TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công nghệ thông tin và Truyền thông

Software Design Document

HỆ THỐNG GỬI XE ECOBIKE-RENTAL

Môn : Thiết kế xây dựng phần mềm

Giảng viên : TS. Nguyễn Thị Thu Trang

Nhóm 8

Danh sách sinh viên: Lê Tường Khanh – 20180109

Nguyễn Phi Phúc – 20180155

Bùi Đức Tuấn Dũng -20180048

## Table of Contents

[Table of Contents 1](#_TOC_250025)

1. [Introduction 3](#_TOC_250024)
   1. [Objective 3](#_TOC_250023)
   2. [Scope 3](#_TOC_250022)
   3. [Glossary 3](#_TOC_250021)
   4. [References 3](#_TOC_250020)
2. [Overall Description 5](#_TOC_250019)
   1. [General Overview 5](#_TOC_250018)
   2. [Assumptions/Constraints/Risks 8](#_TOC_250017)
      1. Assumptions 8
      2. Constraints 9
      3. Risks 9
3. [System Architecture and Architecture Design 10](#_TOC_250016)
   1. [Architectural Patterns 10](#_TOC_250015)
   2. [Interaction Diagrams 10](#_TOC_250014)
   3. [Analysis Class Diagrams 13](#_TOC_250013)
   4. [Unified Analysis Class Diagram 15](#_TOC_250012)
   5. [Security Software Architecture 15](#_TOC_250011)
4. [Detailed Design 16](#_TOC_250010)
   1. [User Interface Design 16](#_TOC_250009)
      1. Screen Configuration Standardization 16
      2. Screen Transition Diagrams 18
      3. Screen Specifications 19
   2. [Data Modeling 32](#_TOC_250008)
      1. Conceptual Data Modeling 32
      2. Database Design 32
   3. [Non-Database Management System Files 36](#_TOC_250007)
   4. [Class Design 36](#_TOC_250006)
      1. General Class Diagram 36
      2. Class Diagrams 38
      3. Class Design 42
5. [Design Considerations 54](#_TOC_250005)
   1. [Goals and Guidelines 54](#_TOC_250004)
   2. [Architectural Strategies 54](#_TOC_250003)
   3. [Coupling and Cohesion 54](#_TOC_250002)
   4. [Design Principles 56](#_TOC_250001)
   5. [Design Patterns 58](#_TOC_250000)

# Introduction

### Objective

Tài liệu này mô tả phần thiết kế phần mềm sau bước phân tích ở tài liệu srs. Tài liệu được sử dụng cho programmers, testers, maintainers, systems integrators, vv. Nó bao gồm việc thiết kế chi tiết cho kiển trúc, thiết kế giao diện và thiết kế lớp cho từng chức năng của hệ thống, cũng như việc thiết kế cơ sở dữ liệu của cả hệ thống để từ đó người đọc sẽ có cái nhìn rõ ràng hơn về phần mềm cần xây dựng và nó sẽ là tài liệu chính thức để từ đó những người xây dựng phần mềm có thể xây dựng nên phần mềm dựa vào tài liệu này.

### Scope

Phần mềm có tên: Eco Bike Rental

Phần mềm sẽ cung cấp cho người dùng một giao diện để thực hiện các chức năng: xem thông chi tiết bãi xe, xem thông tin chi tiết xe trong bãi, thuê xe và trả xe.

### Glossary

Bảng chú thích các thuật ngữ:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuật ngữ | Giải thích | Ví dụ | Ghi chú |
| 1 | Button | Các nút bấm trong phần  mềm |  |  |
| 2 | Cơ sở dữ liệu | Là nơi lưu trữ các thông tin về dữ liệu cần thiết: như danh sách bãi xe, danh sách xe trong bãi, lịch sử các giao  dịch |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | List View | Một danh sách đọc ra từ cơ  sở dữ liệu, có thể ấn chọn vào phần tử trong danh sách |  |  |
| 4 | Text Field | Một ô trống, thường dùng cho người dùng có thể điền  các thông tin cần thiết |  |  |

### References

Tài liệu tham khảo bao gồm: javaFX doc, một số design pattern thông dụng, các tài liệu được cung cấp trong học phần thiết kế và xây dựng phần mềm.

# Overall Description

### General Overview

Ngày nay, nhu cầu tập thể dục thể thao nâng cao sức khỏe của con người ngày càng gia tăng. Một trong những hình thức tập luyện vừa hiệu quả vừa mang lại sử thư giãn tinh thần cao đó chính là đi xe đạp. Xuất phát từ nhu cầu thực tế này, phần mềm quản lí việc cho thuê trả xe ra đời.

Thông qua việc phân tích yêu cầu mà hệ thống đặt ra, nhóm đã thống nhất một số diagram tổng quan cho hệ thống bao gồm:

Sơ đồ UC tổng quan hệ thống:

Diagram

Description automatically generated

Biểu đồ hoạt động usecase xem chi tiết xe trong bãi

Diagram

Description automatically generated

Biểu đồ hoạt động usecase thuê xe

Diagram

Description automatically generated

Biểu đồ hoạt động usecase trả xe

Diagram

Description automatically generated

Biểu đồ hoạt động usecase xem xe đang thuê

Diagram

Description automatically generated

Biểu đồ hoạt động tìm kiếm bãi xe

A picture containing diagram

Description automatically generated

### Assumptions/Constraints/Risks

* + 1. **Assumptions**

Một số yêu cầu phát sinh:

* + - * *Thêm một loại xe mới:* Xe đạp đôi điện, giống xe đạp đôi thường nhưng có motor điện giúp đạp xe nhanh hơn. Cách tính giá xe đạp đôi điện đắt gấp 02 lần cách tính giá xe đạp đơn thường.
      * *Thay đổi cách tính giá thuê xe*
      * *Thêm một tính năng mới:* Cho phép khách hàng tạm dừng thời gian thuê xe. Khách hàng có thể tạm dừng thuê xe mọi lúc bằng cách khóa xe. Lúc này, hệ thống sẽ tạm dừng tính thời gian thuê xe. Hệ thống sẽ tự động tiếp tục tính thời gian thuê xe ngay khi khách hàng mở khóa xe
    1. **Constraints**

Một số ràng buộc để có thể chạy được phần mềm bao gồm:

* Phần cứng: máy tính pc hoặc máy tính xách tay, hệ điều hành window,mac osx hoặc ubuntu đều được
* Phần mềm: 1, Có máy ảo java 2, IDE intelji

3, Thư viện cần thiết:

* mysql-connector-java-5.1.49,
* org.junit.jupiter:junit-jupiter:5.4.2,
* mysql-connector-java-5.1.49.jar
* javafx-sdk-17.0.1
* org.json:json:20201115
* org.apache.httpcomponents:httpclient:4.3-alpha1

4, Database: mysql server, mysql workbench

* + 1. **Risks**

Rủi ro có thể có của phần mềm: bảo mật thấp nên dữ liệu trong database dễ bị lấy mất. Tuy nhiên do phần mềm mang tính chất demo là chính nên vấn đề này không mang lại thiệt hại quá lớn cho cá nhân hay tổ chức nào.

# System Architecture and Architecture Design

### Architectural Patterns

### Interaction Diagrams

### Use case Rent Bike

### Sequence Diagram

### Diagram Description automatically generated

### Communication diagram

### Diagram Description automatically generated

### Use case Return Bike

### Sequence Diagram

### Diagram Description automatically generated

### Communication diagram

### Diagram Description automatically generated

### Use case Search Dock

### Sequence Diagram

### Diagram, engineering drawing Description automatically generated

### Communication diagram

### Diagram Description automatically generated

### Use case View Renting Bike

### Sequence Diagram

### Diagram Description automatically generated

### Communication diagram

### Diagram Description automatically generated

### Use case View Bike In Station

### Sequence Diagram

### Diagram Description automatically generated

### Communication diagram

### Diagram Description automatically generated

### Analysis Class Diagrams

* + 1. **Biểu đồ lớp cho use case Rent Bike**

Diagram

Description automatically generated

* + 1. **Biểu đồ lớp cho use case Return Bike**

**Diagram

Description automatically generated**

* + 1. **Biểu đồ lớp cho use case Search Dock**

**Diagram

Description automatically generated**

* + 1. **Biểu đồ lớp cho use case View Renting Bike**

**Diagram

Description automatically generated**

* + 1. **Biểu đồ lớp cho use case View Bike In Station**

**Diagram

Description automatically generated**

### Unified Analysis Class Diagram

Biểu đồ lớp phân tích cho hệ thống

Diagram

Description automatically generated

### Security Software Architecture

Nếu sau này hệ thống mở rộng thêm chức năng đăng nhập cho người dùng thì người dùng phải nhập mật khẩu với mỗi role tương ứng(như khách hàng hay quản lí bãi xe).

# Detailed Design

### User Interface Design

* + 1. **Screen Configuration Standardization**

***Display***

Số lượng màu được hỗ trợ: 16,777,216 màu Độ phân giải: 1150 x 750 pixels

***Screen***

Vị trí của của button: ở trên thanh header phía trên hoặc ở phần trung tâm màn hình, tùy thuộc vào cấu hình màn hình. Riêng button Back luôn nằm ở góc trên bên phải màn hình.

Vị trí của message: ở giữa trung tâm khung màn hình

Vị trí của screen title: title đặt ở góc trên bên trái của màn hình

Sự nhất quán trong hiển thị chữ số: dấu phẩy để phân cách hàng nghìn và chuỗi chỉ bao gồm các ký tự, chữ số, dấu phẩy, dấu chấm, dấu cách, dấu gạch dưới và ký hiệu gạch nối.

Template: tất cả các màn hình đều có một header mầu trắng và footer màu xám, một ảnh logo và một ảnh minh họa ở phía bên trái hoặc phải .

***Control***

Kích thước text: Color: tùy thuộc vào button chứa nó và tính chất của text.

Xử lý check input: Trước tiên kiểm tra xem input có empty hay không. Tiếp theo, kiểm tra xem input có đúng format hay không.

Dịch chuyển màn hình: Không có các khung chồng lên nhau. Các màn hình được tách biệt. Tuy nhiên, hướng dẫn sử dụng được xem như là 1 popup message vì màn hình chính ở dưới sẽ không thể thao tác trong khi màn hình hướng dẫn sử dụng đang được hiển thị. Ban đầu khi app khởi chạy thì màn hình đầu tiên(Splash Screen) sẽ xuất hiện.

Thứ tự xuất hiện các màn hình trong hệ thống: (Do tùy thuộc vào người dùng sử dụng chức năng nào của app và click vào button tương ứng nào nên thứ tự này chỉ là tương đối).

1. Splash screen
2. Home screen
3. Rent bike screen – Màn hình thuê xe
4. Result screen – Màn hình hiển thị kết quả
5. Rented Bike List Screen – Màn hình hiển thị thông tin danh sách xe đang thuê của người dùng
6. Return bike screen – Màn hình hiển thị trr xe
7. Dock detail screen – Màn hình hiển thị thông tin các xe trong bãi xe
8. Bike detail screen – Màn hình hiển thị thông tin chi tiết của xe

***Nhập input từ bàn phím***

Sẽ không có phím tắt. Có các button quay lại để quay lại các màn hình trước đó. Ngoài ra button “X” nằm ở thanh tiêu đề bên phải để đóng screen như thông thường.

***Error***

Một thông điệp sẽ được hiện lên để thông báo cho người dùng biết vấn đề đang gặp phải là gì. Thông điệp có dạng text màu đỏ.

* + 1. **Screen Transition Diagrams**

Sơ đồ dịch chuyển giữa các màn hình:

**Diagram

Description automatically generated**

* + 1. **Screen Specifications**
       1. ***Splash Screen:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ECO-BIKE SOFTWARE | | **Date**  **of creation** | **Approved by** | **Reviewed**  **by** | **Person in**  **charge** |
| Screen specification | Splash Screen | 07/01/2022 |  |  | Nguyễn Phi Phúc |
| **Background pattern  Description automatically generated** | | **Control** | **Operation** | **Function** | |
| Khu vực hiển thị logo Eco Bike | Initial | Hiển thị logo chương trình Eco Bike ở giữa màn hình | |

* + - 1. ***Home Screen:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ECO-BIKE SOFTWARE | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| Screen specification | Home Screen | 07/01/2022 |  |  | Nguyễn Phi Phúc |
| **Graphical user interface  Description automatically generated** | | **Control** | **Operation** | **Function** | |
| Khu vực hiển thị logo, thanh search và logo cùng với số lượng xe đã thuê | Initial | Hiển thị logo, thanh search và xe đã thuê ở trên cùng màn hình | |
| Khu vực chèn barcode | Initial | Hiển thị thanh chèn barcode và nút bấm xử lý barcode | |
| Khu vực hiển thị danh sách bãi xe | Initial | Hiển thị danh sách bãi xe cùng với thông tin từng bãi theo dạng bảng | |
| Nút “Search” | Click | Tìm kiếm bãi xe có thông tin được người dùng nhập | |
| Nút Option | Click and Select | Chọn đặc tính của bãi xe mà người dùng muốn để search (theo tên hoặc địa chỉ bãi xe) | |
| Nút Xem xe đã thuê | Click | Hiển thị màn hình Rented Bike List Screen. | |
| Nút Process Barcode | Click | Gọi tới api của server barcode bên ngoài hệ thống để trả về id của xe, sau khi có id sẽ chuyển tới màn hình Return Bike Screen. | |
| Logo | Click | Gọi về giao diện HomeScreen | |
| Table Row | Click | Click vào 1 hàng trong bảng để chọn bãi xe tương ứng muốn xem và chuyển tới màn hình DockDetailScreen | |

**Định nghĩa các trường thuộc tính:**

|  |  |
| --- | --- |
| **Screen Name** | HomeScreen |
| **Attribute** | **Number of digits (bytes)** | **Type** | **Field Attribute** | **Remarks** |
| Thông tin tìm kiếm | 200 | String | Đen | Căn trái |
| Số lượng xe đang thuê | 10 | Numeral | Đỏ | Căn giữa |
| Thông tin chèn barcode | 50 | String | Đen | Căn trái |
| Tên bãi xe | 50 | String | Đen | Căn trái |
| Địa chỉ bãi xe | 200 | String | Đen | Căn trái |
| Số lượng xe có sẵn | 10 | Numeral | Đen | Căn giữa |
| Khoảng cách | 10 | Numeral | Đen | Căn giữa |
| Thời gian đi bộ | 10 | Numeral | Đen | Căn giữa |

* + - 1. ***Rent Bike Screen:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ECO-BIKE SOFTWARE | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| Screen specification | Rent Bike Screen | 07/01/2022 |  |  | Nguyễn Phi Phúc |
| **Graphical user interface, application, website  Description automatically generated** | | **Control** | **Operation** | **Function** | |
| Khu vực hiển thị thông tin xe muốn thuê và ảnh của xe | Initial | Hiển thị thông tin xe muốn thuê và ảnh của xe | |
| Khu vực hiển thị form credit card mà người dùng cần điền | Initial | Hiển thị form card gồm 4 trường thuộc tính | |
| Nút confirm rent bike | Click | Gọi tới api của interbank để thuê xe và chuyển tới màn hình Result Screen | |
| Nút valid information | Click | Điền sẵn thông tin về card của nhóm vào form | |
| Nút Back | Click | Quay trở lại màn hình HomeScreen | |
| Logo | Click | Quay về màn hình HomeScreen | |

**Định nghĩa các trường thuộc tính:**

|  |  |
| --- | --- |
| **Screen Name** | Rent Bike Screen |
| **Attribute** | **Number of digits (bytes)** | **Type** | **Field Attribute** | **Remarks** |
| Card Holder Name | 50 | String | Đen | Căn trái |
| Card Number | 50 | String | Đen | Căn trái |
| CVV | 10 | String | Đen | Căn trái |
| Expiration Date | 10 | String | Đen | Căn trái |
| Mã số xe | 10 | Numeral | Đen | Căn trái |
| Tên xe | 50 | String | Đen | Căn trái |
| Trạng thái xe | 10 | Boolean | Đen | Căn trái |
| Biển số xe | 50 | String | Đen | Căn trái |
| **….** (Một số trường khác của thông tin xe) |  |  |  |  |

* + - 1. ***Result Screen:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ECO-BIKE SOFTWARE | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| Screen specification | Result Screen | 07/01/2022 |  |  | Nguyễn Phi Phúc |
| **Graphical user interface, text, application  Description automatically generated** | | **Control** | **Operation** | **Function** | |
| Khu vực hiển thị logo và nút Home | Initial | Hiển thị thanh chứa logo và nút Home ở vị trí trên cùng | |
| Khu vực hiển thị message result | Initial | Hiển thị giữa màn hình | |
| Nút Home | Click | Quay về màn hình HomeScreen | |
|  |  |  | |
|  |  |  | |

**Định nghĩa các trường thuộc tính:**

|  |  |
| --- | --- |
| **Screen Name** | Result Screens |
| **Attribute** | **Number of digits (bytes)** | **Type** | **Field Attribute** | **Remarks** |
| Message | 200 | String | Xanh đậm | Căn giữa |

* + - 1. ***Rented Bike List Screen:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ECO-BIKE SOFTWARE | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| Screen specification | Rented Bike List Screen | 07/01/2022 |  |  | Nguyễn Phi Phúc |
| **Graphical user interface, application  Description automatically generated** | | **Control** | **Operation** | **Function** | |
| Khu vực List View chứa thông tin các xe đã được thuê | Initial | Hiển thị các xe đã được thuê | |
| Khu vực hiển thị thời gian xe được thuê cho tới nay | Initial | Hiển thị thời gian xe được thuê cho tới nay | |
| List View | Click | Chọn lựa 1 xe để return và click để di chuyển tới màn hình ReturnBikeScreen của xe đó. | |
| Logo | Click | Về màn hình HomeScreen | |
| Button Back | Click | Về màn hình HomeScreen | |

**Định nghĩa các trường thuộc tính:**

|  |  |
| --- | --- |
| **Screen Name** | Invoice Screen |
| **Attribute** | **Number of digits (bytes)** | **Type** | **Field Attribute** | **Remarks** |
| Mã xe | 10 | Numeral | Đen | Căn trái |
| Mã bãi xe | 10 | Numeral | Đen | Căn trái |
| Thời điểm bắt đầu thuê xe | 20 | Timestamp | Đen | Căn trái |
| Thời gian đã thuê cho tới nay | 20 | String | Đen | Căn trái, tự động tăng lên theo thời gian |

* + - 1. ***Return bike Screen:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ECO-BIKE SOFTWARE | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| Screen specification | Return Bike Screen | 07/01/2022 |  |  | Nguyễn Phi Phúc |
| **Graphical user interface, website  Description automatically generated** | | **Control** | **Operation** | **Function** | |
| Khu vực hiển thị thông tin của xe và thông tin về quá trình thuê xe (thời gian đã thuê, số tiền cần trả…) | Initial | Hiển thị thông tin xe và quá trình thuê xe, bên trái màn hình chính | |
| Khu vực hiển thị form thông tin về credit card | Initial | Hiển thị các trường thông tin về thẻ Credit card mà người dùng cần nhập: Số thẻ, tên người sở hữu, Ngày hết hạn, Mã bảo mật. | |
| Nút chọn phương thức thẻ | Click | Chọn phương thức thanh toán. Hiện tại bị disable vì chỉ có 1 phương thức. | |
| Nút xác nhận trả xe | Click | Gọi tới api interbank và chuyển tới màn hình Result Screen. | |
| Nút valid information | Click | Điền sẵn thông tin về card của nhóm vào form | |
| Nút Back | Click | Quay trở lại màn hình HomeScreen | |
| Logo | Click | Quay về màn hình HomeScreen | |

**Định nghĩa các trường thuộc tính:**

|  |  |
| --- | --- |
| **Screen Name** | Return Screen |
| **Attribute** | **Number of digits (bytes)** | **Type** | **Field Attribute** | **Remarks** |
| Card Holder Name | 50 | String | Đen | Căn trái |
| Card Number | 50 | String | Đen | Căn trái |
| CVV | 10 | String | Đen | Căn trái |
| Expiration Date | 10 | String | Đen | Căn trái |
| Mã số xe | 10 | Numeral | Đen | Căn trái |
| Tên xe | 50 | String | Đen | Căn trái |
| Trạng thái xe | 10 | Boolean | Đen | Căn trái |
| Biển số xe | 50 | String | Đen | Căn trái |
| **….** (Một số trường khác của thông tin xe) |  |  |  |  |
| Thời gian bắt đầu thuê xe | 50 | String | Đen | Căn trái |
| Tiền đặt cọc | 10 | Numeral | Đen | Căn trái |
| Thời gian đã thuê cho tới nay | 10 | String | Đen | Căn trái |

* + - 1. ***Dock detail Screen:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ECO-BIKE SOFTWARE | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| Screen specification | Dock Detail Screen | 07/01/2022 |  |  | Nguyễn Phi Phúc |
| **Graphical user interface  Description automatically generated** | | **Control** | **Operation** | **Function** | |
| Khu vực hiển thị logo, tên bãi xe, nút Back | Initial | Hiển thị logo, tên bãi xe, nút Back ở trên cùng màn hình | |
| Khu vực hiển thị thanh search xe và ảnh của bãi xe | Initial | Hiển thị thanh search và ảnh của bãi xe | |
| Khu vực hiển thị danh sách xe có trong bãi | Initial | Hiển thị danh sách xe có trong bãi | |
| Nút “Search” | Click | Tìm kiếm xe có thông tin được người dùng nhập | |
| Nút Option | Click and Select | Chọn đặc tính của xe mà người dùng muốn để search (theo tên/barcode/status hoặc category) | |
| Nút Back | Click | Quay trở lại màn hình HomeScreen | |
| Logo | Click | Quay về màn hình HomeScreen | |
| Table Row | Click | Click vào 1 hàng trong bảng để chọn xe tương ứng muốn xem và chuyển tới màn hình BikeDetailScreen | |

**Định nghĩa các trường thuộc tính:**

|  |  |
| --- | --- |
| **Screen Name** | Dock Detail Screen |
| **Attribute** | **Number of digits (bytes)** | **Type** | **Field Attribute** | **Remarks** |
| Thông tin tìm kiếm | 200 | String | Đen | Căn trái |
| Mã xe | 10 | Numeral | Đen | Căn giữa |
| Barcode | 10 | String | Đen | Căn giữa |
| Tên xe | 50 | String | Đen | Căn trái |
| Trạng thái | 10 | Boolean | Đen | Căn giữa |
| Thể loại | 50 | String | Đen | Căn trái |

* + - 1. ***Bike detail Screen:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ECO-BIKE SOFTWARE | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| Screen specification | Bike Detail Screen | 07/01/2022 |  |  | Nguyễn Phi Phúc |
| **Graphical user interface, application  Description automatically generated** | | **Control** | **Operation** | **Function** | |
| Khu vực hiển thị logo, tên xe, nút Back | Initial | Hiển thị logo, tên xe, nút Back ở trên cùng màn hình | |
| Khu vực hiển thị ảnh của xe | Initial | Hiển thị ảnh của xe | |
| Khu vực hiển thị thông tin của xe tương ứng | Initial | Hiển thị thông tin của xe tương ứng | |
| Nút Back | Click | Quay trở lại màn hình DockDetailScreen | |
| Logo | Click | Quay về màn hình HomeScreen | |

**Định nghĩa các trường thuộc tính:**

|  |  |
| --- | --- |
| **Screen Name** | Bike Detail Screen |
| **Attribute** | **Number of digits (bytes)** | **Type** | **Field Attribute** | **Remarks** |
| Mã xe | 10 | Numeral | Đen | Căn trái |
| Barcode | 10 | String | Đen | Căn trái |
| Tên xe | 50 | String | Đen | Căn trái |
| Trạng thái | 10 | Boolean | Đen | Căn trái |
| Thể loại | 50 | String | Đen | Căn trái |
| Giá khởi điểm | 10 | Numeral | Đen | Căn trái |
| Mô tả xe | 50 | String | Đen | Căn trái |
| Lượng PIN hiện tại | 10 | String | Đen | Căn trái |

### Data Modeling

* + 1. **Conceptual Data Modeling**

**Diagram

Description automatically generated**

**Figure 1: E-R Diagram**

* + 1. **Database Design**
       1. ***Database Management Systems***

Nhóm em sử dụng MySQL cho đề tài này, bởi vì có những sự liên kết giữa các thực thể với nhau.

* + - 1. ***Logical Data Model***

**Diagram

Description automatically generated**

**Figure 2: E-R Diagram**

**Diagram

Description automatically generated**

* + - 1. ***Physical Data Model***

**Figure 3: Data modeling**

**1. User:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **PK** | **FK** | **Column name** | **Data type** | **Mandatory** | **Description** |
| 1 | x |  | userId | Integer | Yes | ID người dùng, auto increase |
| 2 |  |  | userName | VARCHAR(255) | Yes | Tên người dùng |

**2. Dock:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **PK** | **FK** | **Column name** | **Data type** | **Mandatory** | **Description** |
| 1 | x |  | dockId | Integer | Yes | ID bãi xe, auto increase |
| 2 |  |  | dockName | VARCHAR(255) | Yes | Tên bãi xe |
| 3 |  |  | address | VARCHAR(255) | Yes | Địa chỉ bãi xe |
| 4 |  |  | dockArea | Float | Yes | Diện tích bãi xe |
| 5 |  |  | availableBikes | Integer | Yes | Số lượng xe còn sẵn trong bãi |
| 6 |  |  | emptyDockingPoints | Integer | Yes | Vị trị trống của từng loại xe |
| 7 |  |  | distance | Float | No | Khoảng cách người dùng tới bãi xe (Đơn vị: Km) |
| 8 |  |  | walkingTime | Float | No | Thời gian người dùng đi bộ tới bãi xe (Đơn vị: giờ) |
| 9 |  | x | imagePath | VARCHAR(255) | Yes | Ảnh minh hoạ bãi xe |

**3. Category:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **PK** | **FK** | **Column name** | **Data type** | **Mandatory** | **Description** |
| 1 | x |  | categoryId | Integer | Yes | ID loại xe, auto increase |
| 2 |  |  | categoryName | VARCHAR(255) | Yes | Tên loại xe, thuộc tập {“Xe đạp đơn, Xe đạp đôi, Xe đạp điện”} |
| 3 |  |  | bikeValue | Float | Yes | Giá trị loại xe, thuộc tập {400.000, 550.000, 700.000} |
| 4 |  |  | CategoryDescription |  | Yes |  |
| 5 |  |  | costRatio | Float | Yes | Tỉ giá, thuộc tập {1.0, 1.5, 1.5} |

**4. Bike:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **PK** | **FK** | **Column name** | **Data type** | **Mandatory** | **Description** |
| 1 | x |  | bikeId | Integer | Yes | ID xe, auto increase |
| 2 |  |  | bikeName | VARCHAR(255) | Yes | Tên xe |
| 3 |  |  | licensePlate | VARCHAR(255) | Yes | Biển số xe |
| 4 |  |  | pin | Float | No | Lượng pin hiện tại của xe (đối với xe đạp điện) |
| 5 |  |  | status | BIT | No | Trạng thái của xe |
| 6 |  |  | initCost | Float | Yes | Giá khởi điểm thuê trong 30 phút đầu |
| 7 |  |  | costPerQuarterHour | Float | Yes | Giá thuê thêm sau 30 phút đầu |
| 8 |  | x | dockId | Integer | Yes | Mã bãi xe |
| 9 |  | x | categoryId | Integer | Yes | Mã thể loại xe |
| 10 |  |  | imagePath | VARCHAR(255) | Yes | Ảnh minh hoạ xe |

**5. Rent:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **PK** | **FK** | **Column name** | **Data type** | **Mandatory** | **Description** |
| 1 | x |  | rentId | Integer | Yes | ID của 1 giao dịch thuê xe, auto increase |
| 2 |  |  | startTime | TIMESTAMP | Yes | Thời gian bắt đầu thuê |
| 3 |  |  | endTime | TIMESTAMP | No | Thời gian kết thúc thuê |
| 4 |  |  | debit | Float | No | Khoản phí thuê xe, đơn vị: nghìn VNĐ |
| 5 |  | x | userId | Integer | Yes | Mã người dùng |
| 6 |  | x | bikeId | Integer | Yes | Mã xe đang thuê |

**6. Transaction:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **PK** | | **FK** | | **Column name** | | **Data type** | | **Mandatory** | | **Description** | |
| 1 | x | |  | | transactionId | | Integer | | Yes | | Mã giao dịch, auto increase | |
| 2 |  | |  | | totalCost | | Float | | Yes | | Phí cần trả, đơn vị: nghìn VNĐ | |
| 3 |  | |  | | content | | LONG TEXT | | Yes | | Nội dung giao dịch | |
| 4 |  | |  | | rentedDuration | | Float | | No | | Thời gian thuê xe, đơn vị: Giờ | |
| 5 |  | |  | | createAt | | TIMESTAMP | | Yes | | Thời gian sửa đổi giao dịch | |
| 6 |  | | x | | userId | | Integer | | Yes | | Mã người dùng | |
| 7 | |  | | x | | bikeId | | Integer | | Yes | | Mã xe đã thuê |
| 8 | |  | | x | | rentId | | Integer | | Yes | | Mã giao dịch thuê |

### Non-Database Management System Files

Nếu trong trường hợp cần back up database thì nhóm sẽ export 1 file sql để có thể tái tạo database trên máy khác.

### Class Design

* + 1. **General Class Diagram**

**Diagram

Description automatically generated**

* + 1. **Class Diagrams**
       1. ***Class Diagram for views package***

**Diagram

Description automatically generated**

* + - 1. ***Class Diagram for controller package***

***Diagram

Description automatically generated***

* + - 1. ***Class Diagram for accessor package***

***Diagram

Description automatically generated***

* + - 1. ***Class Diagram for entity package***

***Diagram

Description automatically generated***

* + - 1. ***Class Diagram for subsystem Barcode***

Diagram

Description automatically generated

* + - 1. ***Class Diagram for subsystem Interbank***

**Diagram

Description automatically generated**

* + 1. **Class Design**
       1. ***Class “BikeDetailScreenHandler”***

***Diagram

Description automatically generated***

- Attribute :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| 1 | bike | Bike | NULL | Thông tin xe |

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | loadData | void | Hiển thị dữ liệu về xe |

- Parameter :

* bikeInfo : thông tin của xe

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “SplashScreenHandler”***

Diagram, table

Description automatically generated

- Attribute : không

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | initialize | void | Hiển thị màn hình splash screen |

- Parameter :

* location : đường dẫn dạng URL
* resources : tài nguyên dạng dữ liệu

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “HomeScreenHandler”***

Timeline

Description automatically generated with medium confidence

- Attribute :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| 1 | userId | int | NULL | id của người sử dụng |

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | initialize | void | khởi tạo màn hình |
| 2 | goDockDetailScreen | void | Xem danh sách bãi xe |
| 3 | loadDataToTable | void | Hiển thị dữ liệu dưới dạng bảng |
| 4 | goRentedBikeListScreen | void | Xem danh sách xe đang thuê |
| 5 | goRentBikeScreen | void | Chuyển sang màn hình thuê xe |

- Parameter :

* e : sự kiện người dùng thao tác trên giao diện
* dockList : danh sách bãi xe

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “BaseScreenHandler”***

Diagram, text

Description automatically generated

- Attribute : không

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | setPreviousScreen | void | đặt lại màn hình trước |
| 2 | getPreviousScreen | BaseScreenHandler | hiển thị màn hình trước |
| 3 | setScreenTittle | void | đặt tiêu đề cho màn hình |
| 4 | setBController | void | thay đổi base controller |
| 5 | getBController | BaseController | lấy ra base controller |
| 6 | setHomeScreen | void | quay về giao diện home |
| 7 | goPreviousScreen | void | quay về màn hình trước |
| 8 | setImage | void | thay đôỉ đường dẫn của các thành phần ảnh trên giao diện |
| 9 | convertCurrencyFormat | String | format định dạng hiển thị cho các giá trị về tiền |
| 10 | show | void | hiển thị |
| 11 | initiate | void | khởi tạo màn hình |

- Parameter :

* string : tiêu đề của giao diện
* prev : handler của màn hình trước
* bController : baseController cần đặt
* homeScreen : handler của giao diện home
* p : giá trị tiền cần format
* imgView : vị trí của ảnh cần thay đổi
* imgPath : đường dẫn tới ảnh

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “RentBikeScreenHandler”***

***Table

Description automatically generated***

- Attribute :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| 1 | bike | Bike | NULL | Thông tin về xe |
| 2 | userId | int | NULL | id của khách hàng |

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | loadData | void | hiển thị thông tin xe |
| 2 | confirmRentBike | void | bắt sự kiện khách hàng xác nhận thuê xe |

- Parameter :

* bike : thông tin xe
* e : sự kiện người dùng thao tác trên giao diện

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “ReturnBikeScreenHandler”***

***Table

Description automatically generated***

- Attribute :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| 1 | rent | Rent | NULL | Thông tin về quá trình thuê và sử dụng xe hiện tại |

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | loadData | void | Hiển thị dữ liệu về quá trình thuê xe |
| 2 | resetDefaultValue | void | Đặt lại giá trị mặc định |
| 3 | confirmReturnBike | void | bắt sự kiện người dùng xác nhận trả xe |

- Parameter :

* rent : thông tin về quá trình thuê xe
* e : sự kiện người dùng thao tác trên giao diện

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “CostComputer”***

Table

Description automatically generated

- Attribute : không có

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | checkout | float | tính tổng chi phí |
| 2 | getDebit | float | lấy giá cọc |

- Parameter :

* rent : thông tin về quá trình thuê xe
* endTime : thời điểm kết thúc thuê xe
* bike : thông tin xe

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “HomeController”***

***Table

Description automatically generated***

- Attribute :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| 1 | dockAccessor | DockAccessor | NULL | đối tượng giúp tương tác với data base |

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | getListDock | List<Dock> | Lấy danh sách bãi xe |
| 2 | searchDock | List<Dock> | Tìm kiếm bãi xe |

- Parameter :

* option : tuỳ chọn tìm kiếm
* info : thông tin mà người dùng cung cấp về bãi xe cần tìm

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “ReturnBikeController”***

***Table

Description automatically generated***

- Attribute :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| 1 | bikeAccessor | BikeAccessor | NULL | đối tượng dùng để tương tác với data base |
| 2 | interbank | InterbankInterface | NULL | dùng để tương tác với liên ngân hàng |
| 3 | costComputer | CostComputer | NULL | phục vụ việc tính tổng chi phí |
| 4 | transactionHistoryAccessor | TransactionHistoryAccessor | NULL | dùng để lưu lịch sử giao dịch |

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | calculateTotalCost | float | tính tổng số tiền người dùng phải trả |
| 2 | requestToReturnBike | void | gửi yêu cầu trả xe |
| 3 | makeTransactionHistory | TransactionHistory | lưu lại bản ghi về thông tin giao dịch |

- Parameter :

* rent : thông tin về quá trình thuê
* user : thông tin khách hàng
* card : thông tin thẻ tín dụng
* paymentTransaction : thông tin giao dịch
* endTime : thời điểm kết thúc thuê

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “RentBikeController”***

***Table

Description automatically generated***

- Attribute :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu | Giá trị mặc định | Mô tả |
| 1 | interbank | InterbankInterface | NULL | dùng để tương tác với liên ngân hàng |
| 2 | bikeAccessor | Bike | NULL | tương tác với CSDL lưu trữ thông tin xe |
| 3 | costComputer | costComputer | NULL | tính tiền cọc |
| 4 | barcodeProcess | barcodeInterface | NULL | tương tác với hệ thống chuyển đổi barcode |
| 5 | rentAccessor | RentAccessor | NULL | tương tác với data base lưu trữ thông tin các lần thuê |

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | requestBikeFromBarcode | int | Gửi yêu cầu lấy thông tin xe dựa vào barcode |
| 2 | requestRentBike | String | gửi yêu cầu thuê xe |

- Parameter :

* barcode : mã vạch của xe
* user : thông tin khách hàng
* bike : thông tin xe
* card : thông tin thẻ tín dụng

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “DataAccessorInterface”***

Text, letter

Description automatically generated

- Attribute : không có

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | getById | T | lấy dữ liệu dựa theo id |
| 2 | getAll | List<T> | lấy tất cả bản ghi trong cơ sở dữ liệu |
| 3 | update | void | cập nhật bản ghi trong cơ sở dữ liệu |
| 4 | delete | void | xoá bản ghi trong cơ sở dữ liệu |
| 5 | save | void | lưu lại bản ghi trong cơ sở dữ liệu |

- Parameter :

* id : id của đối tượng cần truy vấn

- Exception : không

- Method : không

- State : không

* + - 1. ***Class “BarcodeInterface”***

***Diagram

Description automatically generated***

- Attribute : không có

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | processBarcode | int | lấy id của xe theo barcode |

- Parameter :

* barcode : mã vạch của xe

- Exception :

* InvalidBarcodeException : nếu barcode không hợp lệ

- Method : không

- State : không

* + - 1. ***Class “InterbankInterface”***

***Diagram

Description automatically generated***

- Attribute : không có

- Operation:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Tên | Kiểu dữ liệu trả về | Mô tả |
| 1 | payRental | void | thực hiện giao dịch thuê xe |
| 2 | refund | void | thực hiện giao dịch trả xe |

- Parameter :

* card : thông tin thẻ
* amount : chi phí

- Exception :

* PaymentException : nếu mã lỗi trả về đã biết
* UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống

- Method : không

- State : không

# Design Considerations

### Goals and Guidelines

1. Mục tiêu

Phần mềm mô phỏng quá trình quản lý bãi xe của khu đô thị hoặc ở công cộng và cung cấp tiện ích thuê xe cho người dùng, với ứng dụng mô phỏng có thể chạy được đa nền tảng, thuận tiện cho người dùng trong việc tìm kiếm xe và bãi xe. Thiết kế mô hình với các thành phần độc lập với nhau, tức là tại một thời điểm không cần quá nhiều đối tượng phải tồn tại trong hệ thống, giúp tiết kiệm bộ nhớ và tăng thời gian xử lý.

1. Cách dùng

Không nên thuê xe quá lâu vì như vậy số tiền ảo mô phỏng sẽ không đáp ứng được. Không nên thuê qúa nhiều xe vì phải đặt cọc quá nhiều tiền, cũng có thể gây ra lỗi không đủ số dư trong tài khoản.

### Architectural Strategies

1. Hệ thống sử dụng ngôn ngữ lập trình Java, với thư viện JavaFx hỗ trợ đa nền tảng, có thể chạy trên nhiều hệ điều hành khác nhau.
2. Sử dụng hệ quản trị cơ sở dữ liệu mySQL, vì hệ thống yêu cầu tính đồng bộ cao giữa các bãi xe, trạng thái các xe, các giao dịch của người dùng, và dữ liệu của hệ thống cũng không phát sinh trở thành dữ liệu lớn trong tương lai đủ xa. (Ví dụ: Khoảng 10 năm)
3. Trong tương lai sẽ dự định phát triển thêm cả nền tảng web, do các controller độc lập với boundary nên có thể tận dụng lại được ở backend server.
4. Để khác phục lỗi xảy ra ở phía người dùng (ví dụ: hệ thống người dùng bị hỏng trong quá trình đang thuê xe), mọi thông tin của xe đều được lưu trên server và sẽ phát triển một nền tảng web để người dùng có thể trả xe dù điện thoại hay máy tính của họ bị hỏng.
5. Tại nhiều người cùng truy cập đến hệ thống, database sẽ có nhiều slave để đảm nhiểm chức năng đọc, còn việc ghi thì sẽ ghi vào một cụm database được kết nối băng thông với độ trệ thấp, thích hợp cho việc đồng bộ dữ liệu thời gian thực.
6. Các phương thức giao tiếp giữa người dùng và server là giao thức https với việc liên quan đến thông tin cá nhân user và http với việc truy nhập đến thông tin bike hoặc station.

### Coupling and Cohesion

Về tổng quan, chúng tôi đánh giá hệ thống là loose coupling và high cohesion.

* + 1. **Cohesion**
       1. ***views package***

Trong lớp HomeScreenHandler có nhiều hơn 1 phương thức khác khởi tạo, các phương thức này ở đây bởi vì nó đều là các chức năng chính hệ thống cung cấp cho người dùng. Vì vậy, Home Screen có mức cohesion là Logical.

Trong các lớp khác thì chỉ có một phương thức public duy nhất tương ứng với trách nhiệm của lớp ấy, nên nó có mức cohesion là Functional

* + - 1. ***controller package***

Trong lớp ReturnBikeController có 3 và trong lớp RentBikeCotroller có 2 phương thức. Hai lớp này điểu khiển luồng giao tiếp giữa người dùng, hệ thống bike, hệ thống ngân hang và hệ thống barcode, và các phương thức của từng lớp là một phần của chuỗi giao tiếp đó, nên hai lớp này có mức độ cohesion là Communicational

Trong lớp CostComputer có 2 phương thức là getDebit và calculate, đều là tính toán giá tiền nhưng mà cho hai trường hợp khác nhau, có tham số đầu vào khác nhau, nên mức độ Cohesion của lớp này là Logical.

* + - 1. ***accessor package***

Các lớp ở trong accessor packge của triển khai theo interface DataAccessorInterface có thiết kế theo mẫu Data Access Object, với 5 chức năng cơ bản của mẫu này. Do đó, mức cohesion của lớp này là Logical.

* + - 1. ***barcode package***

BarcodeInterface chỉ có 1 hàm duy nhất để xử lí barcode, nên nó có mức độ cohesion là Functional.

* + - 1. ***checkout package***

InterbankInterface cung cấp cho người dùng 2 phương thức xử lý cùng một kiểu dữ liệu, cùng giao tiếp với interbank boundary nhưng theo 2 cách khác nhau, và có entry point cho mỗi phương thức là khác nhau, nên chúng có mức độ cohesion là Communicational.

* + 1. **Coupling**
       1. ***views package***

Lớp BaseScreenHandler có nhiệm vụ chuyển giữa các màn hình ,và các tham số có thể ảnh hưởng đến luồng thực thi nên mức độ coupling là control coupling.

* + - 1. ***controller package***

Các lớp controller trong package này không giao tiếp với nhau nên chúng là độc lập. Lớp RentBikeController và ReturnBikeController giao tiếp với SimpleCostComputer thông qua interface CostComputer nên mức độ coupling là data coupling, tức là controller chỉ phụ thuộc vào SimpleCostComputer ở kiểu dữ liệu cho tham số các phương thức trong interface.

* + - 1. ***accessor package***

Trong accessor package, các accessor class không phụ thuộc vào nhau. Do đó chúng có mức độ coupling là data coupling.

* + - 1. ***barcode package***

Lớp barcode subsystem controller chỉ sử dụng phương thức của barcode boundary và giữa chúng không có tương tác làm thay đổi trạng thái lẫn nhau hay là việc thừa dữ liệu trong tham số trong message, nên mức độ coupling là data coupling.

* + - 1. ***checkout package***

Lớp interbank subsystem controller chỉ sử dụng phương thức của interbank boundary và giữa chúng không có tương tác làm thay đổi trạng thái lẫn nhau hay là việc thừa dữ liệu trong tham số trong message, nên mức độ coupling là data coupling.

### Design Principles

*Về tổng quan, các lớp trong hệ thống được thiết kế tuân theo nguyên lý SOLID.*

**Single Responsibility**

Các lớp trong hệ thống phần lớn đều chỉ thực hiện 1 chức năng duy nhất .

**Open for extension, close for modification**

Các lớp thực hiện thuật toán hay lớp thực hiện quy trình lấy dữ liệu đều là abstraction, nên khi muốn thay đổi thì ta chỉ cần thêm một lớp dẫn suất mới.

**Liskov substitution**

Các instance lớp con đều có thể thay thế được instance của 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. Ta có thể tách nó ra làm 2 interface để đảm bảo nguyên lý này, nhưng trong quá trình thiết kế, nhóm nhận thấy cả hai phương thức này đều cần thiết và được sử dụng trong controller class, và chúng ta có thể tạo ra lớp mới mà giữ nguyên phương thức calculate ở lớp cha và override lại phương thức getDebit ở lớp con, nên nhóm quyết định để 2 phương thức này vào cùng một interface để khi người dùng muốn tạo một CostComputerImplementer mới thì phải định nghĩa hoặc kế thừa cả phương thức trên.

**Dependency inversion**

Trong các sự phụ thuộc giữa package controller và các package khác đều là thông qua các interface. Các thành phần trong subsystem hay package cung cấp chức năng như datalayer, các lớp cụ thể sẽ triển khai (hay là phụ phục) và lớp trừu tượng (interface). Như vậy ta có thể dễ dàng thay đổi cách cài đặt của từng chức năng mà không ảnh hưởng đến các thành phần mà dùng nó.

### Design Patterns

*Trong thiết kế, nhóm có sử dụng một số design pattern sau.*

* + 1. **Factory patterns**

Factory pattern: Tách biệt công việc tạo object / instance ra khỏi xử lý của client code (client code là phần code sử dụng object / instance để thực hiện hoàn thành công việc của nó). Client code sẽ không cần phải quan tâm instance được tạo ra như thế nào, nó là object của class nào, nhờ đó giảm thiểu dependency của client code (Nguyên tắc: Loose Coupling).

* + 1. **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.

* + 1. **Singleton**

- Sử dụng Singleton cho class MySQLConnection để thể hiện tại 1 thời điểm chỉ có 1 đối tượng cổng quản lý kết nối tới CSDL.