# Detailed Design

## User Interface Design

*<Suppose that you design a Graphical User Interface (GUI)>*

### Screen Configuration Standardization

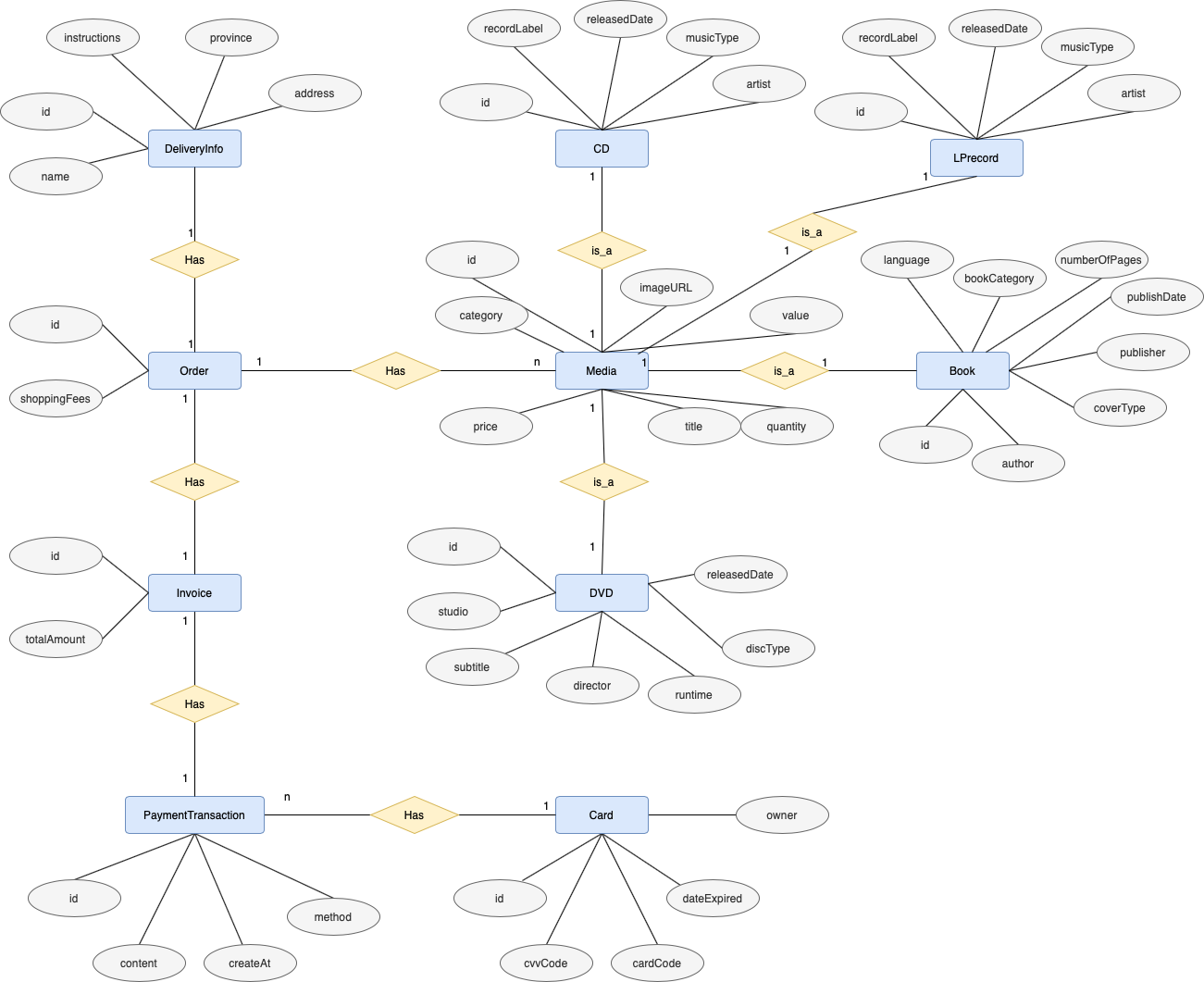
### Screen Transition Diagrams

### Screen Specifications

*<Screen images should be included in the screen specifications>*

## Data Modeling

### Conceptual Data Modeling



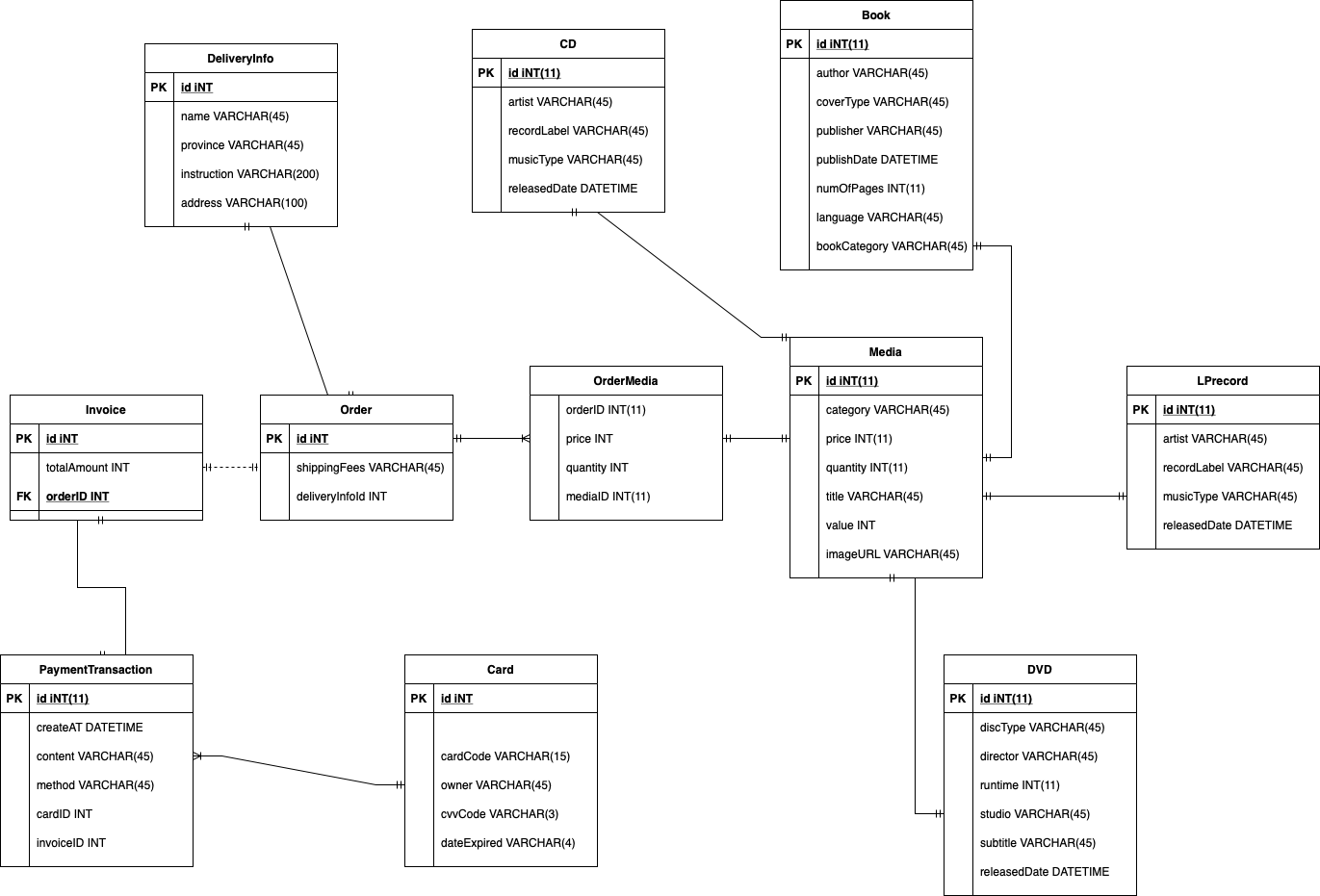
### Database Design

#### Database Management Systems

In this project, we use SQLite as a Database Management System (DBMS) because of these following advantages:

* SQLite is an open-source DBMS

#### Logical Data Model



#### Physical Data Model

* Media

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 | x |  | id | INT | Yes | ID, auto\_increment |
| 2 |  |  | category | VARCHAR(45) | Yes | Media type, eg CD, DVD |
| 3 |  |  | price | INT | Yes | Current price |
| 4 |  |  | quantity | INT | Yes | Number of products |
| 5 |  |  | title | VARCHAR(45) | Yes | Product name |
| 6 |  |  | value | INT | Yes | Value of product |
| 7 |  |  | imageURL | VARCHAR(45) | Yes | Product image path |

* CD

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 |  | x | id | INT | Yes | ID, auto\_increment |
| 2 |  |  | artist | VARCHAR(45) | Yes | Artist’s name |
| 3 |  |  | recordLabel | VARCHAR(45) | Yes | Record label |
| 4 |  |  | musicType | VARCHAR(45) | Yes | Music genres |
| 5 |  |  | releasedDate | DATETIME | No | Released date |

* Book

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 |  | x | id | INT | Yes | ID, auto\_increment |
| 2 |  |  | author | VARCHAR(45) | Yes | Author’s name |
| 3 |  |  | coverType | VARCHAR(45) | Yes | Cover type |
| 4 |  |  | publisher | VARCHAR(45) | Yes | Publisher name |
| 5 |  |  | publishDate | DATETIME | Yes | Publish date |
| 6 |  |  | numOfPages | INT | 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 | INT | Yes | ID, auto\_increment |
| 2 |  |  | discType | VARCHAR(45) | Yes | Disc type |
| 3 |  |  | director | VARCHAR(45) | Yes | Director |
| 4 |  |  | runtime | INT | Yes | Duration |
| 5 |  |  | studio | VARCHAR(45) | Yes | Manufacturer |
| 6 |  |  | subtitle | VARCHAR(45) | Yes | Subtitles |
| 7 |  |  | releasedDate | DATETIME | Yes | Released date |
| 8 |  |  | filmType | VARCHAR(45) | Yes | Genres |

* LPrecord

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 |  | x | id | INT | Yes | ID, auto\_increment |
| 2 |  |  | artist | VARCHAR(45) | Yes | Artist’s name |
| 3 |  |  | recordLabel | VARCHAR(45) | Yes | Record label |
| 4 |  |  | musicType | VARCHAR(45) | Yes | Music genres |
| 5 |  |  | releasedDate | DATETIME | No | Released date |

* Card

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 | x |  | id | INT | Yes | ID, auto\_increment |
| 2 |  |  | cardCode | VARCHAR(45) | Yes | Card code |
| 3 |  |  | owner | VARCHAR(45) | Yes | Card’s holder |
| 4 |  |  | cvvCode | VARCHAR(45) | Yes | CVV code |
| 5 |  |  | expiredDate | VARCHAR(45) | Yes | Expiration date |

* DeliveryInfo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 | x |  | id | INT | Yes | ID, auto\_increment |
| 2 |  |  | name | VARCHAR(45) | Yes | Receiver name |
| 3 |  |  | province | VARCHAR(45) | Yes | Provinces |
| 4 |  |  | instructions | VARCHAR(200) | No | Delivery instructions |
| 5 |  |  | address | VARCHAR(100) | Yes | Delivery address |
| 6 |  |  | startTime | Time |  | startTime for Place Rush Order |
| 7 |  |  | deliveryTime | Time |  | deliveryTime for Place Rush Order |

* Order

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 | x |  | id | INT | Yes | ID |
| 2 |  |  | shippingFees | VARCHAR(45) | Yes | Shipping fee |
| 3 |  | x | deliveryInfoID | INT | Yes | Delivery info ID |

* OrderMedia

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 |  | x | mediaID | INT | Yes | Media ID |
| 2 |  | x | orderID | INT | Yes | Order ID |
| 3 |  |  | price | INT | Yes | Selling price |
| 4 |  |  | quantity | INT | Yes | Number of products |

* Invoice

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 | x |  | id | INT | Yes | ID |
| 2 |  |  | totalAmount | INT | Yes | Total |
| 3 |  |  | orderID | INT | Yes | Order ID |

* PaymentTransaction

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Mandatory* | *Description* |
| 1 | x |  | id | INT | Yes | ID |
| 2 |  |  | createAt | DATETIME | Yes | Create date |
| 3 |  |  | content | VARCHAR(45) | Yes | Transaction contents |
| 4 |  |  | method | VARCHAR(45) | Yes | Payment method |
| 5 |  | x | cardID | INT | Yes | Used card’s ID |
| 6 |  | x | invoiceID | INT | Yes | Invoice ID |

* *Database script*

CREATE TABLE "Media"(

"id" INTEGER PRIMARY KEY AUTOINCREMENT 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

);

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")

);

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")

);

CREATE TABLE "DeliveryInfo"(

"id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,

"name" VARCHAR(45),

"province" VARCHAR(45),

"instructions" VARCHAR(200),

"address" VARCHAR(100),

"isRush" BOOLEAN NOT NULL,

"startTime" TIME,

"deliveryTime" TIME

);

CREATE TABLE "Card"(

"id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,

"cardCode" VARCHAR(15) NOT NULL,

"owner" VARCHAR(45) NOT NULL,

"cvvCode" VARCHAR(3) NOT NULL,

"dateExpired" VARCHAR(4) NOT NULL

);

CREATE TABLE "DVD"(

"id" INTEGER PRIMARY KEY 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,

"genre" VARCHAR(45),

CONSTRAINT "fk\_DVD\_Media1"

FOREIGN KEY("id")

REFERENCES "Media"("id")

);

CREATE TABLE "LPrecord"(

"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\_DVD\_Media1"

FOREIGN KEY("id")

REFERENCES "Media"("id")

);

CREATE TABLE "Order"(

"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")

);

CREATE INDEX "Order.fk\_Order\_DeleveryInfo1\_idx" ON "Order"("deleveryInfoId");

CREATE TABLE "OrderMedia"(

"orderID" INTEGER NOT NULL,

"quantity" INTEGER NOT NULL,

"mediaId" INTEGER NOT NULL,

PRIMARY KEY("orderID","mediaId"),

CONSTRAINT "fk\_ordermedia\_order"

FOREIGN KEY("orderID")

REFERENCES "Order"("id"),

CONSTRAINT "fk\_OrderMedia\_Media1"

FOREIGN KEY("mediaId")

REFERENCES "Media"("id")

);

CREATE INDEX "OrderMedia.fk\_ordermedia\_order\_idx" ON "OrderMedia"("orderID");

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

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 "Order"("id")

);

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

CREATE TABLE "PaymentTransaction"(

"id" INTEGER NOT NULL,

"createAt" 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")

);

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

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

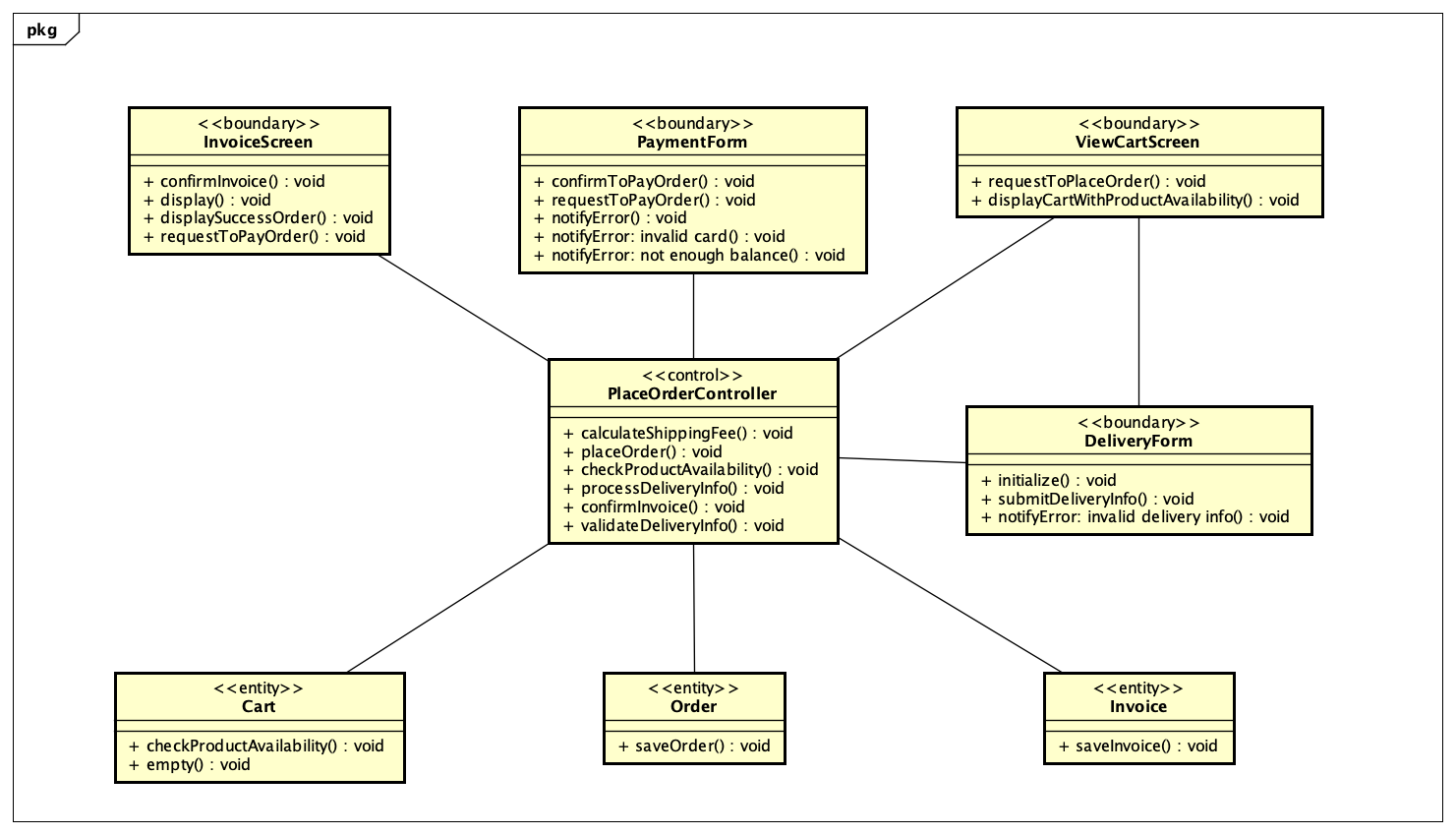
## Non-Database Management System Files

## Class Design

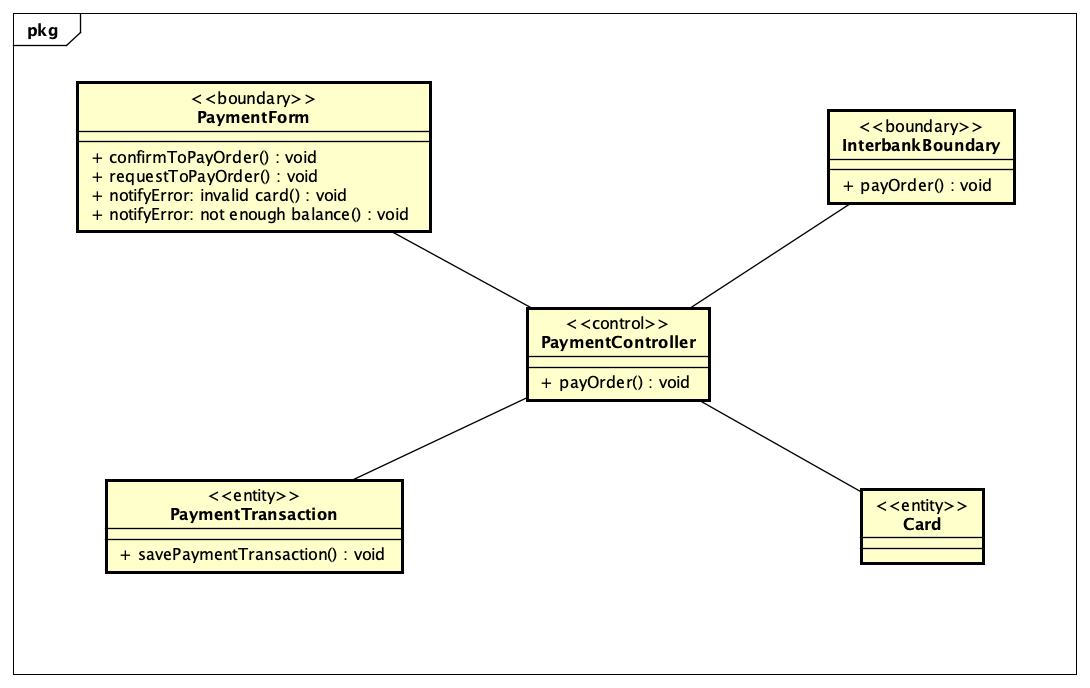
### General Class Diagram

### Class Diagrams

#### Class Diagram for PlaceOrderController

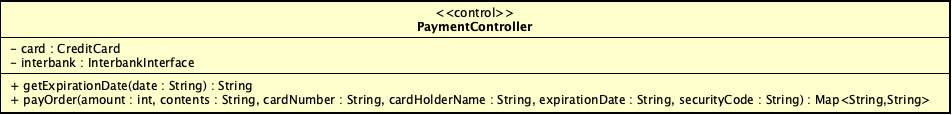


#### Class Diagram for PayOrder



### Class Design

#### Class “PaymentController”



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | Card | CreditCard | NULL | CreditCard used for payment |
| 2 | Interbank | InterbankInterface | NULL | Interbank subsystem |

**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 giao dịch
* cardHolderName: Chủ thẻ
* expirationDate: ngày hết hạn theo định dạng mm/yy
* securityCode: Mã bảo mật

*Exception*:

* Không

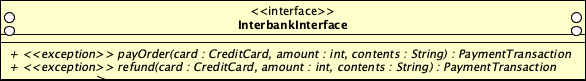
**Method**

* getExpirationDate: chuyển dữ liệu từ định dạng mm/yy sang mmyy

**State**

* Không

#### Class “InterbankInterface”



**Attribute**

* Không

**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 |

*Parameter*:

* card: Thẻ thực hiện giao dịch
* amount: Số tiền giao dịch
* content: Nội dung giao dịch

*Exception*:

* PaymentException: trả về mã lỗi đã biết
* UnrecognizeException: Nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống

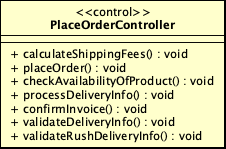
**Method**

* Không

**State**

* Không

#### Class “PlaceOrderController”



**Attribute**

* Không

**Operation**

* Không

**Method**

* calculateShippingFees: trả về phí vận chuyển
* placeOrder: thực hiện đặt hàng
* checkAvailabilityOfProduct: kiểm tra tình trạng hiện tại của hàng
* processDeliveryInfo:
* confirmInvoice: xác nhận hoá đơn
* validateDeliveryInfo: kiểm tra thông tin vận chuyển
* validateRushDeliveryInfo: kiểm tra thông tin vận chuyển nhanh

**State**

* Không

#### Class “ViewCartController”



**Attribute**

* Không

**Operation**

* Không

**Method**

* checkAvailabilityOfProduct: kiểm tra trạng thái hiện tại của hàng hoá

**State**

* Không

#### Class “SplashScreenHandler”

/var/folders/7b/s2_hmx691md96wywgl_g1cjr0000gp/T/TemporaryItems/(A Document Being Saved By screencaptureui 9)/Screen Shot 2021-11-21 at 3.04.08 PM.png

**Attribute**

* Không

**Operation**

* Không

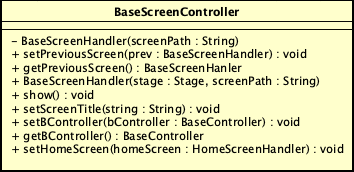
**Method**

* Initialize: Khởi tạo màn hình splash screen

**State**

* Không

#### Class “BaseScreenController”



**Attribute**

* Không

**Operation**

* Không

**Method**

* baseScreenHandler: handler của Base Screen
* setPreviousScreen: đặt trang trước đó
* getPreviousScreen: trả về trang trước đó
* show: hiển thị màn hình
* setScreenTitle: đặt tiêu đề cho trang
* setBController: đặt BController
* getBController: trả về Bcontroller
* setHomeScreen: đặt trang chủ

**State**

* Không