Contents

[1 Class Design 3](#_Toc88390769)

[1.1 General Class Diagram 3](#_Toc88390770)

[1.2 Class Diagrams 3](#_Toc88390771)

[1.2.1 Class Diagram for Package A 3](#_Toc88390772)

[1.2.2 Class Diagram for Subsystem B 3](#_Toc88390773)

[1.3 Class Design 3](#_Toc88390774)

[1.3.1 Class “InterbankInterface” 3](#_Toc88390775)

[1.3.2 Class “PaymentController” 4](#_Toc88390776)

[1.3.3 Class “PlaceOrderController” 5](#_Toc88390777)

[1.3.4 Class “ViewCartController” 6](#_Toc88390778)

[1.3.5 Class “SplashScreenHandler” 7](#_Toc88390779)

[1.3.6 Class “App” 7](#_Toc88390780)

[1.3.7 Class “FXMLScreenHandler” 8](#_Toc88390781)

[1.3.8 Class “BaseScreenHandler” 9](#_Toc88390782)

[1.3.9 Class “InterbankSystemController” 10](#_Toc88390783)

[1.3.10 Class “CartScreenHandler” 11](#_Toc88390784)

[1.3.11 Class “ShippingScreenHandler” 12](#_Toc88390785)

[1.3.12 Class “InvoiceScreenHandler” 13](#_Toc88390786)

[1.3.13 Class “PaymentScreenHandler” 14](#_Toc88390787)

[1.3.14 Class “Cart” 15](#_Toc88390788)

[2 Mô hình hóa dữ liệu (Data modeling) 16](#_Toc88390789)

[2.1 Mô hình khái niệm (Conceptual Data Model) 16](#_Toc88390790)

[2.2 Thiết kế cơ sở dữ liệu 17](#_Toc88390791)

[2.2.1 Mô hình dữ liệu logic (Logical data model) 17](#_Toc88390792)

[*2.2.2* Mô hình dữ liệu vật lý (Physical data model) 17](#_Toc88390793)

[3 BTVN 27](#_Toc88390794)

[3.1 Thiết kế lớp chi tiết cho usecase “Place Rush Order” 27](#_Toc88390795)

[3.1.1 Class “Cart” 27](#_Toc88390796)

[3.1.2 Class “PlaceRushOrderController” 27](#_Toc88390797)

[3.1.3 Class “RushShippingScreenHandler” 28](#_Toc88390798)

[3.1.4 Class “LocationRushShipping” 29](#_Toc88390799)

[3.2 Mô hình hóa dữ liệu cho usecase “PlaceRushOrder” 30](#_Toc88390800)

[3.2.1 Mô hình khái niệm (Conceptual Data Model) 30](#_Toc88390801)

[3.3 Thiết kế cơ sở dữ liệu 31](#_Toc88390802)

[3.3.1 Mô hình dữ liệu logic (Logical data model) 31](#_Toc88390803)

[3.3.2 Mô hình dữ liệu vật lý. 31](#_Toc88390804)

# Class Design

## General Class Diagram

## Class Diagrams

### Class Diagram for Package A

### Class Diagram for Subsystem B

…

## Class Design

### Class “InterbankInterface”

Table

Description automatically generated

**Attribute**

Không

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | payOrder | PaymentTransaction | Thanh toán đơn hàng và trả về giao dịch thanh toán |
| 2 | refund | PaymentTracsaction | Hoàn tiền và trả về giao dịch thanh toán |

*Parameter*:

* card – thẻ tín dụng để giao dịch
* amount – số tiền giao dịch
* contents - nội dung giao dịch

*Exception*:

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

**Method**

Không

**State**

Không

### Class “PaymentController”

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | Card | CreditCard | NULL | Represent the card used for payment |
| 2 | interbank | InterbankInterdace | NULL | Represent the InterbankInterface |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | payOrder | Map<String,String> | Thanh toán đơn hàng và trả về giao dịch thanh toán |

*Parameter*:

* amount – số tiền giao dịch
* contents - nội dung giao dịch
* cardNumber – Số thẻ
* cardHolderName – tên chủ sở hữu thẻ
* exprirationDate – Ngày hết hạn thẻ thẻ theo định dạng mm/yy
* securityCode – mã bảo mật cvv/cvc

*Exception*:

Không

**Method**

* getExpirationDate: Chuyển dữ liệu ngày từ định dạng “mm/yy” sang “mmyy”

Diagram

Description automatically generated

**State**

Không

### Class “PlaceOrderController”

Diagram

Description automatically generated with medium confidence

**Attribute**

Không

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | placeOrder | void | Đặt đơn hàng |
| 2 | createOrder | Order | Tạo đơn hàng |
| 3 | processDeliveryInfo | Void | Thêm thông tin giao hàng |

*Parameter*:

* Info – thông tin giao hàng

*Exception*:

Không

**Method**

* validateDeliveryInfo – kiểm tra thông tin giao hàng
* calculateShippingFee – tính tiền vận chuyển (phí giao hàng)

**State**

Không

### Class “ViewCartController”

Diagram

Description automatically generated

**Attribute**

Không

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | checkAvailiablityOfProduct | void | Kiểm tra số lượng sản phẩm |

*Parameter*:

* Không

*Exception*:

* Không

**Method**

Không

**State**

Không

### Class “SplashScreenHandler”

A picture containing diagram

Description automatically generated

**Attribute**

Không

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | Initialize(location:URL, resources:ResourcesBundle) | void | Thanh toán đơn hàng và trả về giao dịch thanh toán |

*Parameter*:

* Location –
* Resources -

*Exception*:

Không

**Method**

Không

**State**

Không

### Class “App”

A picture containing diagram

Description automatically generated

**Attribute**

Không

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | start | void | Khởi chợi stage |
| 2 | main | void | Chạy chương trình |

*Parameter*:

* primaryStage : giao diện bắt đầu chương trình
* arg : tham số biến đầu vào

*Exception*:

            Không

**Method**

Không

**State**

Không

### Class “FXMLScreenHandler”

Text

Description automatically generated

**Attribute**

Không

**Operation**

  Không

*Parameter*:

Không

*Exception*:

            Không

**Method**

* getContent – lấy nội dung trong AnchorPane
* getLoader –
* setImage – đặt ảnh cho Screen

**State**

Không

### Class “BaseScreenHandler”

Table

Description automatically generated with medium confidence

**Attribute**

**Operation**

*Parameter*:

Không

*Exception*:

Không

**Method**

* setPreviousScreen – thiết lập screen trước đó
* getPreviousScreen – lấy ra screen trước đó
* show – hiển thị screen
* setScreenTitle – thiết lập title cho screen
* setBController – thiết lập controller cho screen
* getBController – lấy ra controller
* setHomeScreen – thiết lập screen cho màn hình

**State**

Không

### Class “InterbankSystemController”

Table

Description automatically generated with medium confidence

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | PUBLIC\_KEY | String | NULL |  |
| 2 | interbank | InterbankInterdace | NULL | Represent the InterbankInterface |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | payOrder | PaymentTransaction | Thanh toán đơn hàng và trả về giao dịch thanh toán |
| 2 | refund | PaymentTransaction | Hoàn tiền và trả về giao dịch thanh toán |
| 3 | getBalance | Int | Trả về số dư tài khoản |

*Parameter*:

-        card – thẻ giao dịch

- amount – số tiền giao dịch

-        contents - nội dung giao dịch

-        cardNumber – Số thẻ

*Exception*:

* InternalServiceErrorException : lỗi hệ thống
* NotEnoughBalanceException : lỗi số dư không đủ
* InvalidCardException : lỗi thẻ không tồn tại.

**Method**

-        generateURL – sinh đường dẫn URL

**State**

Không

### Class “CartScreenHandler”

A picture containing table

Description automatically generated

**Attribute**

Không

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | requestToPlaceOrder | void | Yêu cầu đặt hàng |
| 2 | displayCartWithProductAvailablity | void | Hiển thị giao diện giỏ hàng với sản phẩm có sẵn |

*Parameter*:

Không

*Exception*:

* Không

**Method**

-        generateURL – sinh đường dẫn URL

**State**

Không

### Class “ShippingScreenHandler”

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | shipment | Shipment | NULL | Represent the Shipment |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | initialize | void | Khởi tạo màn hình điền thông tin giao hàng |
| 2 | submitDeliveryInfo | void | Xác nhận thêm thông tin giao hàng |

*Parameter*:

* Shipment – thông tin giao hàng

*Exception*:

* InvalidDeliveryInfoException – thông tin giao hàng không hợp lệ

**Method**

-        notifyError : thông báo lỗi xảy ra, thông tin giao hàng không đúng

**State**

Không

### Class “InvoiceScreenHandler”

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | invoice | Invoice | NULL | Represent the Invoice |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | display | void | Hiển thị màn hình hóa đơn |
| 2 | confirmInvoice | void | Xác nhận hóa đơn thanh toán |
| 3 | requestToPayOrder | void | Yêu cầu thanh toán hóa đơn |

*Parameter*:

-        invoice – hóa đơn mua hàng

*Exception*:

Không

**Method**

* displaySuccessOrder – hiển thị thông báo đặt hàng thành công

**State**

Không

### Class “PaymentScreenHandler”

Text

Description automatically generated

**Attribute**

Không

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | confirmToPayOrder | void | Xác nhận thanh toán |
| 2 | requestToPayOrder | void | Yêu cầu thanh toán |

*Parameter*:

-        invoice – thông tin hóa đơn thanh toán đơn hàng

*Exception*:

* InternalServiceErrorException : lỗi hệ thống
* NotEnoughBalanceException : lỗi số dư không đủ
* InvalidCardException : lỗi thẻ không tồn tại.

**Method**

-        Không

**State**

Không

### Class “Cart”

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | CartMediaList | CartMedia | NULL | Danh sách các sản phẩm có trong giỏ hàng |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | checkProductAvailiblity | void | Kiểm tra lượng sản phẩm có trong giỏ hàng |
| 2 | empty | void | Xóa hết các sản phẩm trong giỏ hàng |

*Parameter*:

-        Không

*Exception*:

Không

**Method**

Không

**State**

Không

# Mô hình hóa dữ liệu (Data modeling)

## Mô hình khái niệm (Conceptual Data Model)

**AISM SYSTEM ERD**

Shape

Description automatically generated

## Thiết kế cơ sở dữ liệu

### **Mô hình dữ liệu logic (Logical data model)**

***Diagram

Description automatically generated***

### Mô hình dữ liệu vật lý (Physical data model)

**Chú thích:**

* PK: primary key
* FK: foreign key
* **Media**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, auto increment |
| 2 |  |  | category | Varchar(45) | Yes | Media type, eg. DVD, CD |
| 3 |  |  | price | Integer | Yes | Current price |
| 4 |  |  | quantity | Integer | Yes | Number of products |
| 5 |  |  | title | Varchar(45) | Yes | Product name |
| 6 |  |  | value | Integer | Yes | Value of the product |
| 7 |  |  | imageUrl | Varchar(45) | Yes | Product image path |

* **CD**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, sane as ID of Media of which type is CD |
| 2 |  |  | artist | Varchar(45) | Yes | Artist’s name |
| 3 |  |  | recordLabel | Varchar(45) | Yes | Record label |
| 4 |  |  | musicType | Varchar(45) | Yes | Music genres |
| 5 |  |  | releaseDate | Date | No | Release date |

* **Book**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, same as ID of Media of which type is Book |
| 2 |  |  | author | Varchar(45) | Yes | Author |
| 3 |  |  | coverType | Varchar(45) | Yes | Cover Type |
| 4 |  |  | publisher | Varchar(45) | Yes | Publishing house |
| 5 |  |  | publishDate | Datetime | Yes | Date of publishing |
| 6 |  |  | numOfPages | Integer | Yes | Page number |
| 7 |  |  | language | Varchar(45) | Yes | Language |
| 8 |  |  | bookCategory | Varchar(45) | Yes | Book category |

* **DVD**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID,same as ID of media of which type is DVD |
| 2 |  |  | discType | Varchar(45) | Yes | Disc Type |
| 3 |  |  | Director | Varchar(45) | Yes | Director |
| 4 |  |  | runtime | Integer | Yes | Duration |
| 5 |  |  | studio | Varchar(45) | Yes | Manufacturer |
| 6 |  |  | subtitle | Varchar(45) | Yes | Subtitle |
| 7 |  |  | releasedDate | Datetime | Yes | Release Date |
| 8 |  |  | firmType | Varchar(45) | Yes | Genres |

* **LP**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, same as ID of Media of which type is LP |
| 2 |  |  | artist | Varchar(45) | Yes | Artist’s name |
| 3 |  |  | recordLabel | Varchar(45) | Yes | Record label |
| 4 |  |  | musicType | Varchar(45) | Yes | Music genres |
| 5 |  |  | releaseDate | Date | No | Release date |

* **Card**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, auto increment |
| 2 |  |  | cardCode | Varchar(45) | Yes | Card code |
| 3 |  |  | Owner | Varchar(45) | Yes | Card holders |
| 4 |  |  | cvvCode | Varchar(45) | Yes | CVV code |
| 5 |  |  | dateExpired | Varchar(4) | Yes | Expiration date |

* **DeliveryInfo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, auto increment |
| 2 |  |  | name | Varchar(45) | Yes | Receiver name |
| 3 |  |  | Province | Varchar(45) | Yes | Provinces |
| 4 |  |  | Instructions | Varcher(200) | No | Delivery intraction |
| 5 |  |  | address | Varchar(100) | Yes | Delivey address |

* **Order**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, auto increment |
| 2 |  |  | shippingFees | Varchar(45) | Yes | Shipping fee |
| 3 |  | x | deliveryInfold | Integer | Yes | Delivey Info ID |

* **OrderMedia**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | MediaID | Integer | Yes | MediaID |
| 2 |  |  | orderId | Integer | Yes | OrderId |
| 3 |  |  | price | Integer | Yes | Selling price |
| 4 |  |  | quantity | Integer | Yes | Number |

* **Invoice**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID |
| 2 |  |  | totalAmount | Integer | Yes | Total |
| 3 |  |  | orderId | Integer | Yes | OrderId |

* **PaymentTransaction**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, auto increment |
| 2 |  |  | createdAt | Datetime | Yes | Date of creation |
| 3 |  |  | content | Varchar(45) | Yes | Transaction Contents |
| 4 |  |  | method | Varchar(45) | Yes | Payment methods |
| 5 |  | x | cardId | Integer | Yes | ID off usser card |
| 6 |  | x | invoiceId | Integer | Yes | InvoiceId |

**Tạo database AIMS**

Ở phần này em sử dụng MYSQL để tạo cơ sở dữ liệu

#Tạo media

CREATE TABLE Media(

id int PRIMARY KEY auto\_increment NOT NULL,

category VARCHAR(45) NOT NULL,

price INTEGER NOT NULL,

quantity INTEGER NOT NULL,

title VARCHAR(45) NOT NULL,

value INTEGER NOT NULL,

imageUrl VARCHAR(45) NOT NULL

);

#Tạo CD

CREATE TABLE CD(

id INTEGER PRIMARY KEY NOT NULL,

artist VARCHAR(45) NOT NULL,

recordLabel VARCHAR(45) NOT NULL,

musicType VARCHAR(45) NOT NULL,

releasedDate DATE,

CONSTRAINT fk\_CD\_Media1

FOREIGN KEY(id)

REFERENCES Media(id)

);

#Tạo Book

CREATE TABLE Book(

id INTEGER PRIMARY KEY NOT NULL,

author VARCHAR(45) NOT NULL,

coverType VARCHAR(45) NOT NULL,

publisher VARCHAR(45) NOT NULL,

publishDate DATETIME NOT NULL,

numOfPages INTEGER NOT NULL,

language VARCHAR(45) NOT NULL,

bookCategory VARCHAR(45) NOT NULL,

CONSTRAINT fk\_Book\_Media1

FOREIGN KEY(id)

REFERENCES Media(id)

);

#Tạo DeliveryInfo

CREATE TABLE DeleveryInfo(

id INTEGER PRIMARY KEY auto\_increment NOT NULL,

name VARCHAR(45),

province VARCHAR(45),

instructions VARCHAR(200),

address VARCHAR(100)

);

#Tạo card

CREATE TABLE Card(

id INTEGER PRIMARY KEY auto\_increment NOT NULL,

cardCode VARCHAR(15) NOT NULL,

owner VARCHAR(45) NOT NULL,

cvvCode VARCHAR(3) NOT NULL,

dateExpired VARCHAR(4) NOT NULL

);

#Tạo DVD

CREATE TABLE DVD(

id INTEGER PRIMARY KEY auto\_increment NOT NULL,

discType VARCHAR(45) NOT NULL,

director VARCHAR(45) NOT NULL,

runtime INTEGER NOT NULL,

studio VARCHAR(45) NOT NULL,

subtitle VARCHAR(45) NOT NULL,

releasedDate DATETIME,

CONSTRAINT fk\_DVD\_Media1

FOREIGN KEY(id)

REFERENCES Media(id)

)

#Tạo order

CREATE TABLE Orders(

id INTEGER NOT NULL,

shippingFees VARCHAR(45),

deleveryInfoId INTEGER NOT NULL,

PRIMARY KEY(id,deleveryInfoId),

CONSTRAINT fk\_Order\_DeleveryInfo1

FOREIGN KEY(deleveryInfoId)

REFERENCES DeleveryInfo(id)

);

#Tạo index  
CREATE unique INDEX fk\_Order\_DeleveryInfo1\_idx ON Orders(deliveryInfoId);

#Tạo bảng OrderMedia  
CREATE TABLE OrderMedia (

orderID int NOT NULL,

price INTEGER NOT NULL,

quantity INTEGER NOT NULL,

mediaId INTEGER NOT NULL,

PRIMARY KEY(orderID,mediaId),

CONSTRAINT fk\_ordermedia\_order

FOREIGN KEY(orderID)

REFERENCES Orders(id),

CONSTRAINT fk\_OrderMedia\_Media1

FOREIGN KEY(mediaId)

REFERENCES Media(id)

);

#Tạo index  
CREATE unique INDEX fk\_ordermedia\_order\_idx ON OrderMedia(orderID);

CREATE Unique INDEX fk\_OrderMedia\_Media1\_idx ON OrderMedia(mediaId);

#Tạo bảng Invoice

CREATE TABLE Invoice(

id INTEGER PRIMARY KEY NOT NULL,

totalAmount INTEGER NOT NULL,

orderId INTEGER NOT NULL,

CONSTRAINT fk\_Invoice\_Order1

FOREIGN KEY(orderId)

REFERENCES Orders(id)

);

#Tạo index

CREATE unique INDEX fk\_Invoice\_Order1\_idx ON Invoice(orderId);

#Tạo bảng PaymentTransaction

CREATE TABLE PaymentTransaction(

id INTEGER NOT NULL,

createdAt DATETIME NOT NULL,

content VARCHAR(45) NOT NULL,

method VARCHAR(45),

cardId INTEGER NOT NULL,

invoiceId INTEGER NOT NULL,

PRIMARY KEY(id,cardId,invoiceId),

CONSTRAINT fk\_PaymentTransaction\_Card1

FOREIGN KEY(cardId)

REFERENCES Card(id),

CONSTRAINT fk\_PaymentTransaction\_Invoice1

FOREIGN KEY(invoiceId)

REFERENCES Invoice(id)

);

#Tạo index

CREATE unique INDEX fk\_PaymentTransaction\_Card1\_idx ON PaymentTransaction(cardId);

CREATE unique INDEX fk\_PaymentTransaction\_Invoice1\_idx ON PaymentTransaction (invoiceId);

#Lưu lại mọi thứ

COMMIT;

# BTVN

## Thiết kế lớp chi tiết cho usecase “Place Rush Order”

### Class “Cart”

Table

Description automatically generated

Class “Cart” vẫn giống ở usecase “PlaceOrder” nhưng có có thêm Operation “checkProductForRushShipping”

* **Operation:** checkProductForRushShipping
  + returnType : boolean
  + Description : kiểm tra sản phẩm trong giỏ hàng có hỗ trợ đặt hàng nhanh hay không
  + Parameters: Không
  + Exception: Không

### Class “PlaceRushOrderController”

Diagram

Description automatically generated with low confidence

**Attribute**

Không

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | placeRushOrder | void | Yêu cầu đặt hàng nhanh |
| 2 | processPlaceRushOrder | void | Thêm thông tin giao hàng nhanh |

*Parameter*:

-        shipment – thông tin giao hàng nhanh

*Exception*:

* Không

**Method**

* Không

**State**

Không

### Class “RushShippingScreenHandler”

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | shipment | Shipment | NULL | Represent the Shipment |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | initializeRushForm | void | Khởi tạo màn hình điền thông tin giao hành nhanh |
| 2 | SubmitDeliveryRushOrderInfo | void | Xác nhận thêm thông tin giao hàng nhanh |

*Parameter*:

-        shipment – thông tin giao hàng nhanh

*Exception*:

* Không

**Method**

-        Không

**State**

Không

### Class “LocationRushShipping”

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | address | String[] | [“Hà Nội”, “HCM”,…] | Các địa điểm hỗ trợ giao hàng nhanh |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | checkAddressForRushShipping | boolean | Kiểm tra địa điểm giao hàng có hỗ trợ giao hàng nhanh hay không |

*Parameter*:

-        address – địa chỉ giao hàng

*Exception*:

* Không

**Method**

-        Không

**State**

Không

## Mô hình hóa dữ liệu cho usecase “PlaceRushOrder”

Trong usecase “placeRushOrder” này, em chỉ thêm thuộc tính rushShipping trong bảng Order để xác định đơn hàng đó được đặt theo phương thức giao hàng nhanh hay giao thường, trong bảng deliveryInfo được bổ sung thêm 2 thuộc tính là dateShipping (ngày giao hàng mong muốn) và scheduledTime (thời gian giao hàng mong muốn)

### Mô hình khái niệm (Conceptual Data Model)

**AIMS SYSTEM ERD**

Shape

Description automatically generated

\*rushShipping, dateShipping, scheduledTime được tô màu đỏ.

## Thiết kế cơ sở dữ liệu

### **Mô hình dữ liệu logic (Logical data model)**

Sơ đồ dữ liệu sau khi thêm các thuộc tính cho bảng orders và deliveryInfo

Diagram

Description automatically generated

### Mô hình dữ liệu vật lý.

Hai bảng có thêm thuộc tính là Orders và DeliveryInfo

* **DeliveryInfo**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, auto increment |
| 2 |  |  | name | Varchar(45) | Yes | Receiver name |
| 3 |  |  | Province | Varchar(45) | Yes | Provinces |
| 4 |  |  | Instructions | Varcher(200) | No | Delivery intraction |
| 5 |  |  | address | Varchar(100) | Yes | Delivey address |
| 6 |  |  | dateShipping | Datetime | No | Date shipping |
| 7 |  |  | scheduledTime | String | No | Time shipping |

* **Order**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column name | Data type | Mandatory | Description |
| 1 | x |  | id | Integer | Yes | ID, auto increment |
| 2 |  |  | shippingFees | Varchar(45) | Yes | Shipping fee |
| 3 |  | x | deliveryInfold | Integer | Yes | Delivey Info ID |
| 4 |  |  | rushShipping | Integer | Yes | Type shipping |

Tạo bảng mới

#Tạo DeliveryInfo

CREATE TABLE DeleveryInfo(

id INTEGER PRIMARY KEY auto\_increment NOT NULL,

name VARCHAR(45),

province VARCHAR(45),

instructions VARCHAR(200),

address VARCHAR(100),

dateShipping Datetime,

scheduledTime varchar(45)

);

#Tạo order

CREATE TABLE Orders(

id INTEGER NOT NULL,

shippingFees VARCHAR(45),

deleveryInfoId INTEGER NOT NULL,

rushShipping int(2),

PRIMARY KEY(id,deleveryInfoId),

CONSTRAINT fk\_Order\_DeleveryInfo1

FOREIGN KEY(deleveryInfoId)

REFERENCES DeleveryInfo(id)

);