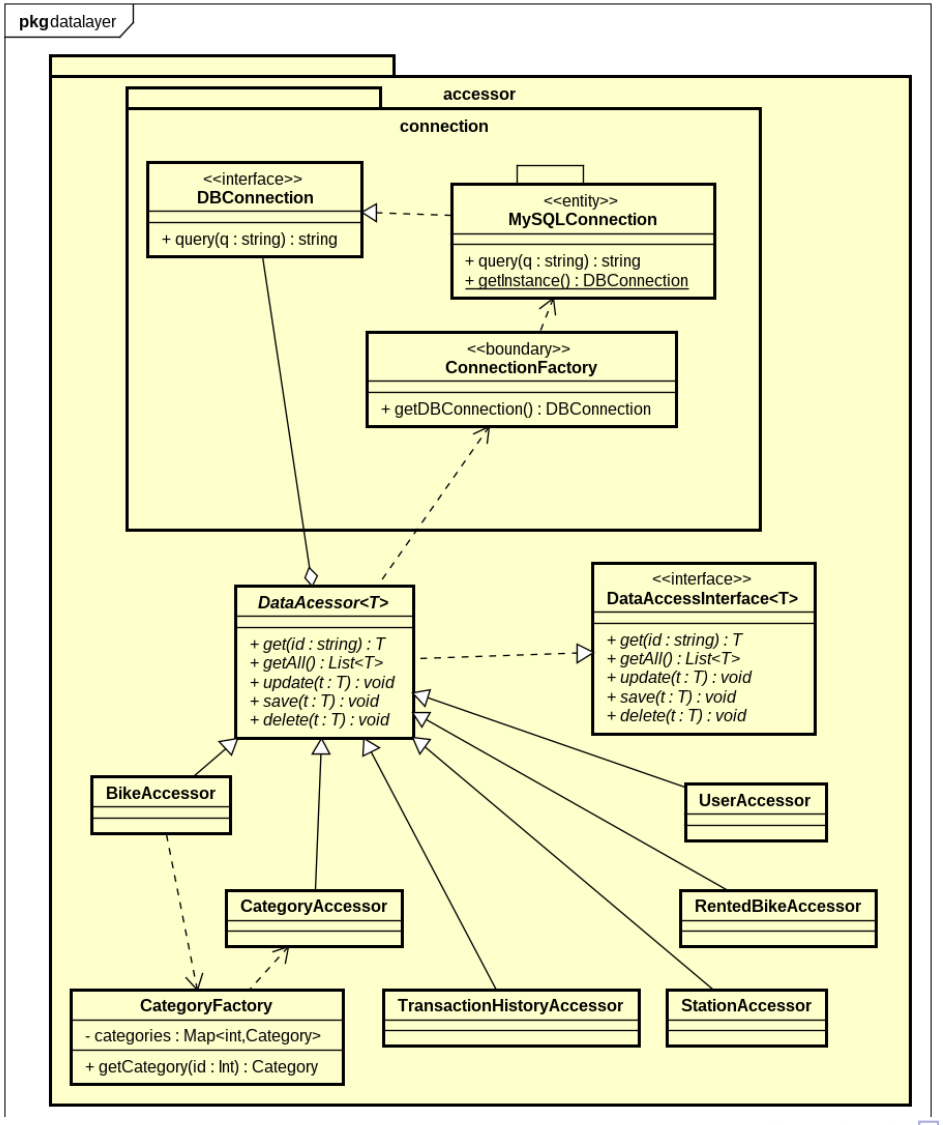
presentation layer package diagram



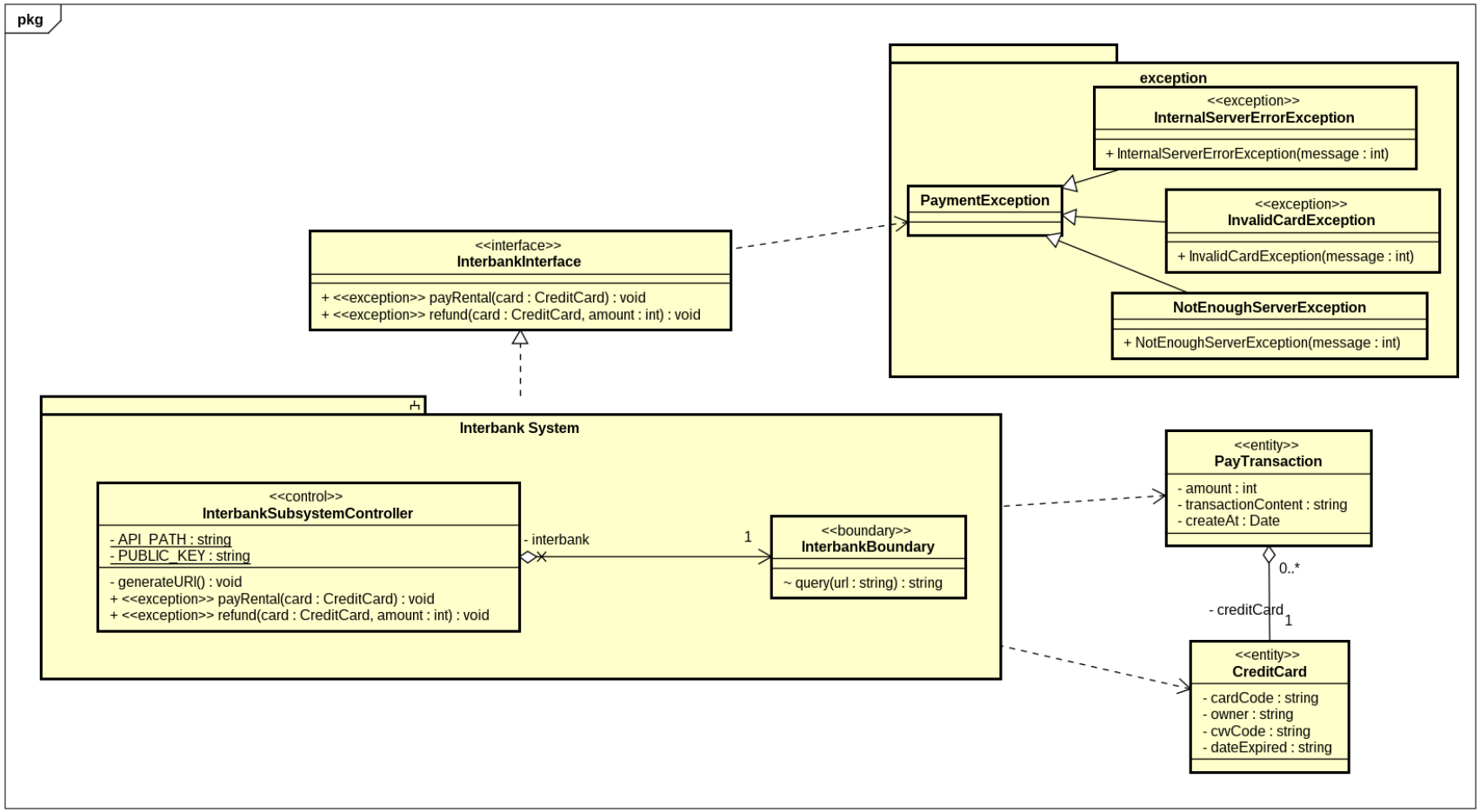
applicationlayer package diagram



datalayer package diagram

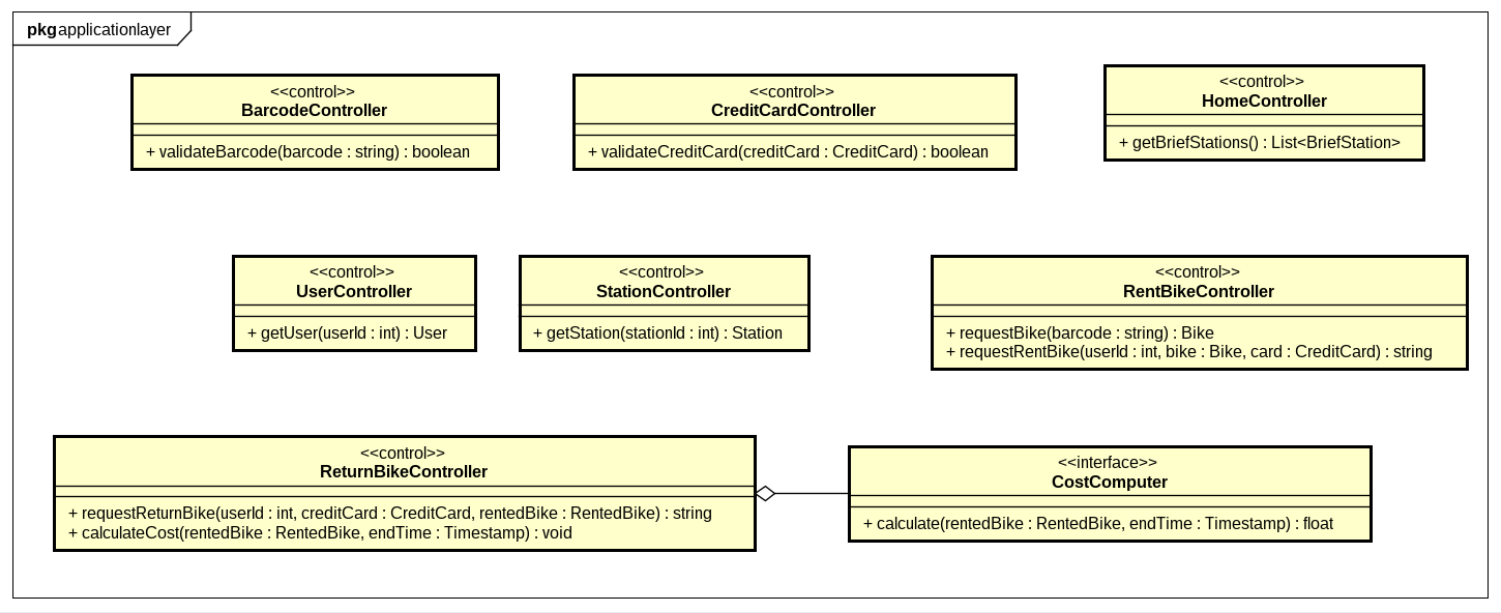


checkout package diagram



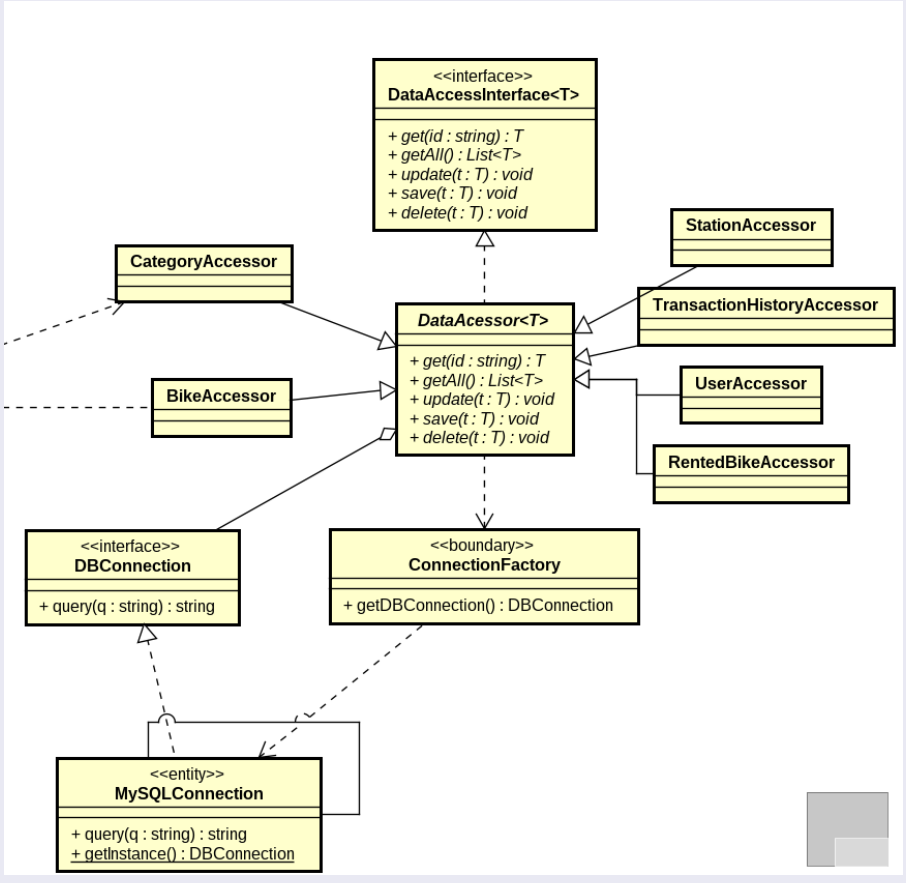
Single Responsibility

Mỗi lớp chỉ có một phương thức thể hiện cho chức năng của lớp đó



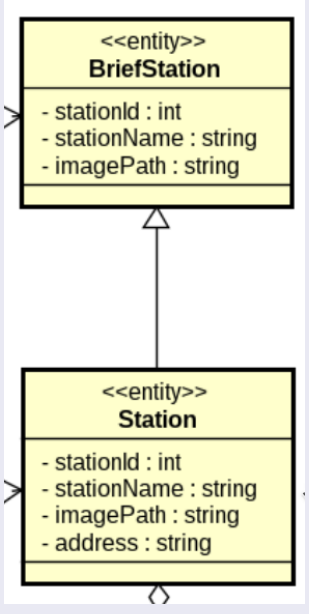
Open for extension, close for modification

Các lớp thực hiện thuật toán hay lớp thực hiện quy trình lấy dữ liệu đều là abstraction



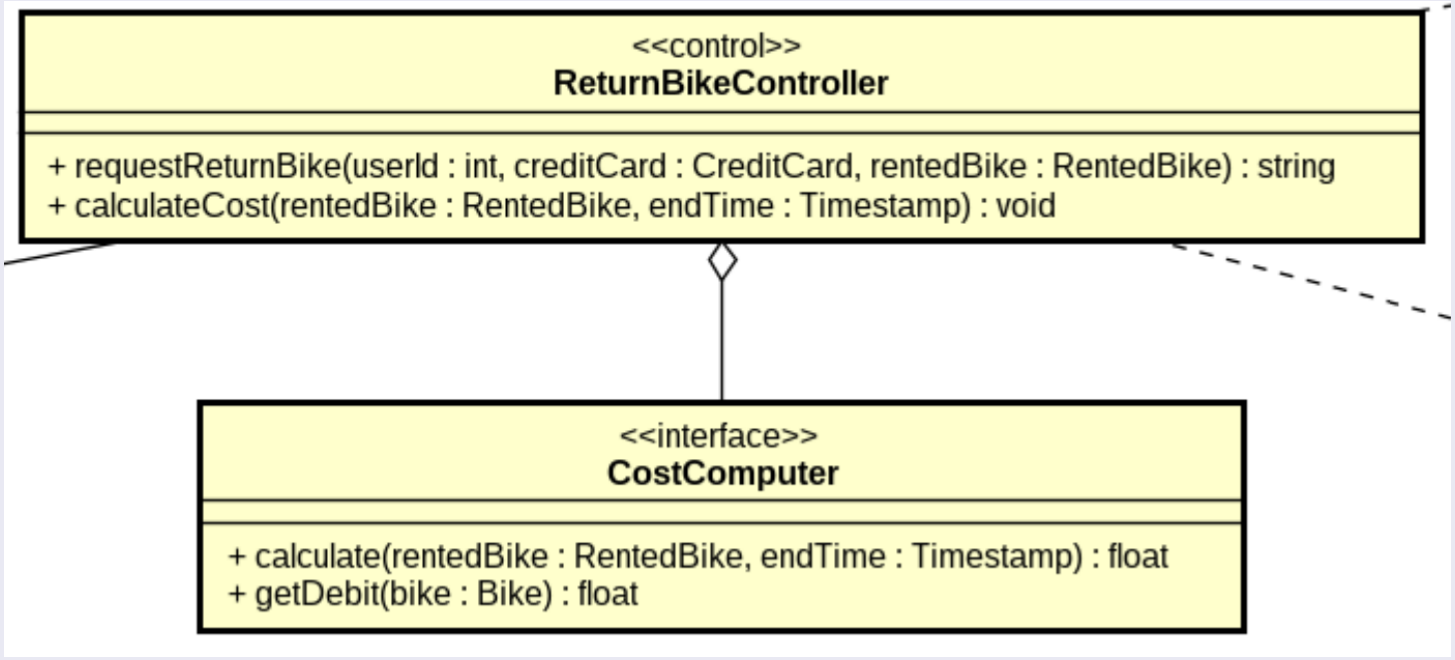
Liskov substitution

lớp Station là lớp con của BriefStation và lớp con này đều có thể thay thế cho lớp cha



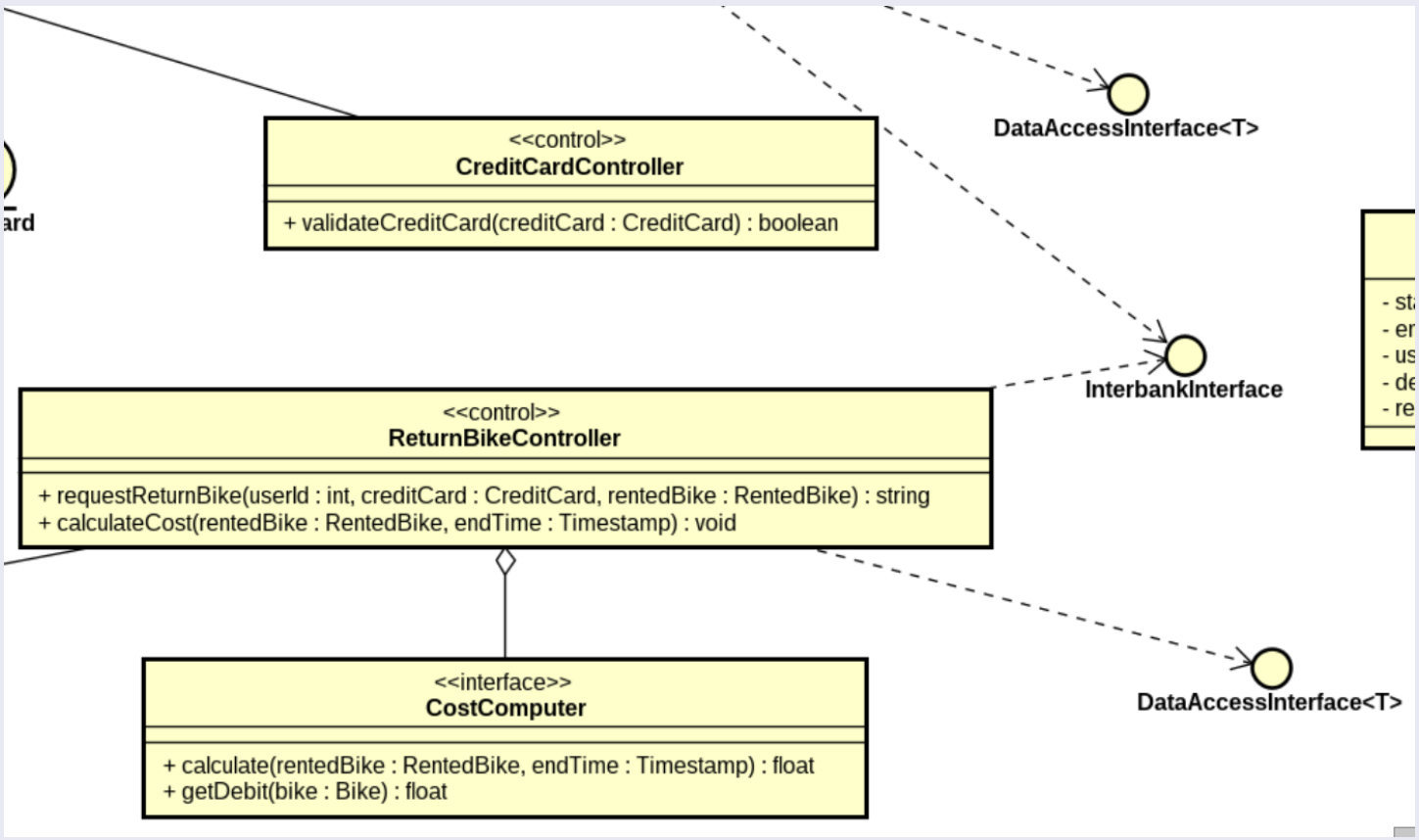
Interface segregation

Không đáp ứng được ở CostComputer Interface do nó có 2 phương thức là calculate và getDebit là độc lập với nhau nhưng lại để ở một interface



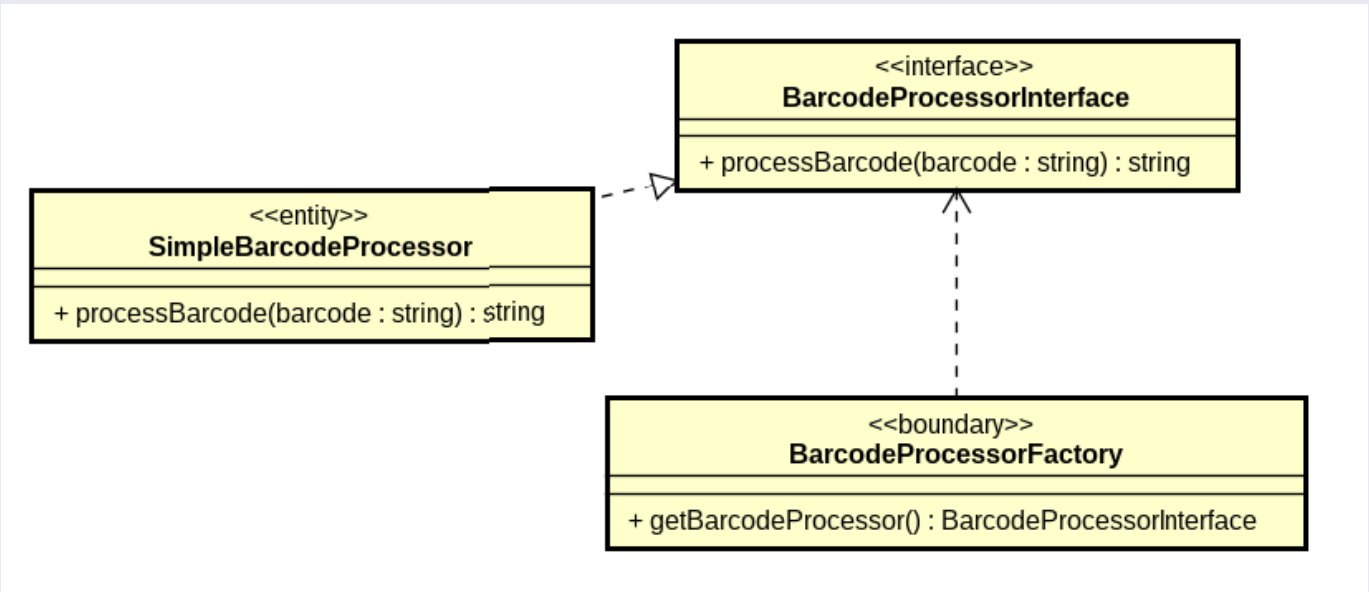
Dependency inversion

Các sự phụ thuộc giữa package controller và các package khác đều là thông qua các interface



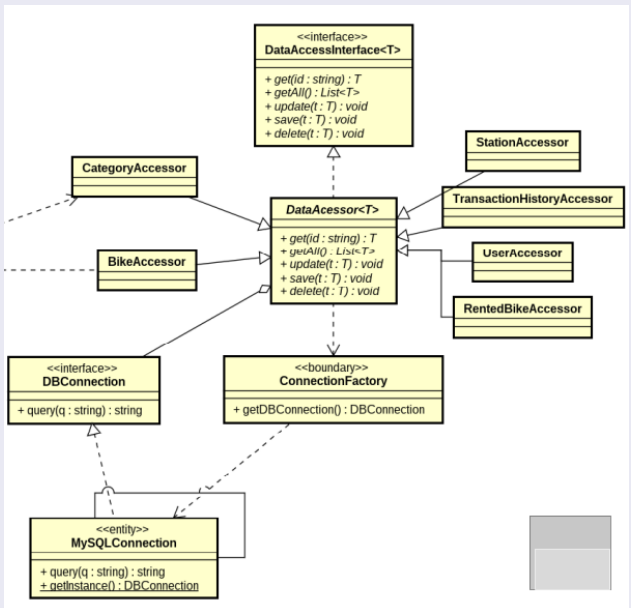
Factory Design Pattern

Việc khởi tạo lớp BarcodeProcessing có thể rất phức tạp vì trong thực tế barcode sẽ có dạng ảnh và chúng ta phải có một hệ thống con để xử lý ảnh. Vì vậy, nhóm sử dụng Factory pattern cho việc khởi tạo



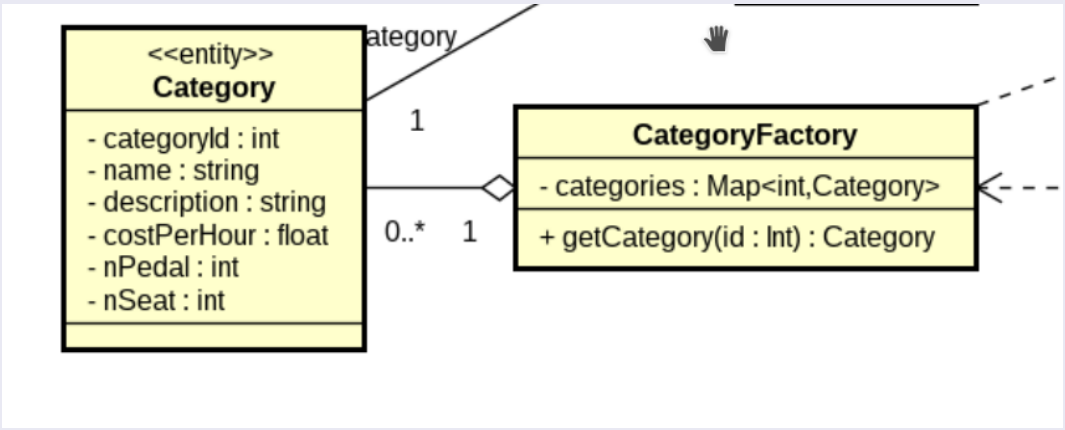
Data Access Object

Các lớp lấy dữ liệu từ nguồn dữ liệu đều có một số đặc điểm chung, nên với Data Access Object pattern thì các lớp sử dụng sẽ không cần quan tầm đến từng lớp lấy dữ liệu cụ thể sẽ có tên là gì mà chỉ cần biết đến interface DAO với kiểu dữ liệu entity tương ứng với nó.



Fly Weight

Nhận thấy lớp Category sẽ không bao giờ thay đổi trạng thái, cũng như là rất nhiều bike chỉ cần dùng một trạng thái thuộc tính của category nên dùng mẫu này để tối ưu bộ nhớ lưu trữ, chi phí cho connection lấy dữ liệu từ database cũng như thời gian tạo mới một category



Visitor (Đã thiết kế nhưng do điều kiện nên không thực hiện)

Nhận thấy lớp App trong package presentaionlayer nhận nhiệm vụ thực hiện display màn hình, và phụ thuộc vào từng màn hình thì sẽ có một kiểu display riêng. Như vậy có thể thiết kế Presentor là một visitor với phương thức visit là display còn các màn hình là lớp dẫn suất của interface Screen tương ứng là acceptor