Đại học Bách Khoa Hà Nội

Viện Công nghệ thông tin và Truyền thông

Software Design Document
version 1.0

Project: EcoBikeRental

Giảng viên hướng dẫn:
TS. Nguyễn Thị Thu Trang

Nhóm 10:
Đặng Lâm San - MSSV: 20170111
Nguyễn Mai Phương - MSSV: 20170106
Dương Hồng Sơn - MSSV: 20173347

Hà Nội - Ngày 20 tháng 12 năm 2020

Mục lục

1 Giới thiệu **4**
1.1 Mục tiêu . . . 4
1.2 Phạm vi . . . 4
1.3 Từ điển thuật ngữ . . . 4
1.4 Tài liệu tham khảo . . . 5

2 Mô tả tổng quan **5**
2.1 Mô tả chung . . . 5
2.2 Các giả định/ràng buộc/rủi ro . . . 5

3 Hệ thống Kiến trúc và thiết kế Kiến trúc (System Architecture and Architecture Design) **5**
3.1 Mô hình kiến trúc(Architectural Patterns) . . . 5
3.2 Biểu đồ tương tác . . . 6
3.3 Biểu đồ lớp phân tích . . . 10
3.4 Biểu đồ lớp phân tích kết hợp . . . 13
3.5 Kiến trúc phần mềm bảo mật . . . 14

4 Thiết kế chi tiết **14**
4.1 Thiết kế giao diện người dùng . . . 14
4.1.1 Chuẩn hóa cấu hình màn hình . . . 14
4.1.2 Display . . . 14
4.1.3 Screen . . . 14
4.1.4 Control . . . 14
4.1.5 Nhập input từ bàn phím . . . 15
4.1.6 Error . . . 15
4.1.7 Sơ đồ chuyển đổi màn hình . . . 15
4.1.8 Mô tả các màn hình . . . 15
4.2 Thiết kế subsystem . . . 27
4.2.1 Thiết kế subsystem hệ thống thanh toán . . . 27
4.2.2 Thiết kế subsystem hệ thống chuyển đổi mã vạch . . . 30

5 Mô hình hóa dữ liệu **32**
5.1 Mô hình hóa khái niệm dữ liệu . . . 33
5.2 Thiết kế cơ sở dữ liệu . . . 34
5.2.1 Mô hình dữ liệu logic . . . 34
5.2.2 Mô hình dữ liệu vật lý . . . 34
5.3 Tệp hệ thống quản lý phi cơ sở dữ liệu . . . 39
5.4 Thiết kế lớp . . . 39
5.4.1 Biểu đồ lớp . . . 40
5.4.2 Thiết kế chi tiết lớp . . . 46

6 Design Considerations **68**
6.1 Mục tiêu và Nguyên tắc 68
6.2 Chiến lược kiến trúc (Architectural Strategies) 68
6.3 Coupling and Cohesion 68
6.4 Các nguyên tắc thiết kế (Design Principles) 68
6.4.1 Single Responsibility Principle 69
6.4.2 Open/Closed Principle 69
6.4.3 Liskov substitution principle 69
6.4.4 Interface segregation principle 69
6.4.5 Dependency Inversion principle 70
6.5 Design Patterns 70
6.5.1 Singleton 70
6.5.2 DAO - Data Access Object pattern 72

1 Giới thiệu

1.1 Mục tiêu

Tài liệu này đưa ra thiết kế chi tiết về kiến trúc trong phần mềm, các hệ thống ngoài cũng như các chuẩn thiết kế để hoàn thiện các chức năng đã được mô tả trong tài liệu SRS. Tài liệu mô tả chi tiết các lớp có trong chương trình phần mềm, cơ sở dữ liệu, các màn hình, các ràng buộc của hệ thống. Tài liệu dành cho các bên liên quan và đặc biệt là các nhà phát triển phần mềm.

1.2 Phạm vi

Khu đô thị Ecopark có dịch vụ cho thuê xe đạp theo giờ với nhiều bãi để xe để thuê/trả xe tự động trong khu đô thị. Để quản lý việc thuê/trả xe một cách tự động, trong tài liệu đặc tả phần mềm này, chúng tôi trình bày về phần mềm EcoBikeRental cho phép thực hiện những nhiệm vụ trên.

Mục đích của ứng dụng là để quản lý việc thuê/trả xe điện trong khu đô thị. Khách hàng có thể sử dụng ứng dụng để xem thông tin về các bãi đỗ xe (địa chỉ, diện tích, số lượng xe còn lại,...) cũng như thông tin về các xe đang đỗ trong bãi đỗ xe đó (lượng pin, thời gian tối đa tương ứng có thể sử dụng được,...). Để thuê xe, phần mềm cần cung cấp chức năng quét mã vạch của xe, sau khi quét, phầm mềm sẽ cung cấp thông tin về chiếc xe đó và cho phép khách hàng thuê. Ngoài ra, để hoàn tất thủ tục, phần mềm yêu cầu khách hàng phải đặt cọc một lượng tiền tùy theo thông tin được quét từ mã vạch của mỗi chiếc xe. Phần mềm thu tiền từ mỗi khách hàng thông qua thẻ tín dụng mà từng khách hàng cung cấp cho phần mềm. Khi trả xe, khách hàng cần thực hiện lại việc quét mã vạch và thanh toán tiền. Phần mềm sẽ thực hiện công đoạn thanh toán, trả lại tiền cọc vào đúng thẻ tín dụng mà khách hàng đã sử dụng trước đó khi thuê xe.

Khi bãi đỗ xe ngày càng được xây dựng nhiều trong đô thị, nếu sử dụng nhân lực để điều hành mỗi bãi đỗ xe thì sẽ rất tốn kém về chi phí. Hơn nữa, khách hàng cũng không thể biết lượng xe còn lại ở mỗi bãi đỗ xe để có thể đưa ra phương án hợp lý để đi đến bãi đỗ xe cho mục đích thuê xe. Phần mềm EcoBikeRental cho phép khách hàng, nhà đầu tư khắc phục những tồn tại trên, cắt giảm chi phí nhân công cũng như cung cấp dịch vụ tối ưu cho khách hàng có thể dễ dàng sử dụng hệ thống thuê/trả xe đạp điện.

1.3 Từ điển thuật ngữ

Thuật ngữ	Chú giải
Thẻ tín dụng	là loại thẻ cho phép chủ thẻ được sử dụng trong hạn mức tín dụng tuần hoàn được cấp mà chủ thẻ phải thanh toán ít nhất mức trả nợ tối thiểu vào ngày đến hạn. [1]
Mã vạch	là sự thể hiện thông tin trong các dạng nhìn thấy trên các bề mặt của sản phẩm, hàng hóa mà máy móc có thể đọc được [2]

1.4 Tài liệu tham khảo

[1] TS.Nguyễn Thị Thu Trang *SDD-Template-EN*, v1.2.
[2] TS.Nguyễn Thị Thu Trang *Bài giảng môn phân tích thiết kế và phát triển phần mềm.*
[3] Kathy Sierra, Elisabeth Freeman, *Head First Design Pattern.*
[4] Wikipedia, *Mã vạch .*
[5] Wikipedia, *Thẻ tín dụng.*

2 Mô tả tổng quan

2.1 Mô tả chung

Phần mềm được thiết kế theo thiết kế 3 tầng, dùng trên desktop. Thiết kế này không thích hợp với người dùng trong thực tế vì không ai mang PC hay laptop đi để thuê xe đạp. Bên cạnh đó, laptop hay PC cũng k có chức năng quét barcode hay có mạng liên tục. Tuy nhiên, với khuôn khổ của bài tập, phầm mềm được thiết kế phù hợp để demo.

2.2 Các giả định/ràng buộc/rủi ro

Không có

3 Hệ thống Kiến trúc và thiết kế Kiến trúc (System Architecture and Architecture Design)

3.1 Mô hình kiến trúc(Architectural Patterns)

Nhóm chọn thiết kế theo kiến trúc 3 tầng

3.2 Biểu đồ tương tác

Hình 1: Xem chi tiết bãi xe

Hình 2: Xem chi tiết xe

Hình 3: Xem xe đang thuê

Hình 4: Chuyển đổi barcode

Hình 5: Trả xe

Hình 6: Trả tiền cọc

Hình 7: Thuê xe

3.3 Biểu đồ lớp phân tích

Hình 8: Xem chi tiết bãi xe

Hình 9: Xem chi tiết xe

Hình 10: Xem xe đang thuê

Hình 11: Thuê xe

Hình 12: Đặt cọc

Hình 13: Trả xe

Hình 14: Chuyển mã vạch

3.4 Biểu đồ lớp phân tích kết hợp

Hình 15: Biểu đồ lớp phân tích kết hợp

3.5 Kiến trúc phần mềm bảo mật

Không

4 Thiết kế chi tiết

4.1 Thiết kế giao diện người dùng

4.1.1 Chuẩn hóa cấu hình màn hình

4.1.2 Display

- Số lượng màu được hỗ trợ: 16,777,216 màu
- Độ phân giải: 1366 x 768 pixels

4.1.3 Screen

- Vị trí của của button: Ở giữa (theo chiều dọc) và ở bên phải (theo chiều ngang) của khung.
- 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.

4.1.4 Control

- Kích thước text: medium size (24px).
- Font: Segoe UI.
- Color: 000000
- Xử lý check input: Nên 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 splash screen (màn hình chớp) sẽ được hiện lên và sau đó màn hình đầu tiên(Home Screen) sẽ xuất hiện
- Thứ tự các màn hình trong hệ thống:
    1. splash screen (first screen)
    2. home screen - màn hình chính

3. dock list – xem danh sách các bãi xe
4. bike list - xem chi tiết danh sách xe trong bãi xe
5. bike detail - xem thông tin chi tiết xe trong bãi
6. bike form renting – điền thông tin thuê xe
7. rented detail - xem thông tin chi tiết xe vừa thuê thành công
8. renting detail - xem thông tin xe đang thuê
9. alert - pop up xác nhận thuê xe
10. barcode - pop up điền mã xe muốn thuê để thuê xe

4.1.5 Nhập input từ bàn phím

Sẽ không có phím tắt. Có các button quay lại "Back" để quay lại các màn hình trước đó. Ngoài ra button “X” nằm ở thanh tiêu đề bên phải để đóng screen. Riêng các pop up sẽ có thêm button "close" để đóng.

4.1.6 Error

Một thông điệp sẽ được hiện lên để thông báo cho người dùng biết vấn đề đang gặp phải là gì.

4.1.7 Sơ đồ chuyển đổi màn hình

Sơ đồ chuyển đổi màn hình

Hình 16: Sơ đồ chuyển màn hình

4.1.8 Mô tả các màn hình

a. splash screen

Hình 17: splash screen

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	splash screen	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Khu vực hiển thị màn hình khởi động cho phần mềm		Initial	Hiển thị màn hình khởi động cho phần mềm để chờ tải thông tin màn hình chính cũng như thông tin người dùng		

Bảng 2: Mô tả màn hình splash screen

b.home screen - màn hình chính

Hình 18: home screen

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	home screen	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Khu vực hiển thị thông tin hệ thống		Initial	Hiển thị thông tin hệ thống		
Thông tin liên hệ		Initial	Hiển thị liên hệ bao gồm hotline, email để người dùng có thể liên hệ tới tư vấn viên của hệ thống		
Nút bấm tìm hiểu thêm		Click	Hiển thị thông tin chi tiết về hệ thống và hướng dẫn sử dụng hệ thống		
Nút bấm View Dock List		Click	Hiển thị thông tin về danh sách bãi xe		
Nút bấm View Renting Bike		Click	Hiển thị thông tin về xe đang thuê		

Bảng 3: Mô tả màn hình home screen

c.dock list – xem danh sách các bãi xe

Hình 19: dock list

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	dock list	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Khu vực hiển thị thông tin tổng quát các bãi xe		Initial	Hiển thị thông tin về các bãi xe hiện có bao gồm số lượng bãi, số lượng xe, số lượng đơn, số lượng khách hàng		
Thông tin liên hệ		Initial	Hiển thị liên hệ bao gồm hotline, email để người dùng có thể liên hệ tới tư vấn viên của hệ thống		
Khu vực hiển thi danh sách bãi xe		Initial	Hiển thị thông tin tổng quát về bãi sẽ bao gồm tên, vị trí, số lương xe, trạng thái		
Nút bấm Choose a random dock		Click	Chọn random một bãi xe trong danh sách và hiển thị thông tin chi tiết bãi xe		
Nút bấm Back		Click	Quay trở lại màn hình trước		

Nút bấm Open	Click	Hiển thị thông tin chi tiết bãi xe được chọn

Bảng 4: Mô tả màn hình dock list

d.bike list - xem chi tiết danh sách xe trong bãi xe

Hình 20: bike list

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	bike list	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Khu vực hiển thị thông tin tổng quát các bãi xe		Initial	Hiển thị thông tin về các bãi xe hiện có bao gồm số lượng bãi, số lượng xe, số lượng đơn, số lượng khách hàng		
Thông tin liên hệ		Initial	Hiển thị liên hệ bao gồm hotline, email để người dùng có thể liên hệ tới tư vấn viên của hệ thống		
Khu vực hiển thi danh sách xe trong		Initial	Hiển thị thông tin tổng quát về xe bao gồm mã xe, bãi xe, pin (nếu có), trạng thái		
Nút bấm Rent a bike		Click	Khi người dùng chọn một xe trong danh sách và bấm nút, hiển thị màn hình bike form renting. Nếu người dùng chưa chọn xe thì sẽ hiển thị pop up barcode để người dùng nhập mã xe muốn thuê.		

Nút bấm Back	Click	Quay trở lại màn hình trước
Nút bấm Available	Click	Khi nút bấm trạng thái xe hiển thị available, khách hàng có thể chọn để xem thông tin chi tiết về xe, hiển thị màn hình bike detail
Nút bấm chọn danh sách xe	Click	Chọn loại danh sách xe muốn xem

Bảng 5: Mô tả màn hình bike list

e.bike detail - xem thông tin chi tiết

Hình 21: bike detail

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	home screen	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Khu vực hiển thị thông tin chi tiết về xe		Initial	Hiển thị thông tin tin chi tiết về xe được chọn		
Nút bấm Close		Click	Đóng pop up		

Bảng 6: Mô tả pop up bike detail

f.bike form renting – điền thông tin thuê xe

Hình 22: bike form renting

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	bike form renting	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Khu vực hiển thị thông tin đơn hàng		Initial	Hiển thị thông tin đơn hàng bao gồm loại xe, giá		
Khu vực điền tên khách hàng		TextField	Cho phép khách hàng điền tên		
Khu vực điền thông tin thẻ		TextField	Cho phép khách hàng điền thông tin thẻ		
Nút bấm Confirm		Click	Hiển thị pop up xác nhận thuê xe		

Bảng 7: Mô tả pop up bike form renting

g.rented detail - xem thông tin chi tiết xe vừa thuê thành công

Hình 23: rented detail

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	rented detail	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Khu vực hiển thị thông tin đơn hàng		Initial	Hiển thị thông tin đơn hàng		
Nút bấm Close		Click	Tắt pop up, trở lại home screen		

Bảng 8: Mô tả pop up rented detail

h.renting detail - xem thông tin xe đang thuê

Hình 24: bike renting detail

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	bike renting detail	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Khu vực hiển thị thông tin đơn hàng		Initial	Hiển thị thông tin xe và đơn hàng đang thuê		
Nút bấm Stop		Click	Tạm dừng thuê xe, khóa xe		
Nút bấm Return		Click	Trả xe		

Bảng 9: Mô tả pop up bike renting detail

i.alert - pop up xác nhận thuê xe

Hình 25: alert

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	alert	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Nút bấm Yes		Click	Xác nhận thuê xe, trả xe, khóa xe, tắt pop up, trở về màn hình chính		
Nút bấm No		Click	Tắt pop up		

Bảng 10: Mô tả pop up alert

j.barcode - pop up điền mã xe muốn thuê để thuê xe

Hình 26: barcode

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	barcode	1/12/2020			Đặng Lâm San
Control		Operation	Function		
Khu vực điền mã xe		TextField	Cho phép khách hàng điền mã xe muốn thuê		
Nút bấm Confirm		Click	Xác nhận muốn thuê xe, tắt pop upbarcode, hiển thị pop up bike form rentin		

Bảng 11: Mô tả pop up barcode

4.2 Thiết kế subsystem

4.2.1 Thiết kế subsystem hệ thống thanh toán

Hệ thống sử dụng Interbank thông qua InterbankInterface giao tiếp với InterbankSystem

Trong đó cấu trúc của subsystem như sau:

Phụ thuộc Subsystem với các thành phần của hệ thống:

Sau đây là biểu đồ nghiệp vụ của hệ thống thanh toán Interbank

Hình 27: Interface for subsystem

Hình 28: Document Subsystem Elements

Hình 29: Describe Subsystem Dependencies

Hình 30: Pay / Deposit

Hình 31: Refurn money

4.2.2 Thiết kế subsystem hệ thống chuyển đổi mã vạch

Hệ thống sử dụng BarcodeConverter thông qua BarcodeInterface giao tiếp với BarcodeSystem

Hình 32: Interface for subsystem

Trong đó cấu trúc của subsystem như sau:

Phụ thuộc Subsystem với các thành phần của hệ thống:

Sau đây là biểu đồ nghiệp vụ của hệ thống thanh toán Barcode

Hình 33: Document Subsystem Elements

Hình 34: Describe Subsystem Dependencies

Hình 35: Exchange barcode

5 Mô hình hóa dữ liệu

5.1 Mô hình hóa khái niệm dữ liệu

Mô hình hóa khái niệm dữ liệu như sau:

Hình 36: Mô hình khái niệm dữ liệu

5.2 Thiết kế cơ sở dữ liệu

Chúng ta sử dụng DBMS là MySQL để thiết kế cơ sở dữ liệu Từ mô hình khái niệm (được thể hiện bằng biểu đồ ER) trong phần trước,, ta có thể thiết kế mô hình dữ liệu logic tương thích với DBMS đã chọn (MySQL)

5.2.1 Mô hình dữ liệu logic

Hình 37: Mô hình dữ liệu logic

5.2.2 Mô hình dữ liệu vật lý

Trong phần này, chúng ta sẽ thiết kế chi tiết cho từng phần tử trong biểu đồ trên.
Ví dụ, trong biểu đồ cơ sở dữ liệu quan hệ (RDBMS), ta thiết kế chi tiết cho từng bảng và ràng buộc.
Chú thích:
PK: Primary Key

FK: Foreign Key

- Bike

	PK	FK	Column name	Data Type	Mandatory	Description
1.	x		id	Integer	Yes	ID
2.			type	Integer	Yes	Loại xe
3.			pin	Integer	Yes	Dung lượng pin
4.			value	Integer	Yes	Giá trị của xe
5.		x	dockId	Integer	Yes	Bãi xe
6.			barcode	VARCHAR(45)	Yes	Mã vạch của xe
7.			bikecol	Integer	Yes	

Bảng 12: Bảng mô tả dữ liệu Bike

- Dock

	PK	FK	Column name	Data Type	Mandatory	Description
1.	x		id	Integer	Yes	ID
2.			name	VARCHAR(45)	Yes	Tên bãi xe
3.			address	VARCHAR(145)	Yes	Địa chỉ bãi xe

Bảng 13: Bảng mô tả dữ liệu Dock

- User

	PK	FK	Column name	Data Type	Mandatory	Description
1.	x		id	Integer	Yes	ID
2.			name	VARCHAR(45)	Yes	Tên khách hàng
3.			card	VARCHAR(45)	Yes	Thẻ thanh toán

Bảng 14: Bảng mô tả dữ liệu User

-Transaction

	PK	FK	Column name	Data Type	Mandatory	Description
1.	x		id	Integer	Yes	ID

2.		x	bikeId	Integer	Yes	ID của bike
3.		x	userId	Integer	Yes	ID của user
4.			status	Integer	Yes	Trạng thái
5.			total	Integer	Yes	Số tiền của giao dịch
6.			date	DATETIME	Yes	Thời gian giao dịch
7.			time	Integer	Yes	Thời gian thuê xe

Bảng 15: Bảng mô tả dữ liệu Transaction

Database Script

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `rent_bike` /*!40100 DEFAULT CHARACTER SET utf8 C

USE `rent_bike`;
--
-- Table structure for table `bike`
--

DROP TABLE IF EXISTS `bike`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `bike` (
  `id` int NOT NULL,
  `type` int DEFAULT NULL,
  `pin` int DEFAULT NULL,
  `value` int DEFAULT NULL,
  `dockId` int DEFAULT NULL,
  `barcode` varchar(45) COLLATE utf8_bin DEFAULT NULL,
  `bikecol` varchar(45) COLLATE utf8_bin DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `bikecol_UNIQUE` (`bikecol`),
  KEY `dockId_idx` (`dockId`),
  CONSTRAINT `dockId` FOREIGN KEY (`dockId`) REFERENCES `dock` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `dock`
--

DROP TABLE IF EXISTS `dock`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `dock` (
  `id` int NOT NULL,
  `name` varchar(45) COLLATE utf8_bin DEFAULT NULL,
  `address` varchar(145) COLLATE utf8_bin DEFAULT NULL,
```

```
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client = @saved_cs_client */;


--
-- Table structure for table `transaction`
--

DROP TABLE IF EXISTS `transaction`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `transaction` (
  `id` int unsigned NOT NULL AUTO_INCREMENT,
  `bikeId` int DEFAULT NULL,
  `userId` int DEFAULT NULL,
  `status` int DEFAULT NULL,
  `total` int DEFAULT NULL,
  `date` datetime DEFAULT NULL,
  `time` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `id_UNIQUE` (`id`),
  KEY `userId_idx` (`userId`),
  KEY `bikeId` (`bikeId`),
  CONSTRAINT `bikeId` FOREIGN KEY (`bikeId`) REFERENCES `bike` (`id`),
  CONSTRAINT `userId` FOREIGN KEY (`userId`) REFERENCES `user` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client = @saved_cs_client */;

DROP TABLE IF EXISTS `user`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `user` (
  `id` int NOT NULL,
  `name` varchar(45) COLLATE utf8_bin DEFAULT NULL,
  `card` varchar(45) COLLATE utf8_bin DEFAULT NULL,
```

```
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
/*!40101 SET character_set_client = @saved_cs_client */;
```

5.3 Tệp hệ thống quản lý phi cơ sở dữ liệu

Không

5.4 Thiết kế lớp

5.4.1 Biểu đồ lớp

Hình 38: Biểu đồ các lớp quan hệ với package controller

Hình 39: Biểu đồ các lớp quan hệ với package database

Hình 40: Biểu đồ các lớp quan hệ với package exception

Hình 41: Biểu đồ các lớp quan hệ với package sample

Hình 42: Biểu đồ các lớp quan hệ với package bank subsystem

Hình 43: Biểu đồ các lớp quan hệ với package barcode subsystem

5.4.2 Thiết kế chi tiết lớp

Các lớp trong package controller

- DockInfoController
- DockListController
- PaymentController
- RentingBikeController
- ReturnBikeController

Các lớp trong package database

- MySQLDriver
- Dao

Các lớp trong package entity

- package mysqlDAO chứ: BikeDAO, DockDao, UserDAO, TransactionDAO.
- Bike
- Dock
- User
- Transaction
- PaymentTransaction
- Card

Các lớp trong package exception

- InternalServerErrorException
- InvalidCardException
- InvalidTransactionAmountException
- InvalidVersionException
- NotEnoughBalanceException
- NotEnoughTransactionInfoException
- PaymentException
- PaymentException
- UnrecognizedException

Các lớp trong package sample: Các lớp biên
Các lớp trong package banksubsystem:

- package interbank gồm: InterbankBoundary, InterbankSubsystemController
- InterbankInterface
- InterbankSubsystem

Các lớp trong package barcodesubsystem:

- package interbank gồm: BarcodeBoundary, BarcodeSubsystemController
- BarcodeInterface
- BarcodeSubsystem

a. Lớp "DockInfoController"

Hình 44: Lớp "DockInfoController"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	dock	Dock	null	Đại diện cho cho bãi xe
2	bikeDao	BikeDao	null	Đại diện cho cho lớp thực thể BikeDao
3	bikes	List<Bike>	null	Đại diện cho cho danh sách các xe trong bãi

Bảng 16: Attribute lớp "DockInfoController"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	getBikes	List<Bike>	Lấy danh sách các xe có trong bãi
2	getNumBike	int	Lấy số lượng xe thuộc tất cả các loại có trong bãi
3	getBikesByType	List<Bike>	Lấy số lượng xe với loại được truyền vào
4	getBikeByBarcode	Bike	Lấy ra xe với barcode được truyền vào
5	getDock	Dock	Lấy ra bãi xe hiện tại

Bảng 17: Operation lớp "DockInfoController"

Parameter:

- type : int, id của kiểu xe muốn lấy ra danh sách
- barcode: String, id của xe

Exception: không
Method: không
State: không
b. Lớp "DockListController"

Hình 45: Lớp "DockListController"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	docks	List<Dock>	null	Đại diện cho danh sách bãi xe trong hệ thống
ư	dockDao	DockDao	null	Đại diện cho lớp thực thể DockDao

Bảng 18: Attribute lớp "DockListController"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	getDocks	List<Dock>	Lấy danh sách các bãi xe có trong hệ thống

Bảng 19: Operation lớp "DockListController"

Parameter: không
Exception: không
Method: không

State: không

c. Lớp "PaymentController"

Hình 46: Lớp "PaymentController"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	interbankSubsystem	InterbankSubsystem	null	Đại diện cho InterbankSubsystem

Bảng 20: Attribute lớp "PaymentController"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	pay	PaymentTransaction	Thanh toán cho giao dịch thuê xe
2	refund	PaymentTransaction	Hoàn tiền cọc cho người thuê xe khi trả xe

Bảng 21: Operation lớp "PaymentController"

Parameter:

- transaction: Transaction, giao dịch cần thanh toán hoặc hoàn trả tiền cọc

Exception:

- PaymentException: nếu mã lỗi trả về đã biết

Method: không
State: không

d. Lớp "RentingBikeoController"

Attribute:

Hình 47: Lớp "RentingBikeController"

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	startTime	long	null	Đại diện cho thời gian bắt đầu thuê xe
2	barcodeSubsystem	BarcodeSubsystem	null	Đại diện cho BarcodeSubsystem
3	dockInfoController	DockInfoController	null	Đại diện cho DockInfoController
4	paymentController	PaymentController	null	Đại diện cho PaymentController
5	returnBikeController	ReturnBikeController	null	Đại diện cho ReturnBikeController

Bảng 22: Attribute lớp "RentingBikeController"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	setTransaction	Transaction	Tạo mới giao dịch thuê xe
2	Transaction	getTransaction	Lấy giao dịch thuê xe hiện tại
3	startTransaction	void	Bắt đầu thuê xe
4	getTimeUsing	long	Lấy thời gian đã thuê xe hiện tại
5	getTotal	int	Lấy số tiền phải trả ở hiện tại
6	requestBarcode	Bike	Lấy xe đang thuê the barcode

Bảng 23: Operation lớp "RentingBikeController"

Parameter:

- transaction: Transaction, giao dịch muốn thanh toán thuê xe name: String, tên người thuê xe
- card: String, thông tin thanh toán
- bike: Bike, xe muốn thuê
- barcode: String, bracode xe đang thuê

Exception:

- PaymentException: nếu mã lỗi trả về đã biết

Method: không
State: không

e. Lớp "ReturnBikeController"

Hình 48: Lớp "ReturnBikeController"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả

1	paymentController	PaymentController	null	Đại diện cho PaymentController

Bảng 24: Attribute lớp "ReturnBikeController"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	refund	void	Trả tiền cọc cho khách hàng khi khách trả xe
2	makeTransactionDao	void	Lưu lịch sử các giao dịch thuê xe và đặt chế độ available cho xe được trả ở trong bãi

Bảng 25: Operation lớp "RentingBikeController"

Parameter:

- transaction: Transaction, giao dịch đã hoàn thành dock: Dock, bãi xe trả xe
- total: int, tổng tiền cần hoàn trả

Exception:

- PaymentException: nếu mã lỗi trả về đã biết

Method: không
State: không

f. Lớp "InterbankSubsystemController"

Hình 49: Lớp "InterbankSubsystemController"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	interbankBoundary	InterbankBoundary	null	Đại diện cho InterbankBoundary

Bảng 26: Attribute lớp "InterbankSubsystemController"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	pay	PaymentTransaction	Xử lí thanh toán cho giao dịch thuê xe
2	refund	PaymentTransaction	Xử lí trả tiền cọc cho giao dịch thuê xe
3	extractPaymentTransaction	PaymentTransaction	Trả lại PaymentTransaction của xe đang block
4	reset	String	Tiếp tục thuê xe

Bảng 27: Operation lớp "InterbankSubsystemController"

Parameter:

- card: Card, thông tin thẻ khách hàng thanh toán amount: int, số lượng tiền cần thanh toán
- contents: String, thông tin giao dịch dưới dạng String Json
- respond:String, .

Exception:

- PaymentException: nếu mã lỗi trả về đã biết

Method: không
State: không
g. Lớp "InterbankInterface"

Hình 50: Lớp "InterbankInterface"

Attribute: không
Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	pay	PaymentTransaction	Xử lí thanh toán cho giao dịch thuê xe
2	refund	PaymentTransaction	Xử lí trả tiền cọc cho giao dịch thuê xe

Bảng 28: Operation lớp "InterbankInterface"

Parameter:

- card: Card, thông tin thẻ khách hàng thanh toán amount: int, số lượng tiền cần thanh toán
- contents: String, thông tin giao dịch dưới dạng String Json

Exception: không
Method: không
State: không

h. Lớp "InterbankSubsystem"

Hình 51: Lớp "InterbankSubsystem"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	ctrl	InterbankSubsystemController	null	Đại diện cho InterbankSubsystemController

Bảng 29: Attribute lớp "InterbankSubsystem"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	pay	PaymentTransaction	Xử lí thanh toán cho giao dịch thuê xe
2	refund	PaymentTransaction	Xử lí trả tiền cọc cho giao dịch thuê xe

Bảng 30: Operation lớp "InterbankSubsystem"

Parameter:

- transaction: Transaction, giao dịch đã hoàn thành dock: Dock, bãi xe trả xe.
- total: int, tổng tiền cần hoàn trả

Exception:

- PaymentException: nếu mã lỗi trả về đã biết

Method: không
State: không

i. Lớp "BarcodeSubsystem"

Hình 52: Lớp "BarcodeSubsystem"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	barcodeController	BarcodeController	null	Đại diện cho BarcodeController

Bảng 31: Attribute lớp "BarcodeSubsystem"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	exchangeBarcode	String	Trả về chuỗi tương ứng với barcode

Bảng 32: Operation lớp "BarcodeSubsystem"

Parameter: không
Exception: không
Method: không
State: không

j. Lớp "BarcodeInterface"

Hình 53: Lớp "BarcodeInterface"

Attribute: không
Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	exchangeBarcode	String	Trả về chuỗi tương ứng với barcode

Bảng 33: Operation lớp "BarcodeInterface"

Parameter: không
Exception: không
Method: không
State: không
k. Lớp "BarcodeController"

Hình 54: Lớp "BarcodeController"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	barcodeBoundary	BarcodeBoundary	null	Đại diện cho BarcodeBoundary

Bảng 34: Attribute lớp "BarcodeController"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	exchangeBarcode	String	Trả về chuỗi tương ứng với barcode

Bảng 35: Operation lớp "BarcodeController"

Parameter: không
Exception: không
Method: không

State: không

l. Lớp "MySQLDriver" Chịu trách nhiệm giao tiếp chính với database

Hình 55: Lớp "MySQLDriver"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	url	String	null	Đại diện cho địa chỉ đến local host của database
2	dbName	String	null	Đại diện cho tên database
3	driver	String	null	Đại diện cho db driver
4	username	String	null	Đại diện cho username để truy cập db
5	password	String	null	Đại diện cho password để truy cập db

Bảng 36: Attribute lớp "MySQLDriver"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)

1	getDriverConnection	MySQLDriver	Trả vể thực thể MySQLDriver
2	query	ResultSet	Thực hiện query với database
3	insert	int	Thêm vào database
4	update	void	Cập nhật vào database

Bảng 37: Operation lớp "MySQLDriver"

Parameter: không
Exception: không
Method: không
State: không

m. Lớp "Bike" Lớp thực thể đại diện cho thực thể xe

Hình 56: Lớp "Bike"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	id	int	null	Đại diện cho mã xe
2	type	int	null	Đại diện cho loại xe
3	pin	int	null	Đại diện cho lượng in hiện tại của xe (nếu có)
4	value	int	null	Đại diện cho giá của xe
5	dockId	int	null	Đại diện cho mã của bãi xe đặt xe đó
6	barcode	String	null	Đại diện cho barcode của xe

Bảng 38: Attribute lớp "Bike"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	toString	String	Lấy String Json các thuộc tính của xe đó

Bảng 39: Operation lớp "Bike"

Parameter: không
Exception: không
Method: không
State: không
n. Lớp "Dock"
Lớp thực thể đại diện cho bãi xe

Hình 57: Lớp "Dock"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	id	int	null	Đại diện cho mã bãi xe
2	type	String	null	Đại diện cho tên bãi xe
3	address	String	null	Đại diện cho địa chỉ bãi xe

Bảng 40: Attribute lớp "Dock"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	toString	String	Lấy String Json thông tin xe

Bảng 41: Operation lớp "Dock"

Parameter: không
Exception: không
Method: không
State: không

o. Lớp "Card"

Lớp thực thể đại diện cho thẻ

Hình 58: Lớp "Card"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	id	String	null	Đại diện cho mã thẻ
2	owner	String	null	Đại diện tên chủ sở hữu thẻ

3	ccv	String	null	Đại diện cho ccv
4	dateExpire	String	null	Đại diện cho hạn dùng của thẻ
5	amount	int	null	Đại diện cho số dư thẻ

Bảng 42: Attribute lớp "Card"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	getInstance	Card	Trả vể thực thể Card đó

Bảng 43: Operation lớp "Card"

Parameter: không
Exception: không
Method: không
State: không

p. Lớp "User"
Lớp thực thể đại diện cho khách hàng

Hình 59: Lớp "User"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	id	int	null	Đại diện cho mã định danh người dùng
2	name	String	null	Đại diện cho tên người dùng
3	card	String	null	Đại diện cho db thẻ người dùng sử dụng

Bảng 44: Attribute lớp "User"

Operation: không
Parameter: không
Exception: không
Method: không
State: không

q. Lớp "Transaction"

Lớp thực thể đại diện cho giao dịch thuê xe

Hình 60: Lớp "Transaction"

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	userId	int	null	Đại diện cho mã khách hàng thuê xe
2	bikeId	int	null	Đại diện cho xe được chọn thuê
3	id	int	null	Đại diện cho mã đơn thuê xe
4	status	int	null	Đại diện cho trạng thái đơn thuê xe
5	total	int	null	Đại diện cho tổng số tiền hiện tại cần thanh toán

6	date	String	null	Đại diện cho thời gian thuê xe
7	totalTime	int	null	Đại diện cho tổng thời gian đã thuê xe

Bảng 45: Attribute lớp "Transaction"

Operation:

STT	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	toString	String	Trả về String Json mô tả các thuộc tính của transaction

Bảng 46: Operation lớp "Transaction"

Parameter: không
Exception: không
Method: không
State: không
r. Lớp "PaymentTransaction"
Lớp thực thể đại diện cho giao dịch thanh toán hoặc hoàn tiền

Attribute:

STT	Tên	Kiểu dữ liệu trả về	Giá trị mặc định	Mô tả
1	errorCode	String	null	Mã lỗi khi thực hiện
2	card	Card	null	Đại diện cho thẻ thanh toán giao dịch này
3	transactionId	String	null	Đại diện cho giao dịch được thanh toán
4	transactionContent	String	null	Đại diện cho loại giao dịch
5	amount	int	null	Đại diện cho tổng số tiền cần xử lí
6	createdAt	String	null	Đại diện cho thời gian thuê xe
7	totalTime	String	null	Đại diện cho thời gian tạo xử lí

Bảng 47: Attribute lớp "PaymentTransaction"

Hình 61: Lớp "PaymentTransaction"

Operation: không
Parameter: không
Exception: không
Method: không
State: không

6 Design Considerations

6.1 Mục tiêu và Nguyên tắc

6.2 Chiến lược kiến trúc (Architectural Strategies)

Chúng tôi chọn làm phần mềm theo kiến trúc 3 tầng

6.3 Coupling and Cohesion

Trong hệ thống này, tính kết dính - cohesion khá cao. Các module trong hệ thống được tách theo các vai trò mà nó quản lý, không có lớp nào mang nhiều hơn hai trách nhiệm nghiệp vụ. Cụ thể ví dụ như lớp ReturnBikeController chỉ có một phương thức là refund() và một phương thức phụ trợ việc lưu giao dịch vào cơ sở dữ liệu là makeTransactionDao().
Bên cạnh đó, hệ thống chưa đạt được loose coupling. Các thành phần còn phụ thuộc vào nhau khá nhiều, ngoài ra một số module còn quản lý chung một số dữ liệu (Data Global) như PaymentTransaction. Cụ thể, RentingBikeController và ReturnBikeController đều có quyền chỉnh sửa trạng thái của lớp Singleton PaymentTransaction.

6.4 Các nguyên tắc thiết kế (Design Principles)

Thiết kế của hệ thống đã cố gắng để tuân theo nguyên tắc trong SOLID. Tuy nhiên, do điều kiện kiến thức cũng như kỹ năng thực tế còn thiếu, nên chúng tôi phát triển và thiết kế

phần mềm này cũng mắc phải một số lỗi. Các phần phía dưới bàn luận các thiết kế của hệ thống theo 5 nguyên tắc của SOLID.

6.4.1 Single Responsibility Principle

Trách nhiệm của hệ thống được phân bổ tới các package, các subsystem và trong mỗi package, subsystem, trách nhiệm được chia nhỏ cho từng Class, mỗi Class đảm nhận một trách nhiệm duy nhất.

Tuy nhiên, ta xem xét thử một lớp InterbankSubsystemController chịu trách nhiệm cho 2 nhiệm vụ: (1) điều khiển luồng dữ liệu (thanh toán, hoàn tiền), (2) chuyển đổi dữ liệu (chuyển đổi dữ liệu nhận về từ api sang dạng controller yêu cầu, hàm extractPaymentTransaction). Do đó, lớp này phải được thay đổi khi mà luồng dữ liệu thay đổi (ví dụ các tính năng mới được thêm vào) hoặc cách chuyển đổi dữ liệu thay đổi (ví dụ như định dạng payment transaction thay đổi), có lẽ bản thiết kế nên được thay đổi để tốt hơn.

6.4.2 Open/Closed Principle

Barcode subsystem implement các phương thức được định nghĩa trong Barcode interface. Các lớp của hệ thống chỉ phụ thuộc vào Barcode Interface chứ không phụ thuộc trực tiếp vào Barcode subsystem. Do đó, có thể dễ dàng thay thế subsystem sẵn có bằng một subsystem khác hoặc thêm một số phương thức khác cho Barcode interface và implement các phương thức này trong subsystem. Các thay đổi bên phía subsystem hoàn toàn trong suốt với các bên liên quan sử dụng giao diện của Barcode interface.

Tương tự với Interbank subsystem. Interbank subsystem implement các phương thức được định nghĩa trong Interbank interface. Các lớp của hệ thống chỉ phụ thuộc vào Interbank Interface chứ không phụ thuộc trực tiếp vào Interbank subsystem. Do đó, có thể dễ dàng thay thế subsystem sẵn có bằng một subsystem khác hoặc thêm một số phương thức khác cho Interbank interface và implement các phương thức này trong subsystem. Các thay đổi bên phía subsystem hoàn toàn trong suốt với các bên liên quan sử dụng giao diện của Interbank interface.

6.4.3 Liskov substitution principle

Không có

6.4.4 Interface segregation principle

Nguyên lý này nói rằng, thay vì một sử dụng một interface quá lớn, quá nhiều phương thức thì chúng ta sẽ tách nhỏ ra thành các interface con với mục đích cụ thể. Vì khi để một interface quá to, các lớp implement sẽ phải implement các phương thức mà bản thân nó không cần dùng đến.

Thiết kế hiện tại về cơ bản đã đáp ứng được nguyên tắc này, ví dụ với InterbankInterface, cả 2 phương thức payOrder và refund đều được lớp InterfaceSystemController implement. Với BarcodeInterface cũng tương tự.

Tuy nhiên, cũng cần phải cân nhắc một chút với InterfaceSystemController , bởi trong tương lai, có thể có một số hệ thống interbank khác không hoàn tiền cho khách hàng mà chỉ thanh toán, lúc này phương thức refund của InterbankInterface trở nên dư thừa đối với interbanksubsystem đó è vi phạm Interface Segregation.

6.4.5 Dependency Inversion principle

Có thể hiểu nguyên lý này như sau: những thành phần trong một chương trình chỉ nên phụ thuộc vào những cái trừu tượng. Những thành phần trừu tượng không nên phụ thuộc vào một thành phần mang tính cụ thể mà nên ngược lại. Hiện tại, PaymentTransaction đang phụ thuộc chặt chẽ vào lớp Card, sau này giả sử không sử dụng Card để thanh toán mà sử dụng một loại phương thức thanh toán khác, ví dụ như domestic debit card... như vậy thiết kế hiện tại đã vi phạm nguyên lý D trong SOLID.

6.5 Design Patterns

Thiết kế áp dụng hai design pattern là Singleton pattern và DAO pattern

6.5.1 Singleton

Thiết kế áp dụng Singleton pattern cho lớp Transaction, Card và MySQLDriver.

Với lớp Transaction, áp dụng thiết kế như vậy nhằm mục đích với mỗi khách hàng, cùng một thời điểm chỉ được đặt một xe. Điều này áp dụng tốt trong thực tế vì mỗi khách chỉ có thể dùng 1 xe tại một thời điểm, trách gây mất mát tài sản vì trong tương lai, hệ thống sẽ có thể có thểm chức nắng theo dõi vị trí của khách hàng khi thuê xe qua app điện thoại.

Với lớp Card, áp dụng Singleton pattern để mỗi khách hàng chỉ có thể dùng 1 thể để thanh toán. Thiết kế này nếu áp dụng vào thực tế thì sẽ không hợp lí vì khách có thể có nhiều thẻ thanh toán khác nhau. Tuy nhiên, với phần mềm demo ở hiện tại, thiết kế này được coi là khả thi.

Bên cạnh đó, lớp MySQLDriver chịu trách nhiệm quản lí liên kết với server database. Áp dụng pattern này cho MySQLDriver nhằm giúp hệ thống hoạt động tránh gặp lỗi hay xung đột nếu nhà phát triển chẳng may tạo nhiều thực thể MySQLDriver khác nhau tại nhiều vị trí trong phần mềm.

Hình 62: Transaction Singleton Pattern

Hình 63: Card Singleton Pattern

Hình 64: MySQLDriver Singleton Pattern

6.5.2 DAO - Data Access Object pattern

Data Access Object (DAO) Pattern là một trong những Pattern thuộc nhóm cấu trúc (Structural Pattern). Mẫu thiết kế DAO được sử dụng để phân tách logic lưu trữ dữ liệu trong một lớp riêng biệt. Theo cách này, các service được che dấu về cách các hoạt động cấp thấp để truy cập cơ sở dữ liệu được thực hiện. Nó còn được gọi là nguyên tắc Tách logic (Separation of Logic).

Lớp Interface DAO là một interface định nghĩa các phương thức trừu tượng việc triển khai truy cập dữ liệu cơ bản cho BusinessObject để cho phép truy cập vào nguồn dữ liệu (DataSource).

DockDAO, BikeDAO, TransactionDAO, UserDao cài đặt các phương thức được định nghĩa trong DAO, lớp này sẽ thao tác trực tiếp với nguồn dữ liệu (DataSource).

Hình 65: MySQLDriver Singleton Pattern

Danh sách hình vẽ

1 Xem chi tiết bãi xe . . . 6
2 Xem chi tiết xe . . . 7
3 Xem xe đang thuê . . . 7
4 Chuyển đổi barcode . . . 8
5 Trả xe . . . 8
6 Trả tiền cọc . . . 9
7 Thuê xe . . . 9
8 Xem chi tiết bãi xe . . . 10
9 Xem chi tiết xe . . . 10
10 Xem xe đang thuê . . . 11
11 Thuê xe . . . 11
12 Đặt cọc . . . 12
13 Trả xe . . . 12
14 Chuyển mã vạch . . . 13
15 Biểu đồ lớp phân tích kết hợp . . . 13
16 Sơ đồ chuyển màn hình . . . 15
17 splash screen . . . 16
18 home screen . . . 17
19 dock list . . . 18
20 bike list . . . 20
21 bike detail . . . 22
22 bike form renting . . . 23
23 rented detail . . . 24
24 bike renting detail . . . 25
25 alert . . . 26
26 barcode . . . 27
27 Interface for subsystem . . . 28
28 Document Subsystem Elements . . . 28
29 Describe Subsystem Dependencies . . . 29
30 Pay / Deposit . . . 29
31 Refurn money . . . 30
32 Interface for subsystem . . . 30
33 Document Subsystem Elements . . . 31
34 Describe Subsystem Dependencies . . . 31
35 Exchange barcode . . . 32
36 Mô hình khái niệm dữ liệu . . . 33
37 Mô hình dữ liệu logic . . . 34
38 Biểu đồ các lớp quan hệ với package controller . . . 41
39 Biểu đồ các lớp quan hệ với package database . . . 42
40 Biểu đồ các lớp quan hệ với package exception . . . 43
41 Biểu đồ các lớp quan hệ với package sample . . . 44
42 Biểu đồ các lớp quan hệ với package bank subsystem . . . 45
43 Biểu đồ các lớp quan hệ với package barcode subsystem . . . 45

44 Lớp "DockInfoController" 48
45 Lớp "DockListController" 49
46 Lớp "PaymentController" 50
47 Lớp "RentingBikeController" 51
48 Lớp "ReturnBikeController" 52
49 Lớp "InterbankSubsystemController" 54
50 Lớp "InterbankInterface" 55
51 Lớp "InterbankSubsystem" 56
52 Lớp "BarcodeSubsystem" 57
53 Lớp "BarcodeInterface" 57
54 Lớp "BarcodeController" 58
55 Lớp "MySQLDriver" 59
56 Lớp "Bike" 61
57 Lớp "Dock" 62
58 Lớp "Card" 63
59 Lớp "User" 65
60 Lớp "Transaction" 66
61 Lớp "PaymentTransaction" 68
62 Transaction Singleton Pattern 71
63 Card Singleton Pattern 71
64 MySQLDriver Singleton Pattern 72
65 MySQLDriver Singleton Pattern 72

Danh sách bảng

2 Mô tả màn hình splash screen . . . 16
3 Mô tả màn hình home screen . . . 17
4 Mô tả màn hình dock list . . . 19
5 Mô tả màn hình bike list . . . 21
6 Mô tả pop up bike detail . . . 22
7 Mô tả pop up bike form renting . . . 23
8 Mô tả pop up rented detail . . . 24
9 Mô tả pop up bike renting detail . . . 25
10 Mô tả pop up alert . . . 26
11 Mô tả pop up barcode . . . 27
12 Bảng mô tả dữ liệu Bike . . . 35
13 Bảng mô tả dữ liệu Dock . . . 35
14 Bảng mô tả dữ liệu User . . . 35
15 Bảng mô tả dữ liệu Transaction . . . 36
16 Attribute lớp "DockInfoController" . . . 48
17 Operation lớp "DockInfoController" . . . 48
18 Attribute lớp "DockListController" . . . 49
19 Operation lớp "DockListController" . . . 49
20 Attribute lớp "PaymentController" . . . 50
21 Operation lớp "PaymentController" . . . 50
22 Attribute lớp "RentingBikeController" . . . 51
23 Operation lớp "RentingBikeController" . . . 52
24 Attribute lớp "ReturnBikeController" . . . 53
25 Operation lớp "RentingBikeController" . . . 53
26 Attribute lớp "InterbankSubsystemController" . . . 54
27 Operation lớp "InterbankSubsystemController" . . . 54
28 Operation lớp "InterbankInterface" . . . 55
29 Attribute lớp "InterbankSubsystem" . . . 56
30 Operation lớp "InterbankSubsystem" . . . 56
31 Attribute lớp "BarcodeSubsystem" . . . 57
32 Operation lớp "BarcodeSubsystem" . . . 57
33 Operation lớp "BarcodeInterface" . . . 58
34 Attribute lớp "BarcodeController" . . . 58
35 Operation lớp "BarcodeController" . . . 58
36 Attribute lớp "MySQLDriver" . . . 59
37 Operation lớp "MySQLDriver" . . . 60
38 Attribute lớp "Bike" . . . 61
39 Operation lớp "Bike" . . . 62
40 Attribute lớp "Dock" . . . 62
41 Operation lớp "Dock" . . . 63
42 Attribute lớp "Card" . . . 64
43 Operation lớp "Card" . . . 64
44 Attribute lớp "User" . . . 65

45 Attribute lớp "Transaction" 67
46 Operation lớp "Transaction" 67
47 Attribute lớp "PaymentTransaction" 67