Bài thực hành 03 – Tuần 8, ngày 16/11/2021

Contents

[I. Thiết kế lớp: 3](#_Toc88258788)

[1. Tạo các lớp thiết kế: 3](#_Toc88258789)

[a. Thiết kế lớp Boundary: 3](#_Toc88258790)

[b. Thiết kế lớp Entity: 4](#_Toc88258791)

[c. Thiết kế lớp Control: 4](#_Toc88258792)

[d. Nhóm các lớp thiết kế: 5](#_Toc88258793)

[2. Xác định mối quan hệ giữa các lớp: 6](#_Toc88258794)

[3. Lớp thiết kế: 7](#_Toc88258795)

[a. Lớp “InterbankInterface”: 7](#_Toc88258796)

[b. Lớp “PaymentController”: 9](#_Toc88258797)

[c. Biểu đồ trạng thái cho đối tượng Order: 10](#_Toc88258798)

[4. Biểu đồ lớp thiết kế: 11](#_Toc88258799)

[II. Mô hình hóa dữ liệu: 12](#_Toc88258800)

[1. Mô hình khái niệm: 12](#_Toc88258801)

[2. Thiết kế cơ sỡ dữ liệu: 13](#_Toc88258802)

[a. Mô hình dữ liệu logic: 13](#_Toc88258803)

[b. Mô hình dữ liệu vật lý: 14](#_Toc88258804)

[c. Script tạo database: 18](#_Toc88258805)

[III. Bài tập thiết kế lớp chi tiết và mô hình hóa dữ liệu cho Use case “Place Rush Order”: 23](#_Toc88258806)

[1. Thiết kế lớp: 23](#_Toc88258807)

[a. Quan hệ giữa các lớp liên quan trong use case: 23](#_Toc88258808)

[b. Thiết kế lớp: 23](#_Toc88258809)

[2. Mô hình hóa dữ liệu: 24](#_Toc88258810)

[a. Entity Relationship Diagram: 24](#_Toc88258811)

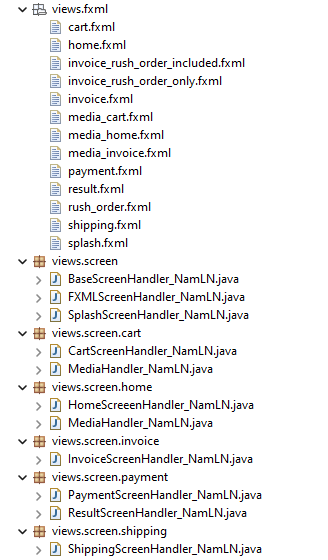
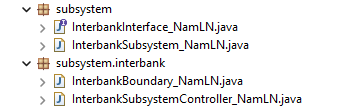
[b. Thiết kế chi tiết các bảng: 25](#_Toc88258812)

[c. Script tạo database: 25](#_Toc88258813)

# Thiết kế lớp:

## Tạo các lớp thiết kế:

### Thiết kế lớp Boundary:

* Giao diện người dùng (User Interface):
* Giao diện hệ thống (System Boundary):

### Thiết kế lớp Entity:

### Thiết kế lớp Control:

### Text Description automatically generated with medium confidenceNhóm các lớp thiết kế:

## Diagram Description automatically generatedDiagram Description automatically generatedXác định mối quan hệ giữa các lớp:

## Lớp thiết kế:

### Text Description automatically generatedLớp “InterbankInterface”:

***Attribute***

Không

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 1 | payOrder | PaymentTransaction | Thanh toán đơn hàng 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 – nếu mã lỗi trả về đã biết
* UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 2 | refund | PaymentTransaction | 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 – nếu mã lỗi trả về đã biết
* UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 3 | getBalance | int | Trả về số dư trong tài khoản |

*Parameter:*

* card – thẻ tín dụng để giao dịch

*Exception:*

***Không***

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

### Table Description automatically generated with medium confidenceLớp “PaymentController”:

***Attribute***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Giá trị mặc định* | *Mô tả* |
| 1 | card | CreditCard | NULL | Represent the card used for payment |
| 2 | interbank | InterbankInterface | NULL | Represent the Interbank subsystem |

***Operation***

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 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
* expirationDate – ngày hết hạn theo định dạng "mm/yy"
* securityCode - mã bảo mật cvv/cvc

*Exception:*

***Không***

***Operation***

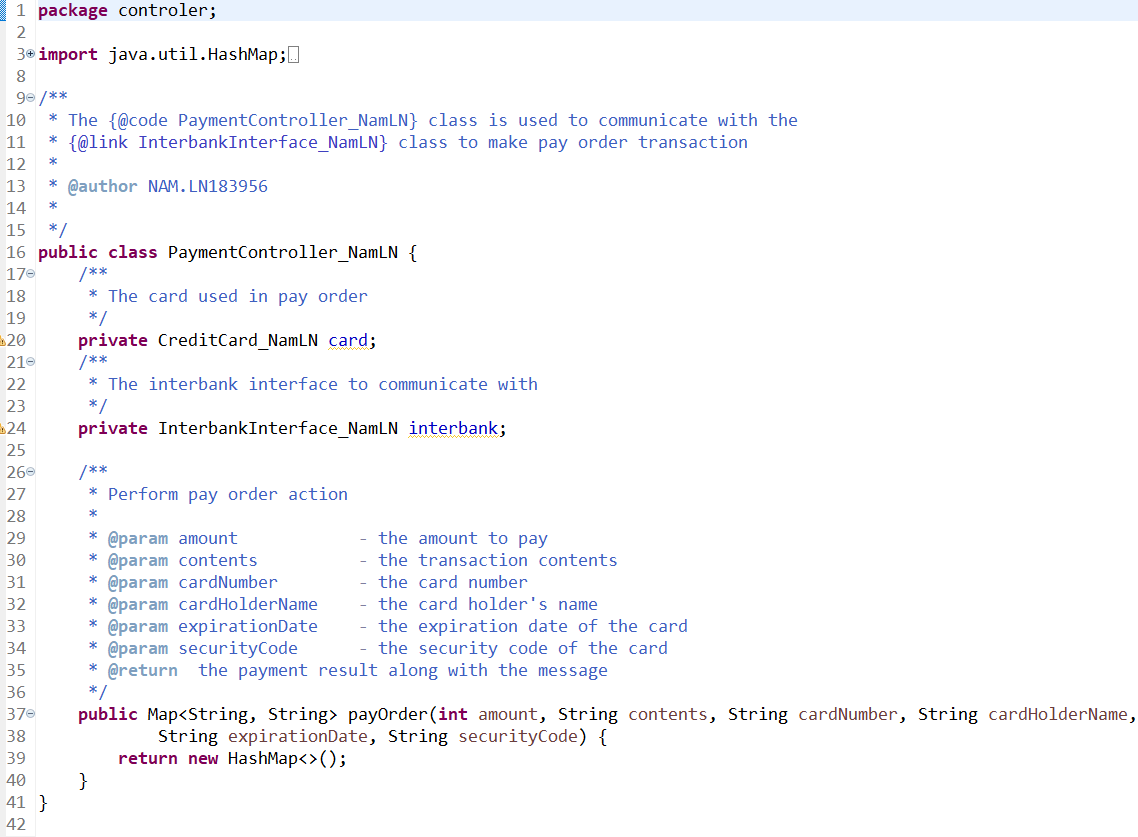
|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Tên* | *Kiểu dữ liệu trả về* | *Mô tả (mục đích)* |
| 2 | getExpirationDate | String | Chuyển định dạng dữ liệu ngày tháng sang “mmyy” |

*Parameter:*

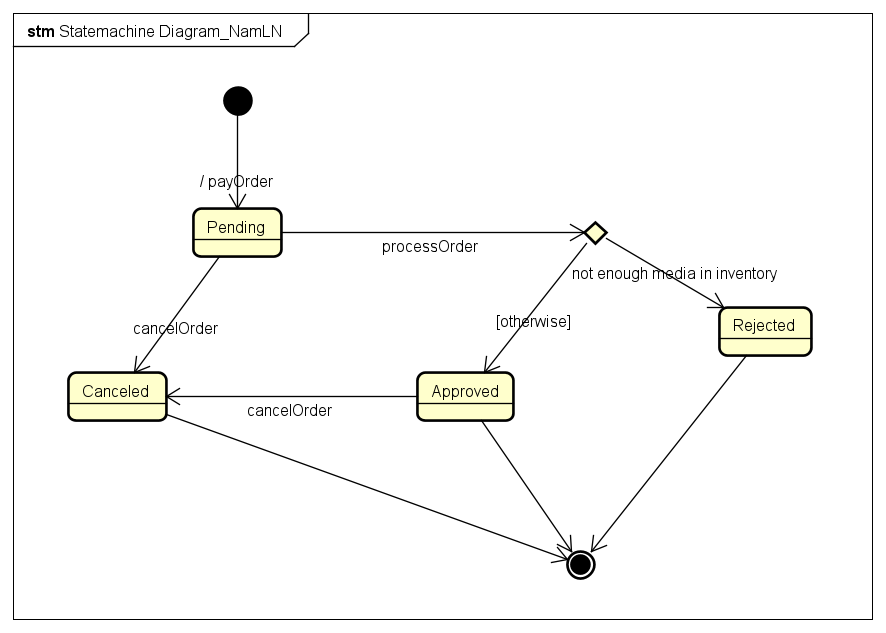
* date – dữ liệu ngày tháng cần chuyển đổi

*Exception:*

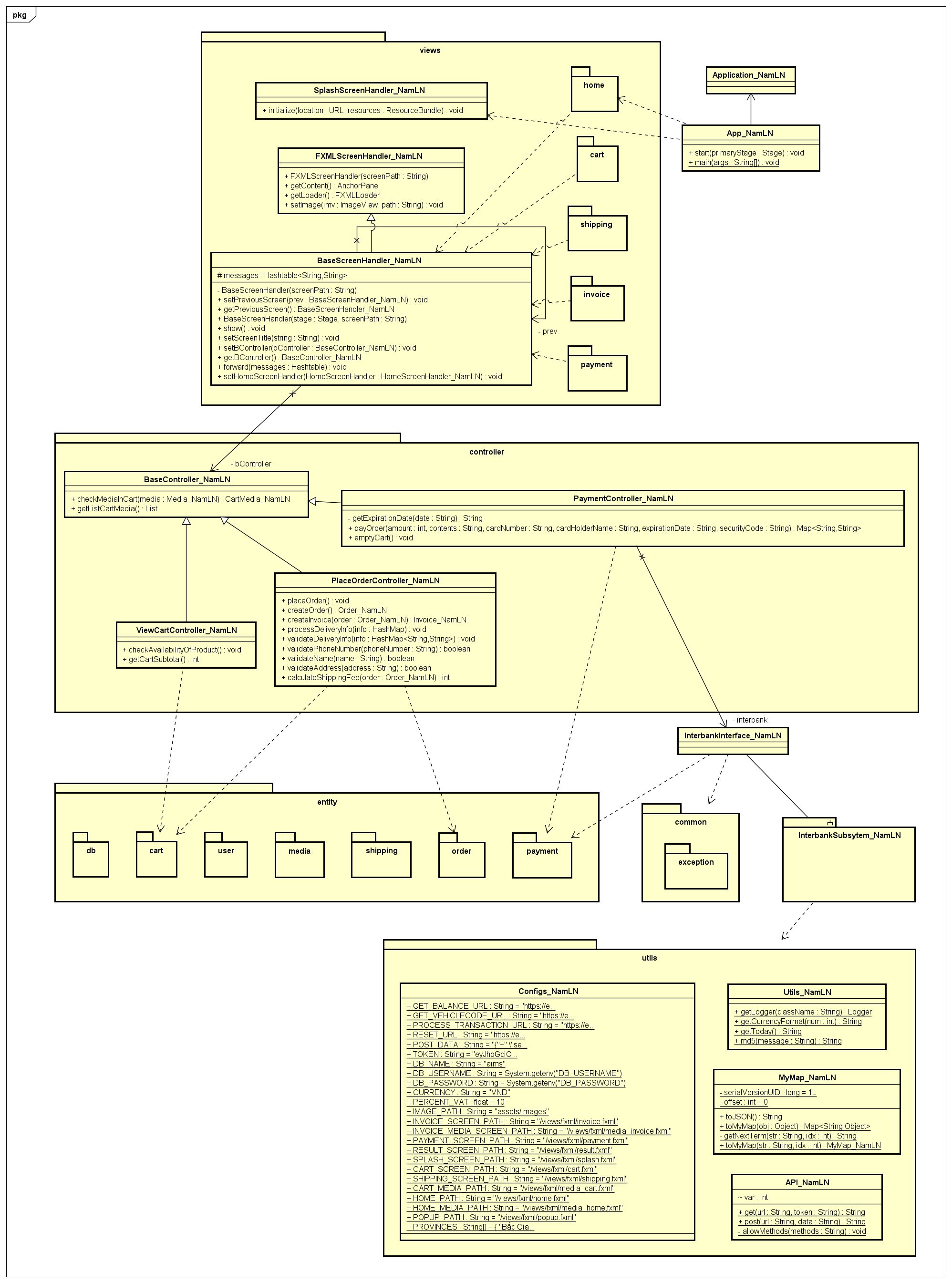
***Không***



### Biểu đồ trạng thái cho đối tượng Order:



## Biểu đồ lớp thiết kế:



# Mô hình hóa dữ liệu:

## Mô hình khái niệm:

## Thiết kế cơ sỡ dữ liệu:

### Mô hình dữ liệu logic:

### Mô hình dữ liệu vật lý:

* **Media\_NamLN**

| # | PK | FK | Column Name | Data type | Mandatory | Description |
| --- | --- | --- | --- | --- | --- | --- |
|  | x |  | id | Integer | Yes | ID, auto increment |
|  |  |  | category | VARCHAR(45) | Yes | Media type, e.g., CD, DVD |
|  |  |  | price | Integer | Yes | Current price |
|  |  |  | quantity | Integer | Yes | Number of products |
|  |  |  | title | VARCHAR(45) | Yes | Product name |
|  |  |  | value | Integer | Yes | Value of the product |
|  |  |  | imageUrl | VARCHAR(45) | Yes | Product image path |

* **CD\_NamLN**

| # | PK | FK | Column Name | Data type | Mandatory | Description |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | x | id | Integer | Yes | ID, same as ID of Media of which type is CD |
|  |  |  | artist | VARCHAR(45) | Yes | Artist's name |
|  |  |  | recordLabel | VARCHAR(45) | Yes | Record label |
|  |  |  | musicType | VARCHAR(45) | Yes | Music genres |
|  |  |  | releasedDate | DATE | No | Release date |

* **Book\_NamLN**

| # | PK | FK | Column Name | Data type | Mandatory | Description |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | x | id | Integer | Yes | ID, same as ID of Media of which type is Book |
|  |  |  | author | VARCHAR(45) | Yes | Author |
|  |  |  | coverType | VARCHAR(45) | Yes | Cover type |
|  |  |  | Publisher | VARCHAR(45) | Yes | Publishing house |
|  |  |  | publishDate | DATETIME | Yes | Date of publishing |
|  |  |  | numOfPages | Integer | Yes | Page number |
|  |  |  | language | VARCHAR(45) | Yes | Language |
|  |  |  | bookCategory | VARCHAR(45) | Yes | Book category |

* **DVD\_NamLN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data type | Mandatory | Description |
|  |  | x | id | Integer | Yes | ID, same as ID of Media of which type is DVD |
|  |  |  | discType | VARCHAR(45) | Yes | Disc type |
|  |  |  | director | VARCHAR(45) | Yes | Director |
|  |  |  | runtime | Integer | Yes | Duration |
|  |  |  | studio | VARCHAR(45) | Yes | Manufacturer |
|  |  |  | subtitle | VARCHAR(45) | Yes | Subtitles |
|  |  |  | releasedDate | DATETIME | Yes | Release date |
|  |  |  | filmType | VARCHAR(45) | Yes | Genres |

* **Card\_NamLN**

| # | PK | FK | Column Name | Data type | Mandatory | Description |
| --- | --- | --- | --- | --- | --- | --- |
|  | x |  | id | Integer | Yes | ID, auto increment |
|  |  |  | cardCode | VARCHAR(45) | Yes | Card code |
|  |  |  | owner | VARCHAR(45) | Yes | Cardholders |
|  |  |  | cvvCode | VARCHAR(3) | Yes | CVV code |
|  |  |  | dateExpired | VARCHAR(4) | Yes | Expiration date |

* **DeliveryInfo\_NamLN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data type | Mandatory | Description |
|  | X |  | id | Integer | Yes | ID, auto increment |
|  |  |  | name | VARCHAR(45) | Yes | Receiver name |
|  |  |  | province | VARCHAR(45) | Yes | Provinces |
|  |  |  | instructions | VARCHAR(200) | No | Delivery instructions |
|  |  |  | address | VARCHAR(100) | Yes | Delivery address |

* **Order\_NamLN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data type | Mandatory | Description |
|  | X |  | id | Integer | Yes | ID |
|  |  |  | shippingFees | VARCHAR(45) | Yes | Shipping fee |
|  |  | X | deliveryInfoId | Integer | Yes | Delivery Info ID |

* **OrderMedia\_NamLN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data type | Mandatory | Description |
|  |  | X | mediaID | Integer | Yes | Media ID |
|  |  | X | orderID | Integer | Yes | Order ID |
|  |  |  | price | Integer | Yes | Selling price |
|  |  |  | quantity | Integer | Yes | Number |

* **Invoice\_NamLN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data type | Mandatory | Description |
|  | x |  | id | Integer | Yes | ID |
|  |  |  | totalAmount | Integer | Yes | Total |
|  |  | x | orderId | Integer | Yes | Order ID |

* **PaymentTransaction\_NamLN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data type | Mandatory | Description |
|  | x |  | id | Integer | Yes | ID |
|  |  |  | createAt | DATETIME | Yes | Date of creation |
|  |  |  | content | VARCHAR(45) | Yes | Transaction contents |
|  |  |  | method | VARCHAR(45) | Yes | Payment methods |
|  |  | x | cardId | Integer | Yes | ID of used card |
|  |  | x | invoiceId | Integer | Yes | Invoice ID |

### Script tạo database:

CREATE TABLE IF NOT EXISTS `aims`.`Media\_NamLN` (

`id` INT NOT NULL AUTO\_INCREMENT,

`category` VARCHAR(45) NOT NULL,

`price` INT NOT NULL,

`quantity` INT NOT NULL,

`title` VARCHAR(45) NOT NULL,

`value` INT NOT NULL,

`imageUrl` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`Book\_NamLN` (

`id` INT NOT NULL,

`author` VARCHAR(45) NOT NULL,

`coverType` VARCHAR(45) NOT NULL,

`publisher` VARCHAR(45) NOT NULL,

`publishDate` DATETIME NOT NULL,

`numOfPages` INT NOT NULL,

`language` VARCHAR(45) NOT NULL,

`bookCategory` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`),

CONSTRAINT `fk\_Book\_Media1`

FOREIGN KEY (`id`)

REFERENCES `aims`.`Media\_NamLN` (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`Card\_NamLN` (

`id` INT NOT NULL AUTO\_INCREMENT,

`cardCode` VARCHAR(15) NOT NULL,

`owner` VARCHAR(45) NOT NULL,

`cvvCode` VARCHAR(3) NOT NULL,

`dateExpired` VARCHAR(4) NOT NULL,

PRIMARY KEY (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`CD\_NamLN` (

`id` INT NOT NULL,

`artist` VARCHAR(45) NOT NULL,

`recordLabel` VARCHAR(45) NOT NULL,

`musicType` VARCHAR(45) NOT NULL,

`releasedDate` DATE NULL DEFAULT NULL,

PRIMARY KEY (`id`),

CONSTRAINT `fk\_CD\_Media1`

FOREIGN KEY (`id`)

REFERENCES `aims`.`Media\_NamLN` (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`DeleveryInfo\_NamLN` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL DEFAULT NULL,

`province` VARCHAR(45) NULL DEFAULT NULL,

`instructions` VARCHAR(200) NULL DEFAULT NULL,

`address` VARCHAR(100) NULL DEFAULT NULL,

PRIMARY KEY (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`DVD\_NamLN` (

`id` INT NOT NULL,

`discType` VARCHAR(45) NOT NULL,

`director` VARCHAR(45) NOT NULL,

`runtime` INT NOT NULL,

`studio` VARCHAR(45) NOT NULL,

`subtitle` VARCHAR(45) NOT NULL,

`releasedDate` DATETIME NULL DEFAULT NULL,

PRIMARY KEY (`id`),

CONSTRAINT `fk\_DVD\_Media1`

FOREIGN KEY (`id`)

REFERENCES `aims`.`Media\_NamLN` (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`Order\_NamLN` (

`id` INT NOT NULL,

`shippingFees` VARCHAR(45) NULL DEFAULT NULL,

`deleveryInfoId` INT NOT NULL,

PRIMARY KEY (`id`, `deleveryInfoId`),

INDEX `idx\_fk\_Order\_DeleveryInfo1` (`deleveryInfoId` ASC) VISIBLE,

CONSTRAINT `fk\_Order\_DeleveryInfo1`

FOREIGN KEY (`deleveryInfoId`)

REFERENCES `aims`.`DeleveryInfo\_NamLN` (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`Invoice\_NamLN` (

`id` INT NOT NULL,

`totalAmount` INT NOT NULL,

`orderId` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `idx\_fk\_Invoice\_Order1` (`orderId` ASC) VISIBLE,

CONSTRAINT `fk\_Invoice\_Order1`

FOREIGN KEY (`orderId`)

REFERENCES `aims`.`Order\_NamLN` (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`OrderMedia\_NamLN` (

`orderID` INT NOT NULL,

`price` INT NOT NULL,

`quantity` INT NOT NULL,

`mediaId` INT NOT NULL,

PRIMARY KEY (`orderID`, `mediaId`),

INDEX `idx\_fk\_ordermedia\_order` (`orderID` ASC) VISIBLE,

INDEX `idx\_fk\_OrderMedia\_Media1` (`mediaId` ASC) VISIBLE,

CONSTRAINT `fk\_OrderMedia\_Media1`

FOREIGN KEY (`mediaId`)

REFERENCES `aims`.`Media\_NamLN` (`id`),

CONSTRAINT `fk\_ordermedia\_order`

FOREIGN KEY (`orderID`)

REFERENCES `aims`.`Order\_NamLN` (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`PaymentTransaction\_NamLN` (

`id` INT NOT NULL,

`createAt` DATETIME NOT NULL,

`content` VARCHAR(45) NOT NULL,

`method` VARCHAR(45) NULL DEFAULT NULL,

`cardId` INT NOT NULL,

`invoiceId` INT NOT NULL,

PRIMARY KEY (`id`, `cardId`, `invoiceId`),

INDEX `idx\_fk\_PaymentTransaction\_Card1` (`cardId` ASC) VISIBLE,

INDEX `idx\_fk\_PaymentTransaction\_Invoice1` (`invoiceId` ASC) VISIBLE,

CONSTRAINT `fk\_PaymentTransaction\_Card1`

FOREIGN KEY (`cardId`)

REFERENCES `aims`.`Card\_NamLN` (`id`),

CONSTRAINT `fk\_PaymentTransaction\_Invoice1`

FOREIGN KEY (`invoiceId`)

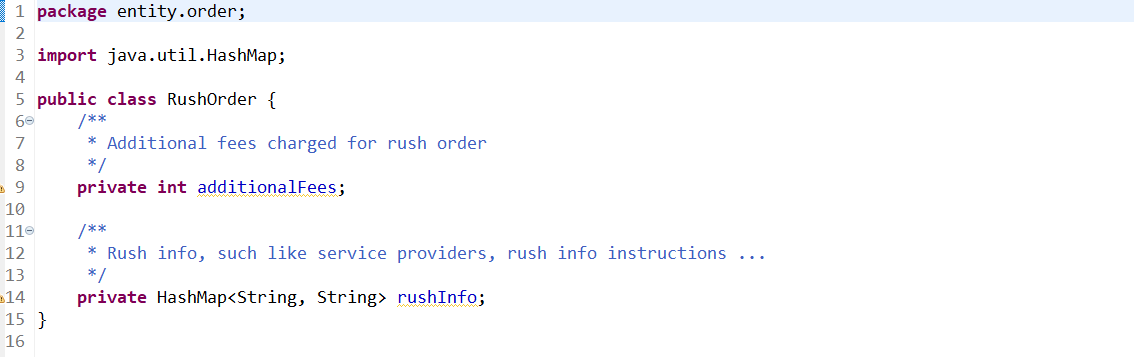
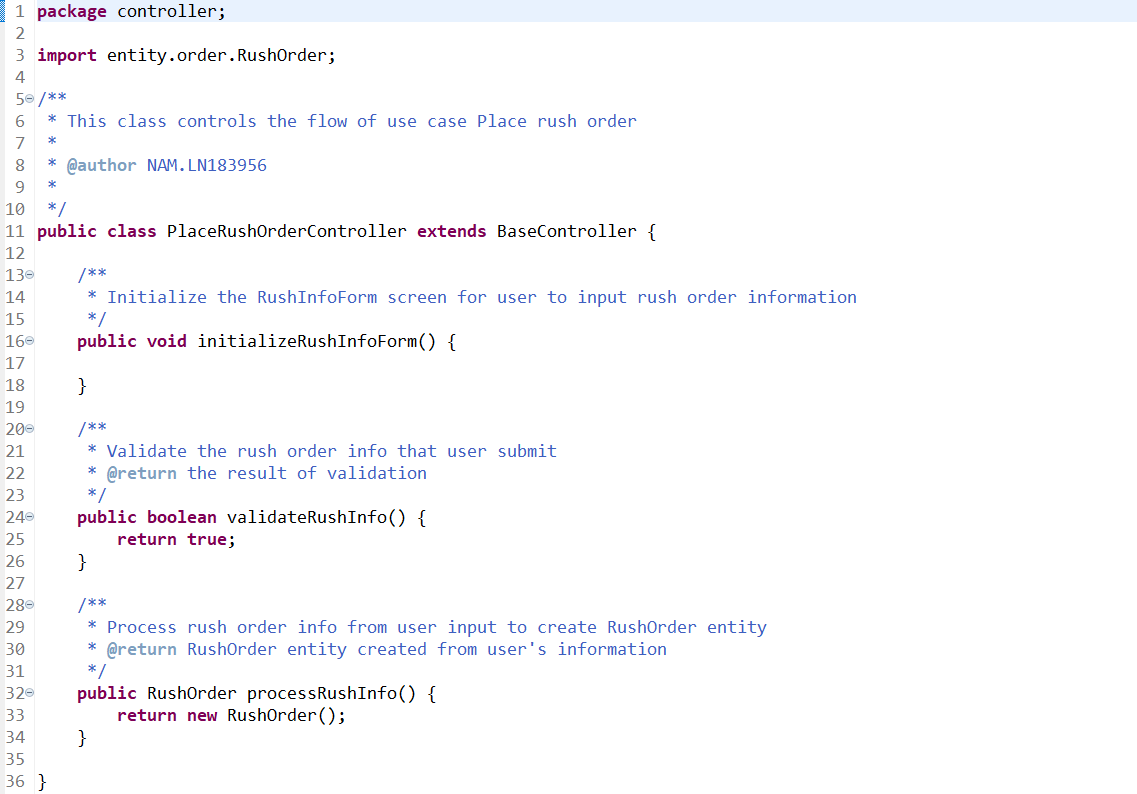
REFERENCES `aims`.`Invoice\_NamLN` (`id`))

# Bài tập thiết kế lớp chi tiết và mô hình hóa dữ liệu cho Use case “Place Rush Order”:

## Thiết kế lớp:

### Quan hệ giữa các lớp liên quan trong use case:

### Thiết kế lớp:

* Lớp RushOrder (Entity):
* Lớp RushOrderController (Control):

## Mô hình hóa dữ liệu:

### Diagram Description automatically generatedEntity Relationship Diagram:

### Thiết kế chi tiết các bảng:

* **RushInfo\_NamLN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data type | Mandatory | Description |
|  | X |  | id | Integer | Yes | ID, auto increment |
|  |  |  | serviceProvider | VARCHAR(45) | Yes | Service provider |
|  |  |  | rushInstruction | VARCHAR(45) | Yes | Rush order instruction |

* **RushOrder\_NamLN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data type | Mandatory | Description |
|  | X |  | id | Integer | Yes | ID |
|  |  |  | additionalFees | VARCHAR(45) | Yes | Additional fee for rush order |
|  |  | X | rushInfoId | Integer | Yes | Rush Order Info ID |

### Script tạo database:

CREATE TABLE IF NOT EXISTS `aims`.`rushinfo\_namln` (

`id` INT NOT NULL AUTO\_INCREMENT,

`serviceProvider` VARCHAR(45) NULL DEFAULT NULL,

`rushInstruction` VARCHAR(45) NULL DEFAULT NULL,

PRIMARY KEY (`id`))

CREATE TABLE IF NOT EXISTS `aims`.`rushorder\_namln` (

`id` INT NOT NULL AUTO\_INCREMENT,

`additionalFees` VARCHAR(45) NULL DEFAULT NULL,

`rushInfoId` INT NOT NULL,

PRIMARY KEY (`id`),

INDEX `rushInfoId` (`rushInfoId` ASC) VISIBLE,

CONSTRAINT `rushorder\_namln\_ibfk\_1`

FOREIGN KEY (`rushInfoId`)

REFERENCES `aims`.`rushinfo\_namln` (`id`))