HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Design Document

AIMS – An Internet Media Store

Subject: ITSS Software Development

Group 13

|  |  |
| --- | --- |
| Trương Công Thái Đức | 20225815 |
| Vũ Ngọc Lâm | 20225645 |
| Phan Sỹ Hùng | 20225631 |
| Nguyễn Mạnh Tuấn | 20225679 |

*Hanoi, May 2025*

Table of Contents

Table of Contents 1

1 Introduction 5

1.1 Objective 5

1.2 Scope 5

1.3 Glossary 5

1.4 References 5

2 Overall Description 7

2.1 General Overview 7

2.2 Assumptions/Constraints/Risks 7

2.2.1 Assumptions 7

2.2.2 Constraints 7

2.2.3 Risks 8

3 System Architecture and Architecture Design 9

3.1 Architectural Patterns 9

3.2 Interaction Diagrams 9

3.3 Analysis Class Diagrams 9

3.4 Unified Analysis Class Diagram 9

3.5 Security Software Architecture 9

4 Detailed Design 10

4.1 User Interface Design 10

4.1.1 Screen Configuration Standardization 10

4.1.2 Screen Transition Diagrams 10

4.1.3 Screen Specifications 10

4.2 Data Modeling 10

4.2.1 Conceptual Data Modeling 10

4.2.2 Database Design 10

4.3 Non-Database Management System Files 11

4.4 Class Design 11

4.4.1 General Class Diagram 11

4.4.2 Class Diagrams 11

4.4.3 Class Design 11

5 Design Considerations 13

5.1 Goals and Guidelines 13

5.2 Architectural Strategies 13

5.3 Coupling and Cohesion 14

5.4 Design Principles 14

5.5 Design Patterns 15

**List of Figures**

**No table of figures entries found.**

**List of Tables**

[Table 1. Example of table design 10](#_Toc197437803)

[Table 1. Example of attribute design 12](#_Toc197437804)

[Table 1. Example of operation design 12](#_Toc197437805)

[Table 4. Example of attribute design 14](#_Toc197437806)

# Introduction

This SRS document serves as a comprehensive guide outlining the requirements, features, and constraints of the software project. It is intended to provide a clear understanding of the scope and objectives of the project for all stakeholders involved, including clients, users, and development team members.

## Objective

The primary purpose of this document is to establish a common understanding of the software requirements among all stakeholders. It defines the functionalities, features, and performance expectations of the software system to be developed. By documenting these requirements in detail, this document serves as a reference point throughout the software development lifecycle, guiding the design, implementation, testing, and deployment phases. The intended audience includes project stakeholders, development teams, system architects, UI/UX designers, quality assurance testers, end-user representatives, and regulatory officers, all of whom rely on this document to ensure the successful realization of AIMS.

## Scope

This AIMS – ‘An Internet Media Store’ software is developed to be a desktop platform e-commerce software, which helps users to order media products on the Internet, and the store managers, at the same time, are easier to manage their store as well as the orders.

This software can serve up to 1,000 customers simultaneously without significantly reducing performance and can operate continuously for 300 hours without failure. Additionally, the software can resume normal operation within a maximum of 1 hour after an incident. The maximum response time of the software is 2 seconds under normal conditions or 5 seconds during peak hours.

In AIMS, customers can not only search for products, but also sort products as they desire, they can place order or rush order for necessary cases. AIMS is supported by VNPay transactions; thus, customers can easily pay for their order. Moreover, customers can review their order and modify any information during the processing order stage. While shopkeepers can many their store by managing products directly in the system. They, meanwhile, can process the orders of the customers. For administrators, they are capable of managing users and privileges problems of users.

Additionally, for a desktop website, the graphical user interface (GUI) is carefully considered to meet end-user requirements and enhance the overall user experience. Throughout the development process, all documentation is systematically recorded to facilitate future maintenance and upgrades. We maintain a strong focus on each stage, ensuring adherence to the client’s timeline and the delivery of high-quality software. If any modifications are required, our team swiftly adapts to revise and refine our work accordingly.

## Glossary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***No*** | ***Term*** | ***Explanation*** | ***Example*** | ***Note*** |
| 1 | Session | a session is a temporary interaction period between a user and the software, starting when the application is launched and ending when it is closed | Shopping Cart Session |  |
| 2 | VAT  (Value-added tax) | A **10% tax** applied to the product price, which is not included in the base value set by the product manager |  |  |
| 3 | API (Application Programming Protocol | API is a set of rules, protocols, and tools that allows different software apps to communicate with each other. |  |  |
| 4 | VNPay | An online payment gateway integrated with AIMS for online transaction |  |  |
| 5 | Shipping fees | Costs associated with delivering a product, calculated based on weight and location |  |  |

## References

Centers for Medicare & Medicaid Services. (n.d.). *System Design Document Template.* Retrieved from Centers for Medicare & Medicaid Services: https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SystemDesignDocument.docx

*<Listing the referenced material used in this document, including the one related to the project>*

# Overall Description

<*This section describes the principles and strategies to be used as guidelines when designing and implementing the system.>*

## General Overview

*<Briefly introduce the system context and the basic design approach or organization. Provide a brief overview of the system and software architectures and the design goals. Include the high-level context diagram(s) for the system and subsystems provided in previous documents like SRS (e.g., general use case diagram, lower-level use case diagrams, activity diagrams), updated as necessary to reflect any changes that have been made based on more current information or understanding. If the high-level context diagram has been updated, identify the changes that were made and why>*

## Assumptions/Constraints/Risks

### Assumptions

*<Describe any assumptions or dependencies regarding the system, software and its use. These may concern such issues as: related software or hardware, operating systems, end-user characteristics, and possible and/or probable changes in functionality>*

### Constraints

*<Describe any global limitations or constraints that have a significant impact on the design of the system’s hardware, software and/or communications, and describe the associated impact. Such constraints may be imposed by any of the following (the list is not exhaustive):*

* *Hardware or software environment*
* *End-user environment*
* *Availability or volatility of resources*
* *Standards compliance*
* *Interoperability requirements*
* *Interface/protocol requirements*
* *Licensing requirements*
* *Data repository and distribution requirements*
* *Security requirements (or other such regulations)*
* *Memory or other capacity limitations*
* *Performance requirements*
* *Network communications*
* *Verification and validation requirements (testing)*
* *Other means of addressing quality goals*
* *Other requirements described in the Requirements Document*

*>*

### Risks

*<Describe any risks associated with the system design and proposed mitigation strategies.>*

# System Architecture and Architecture Design

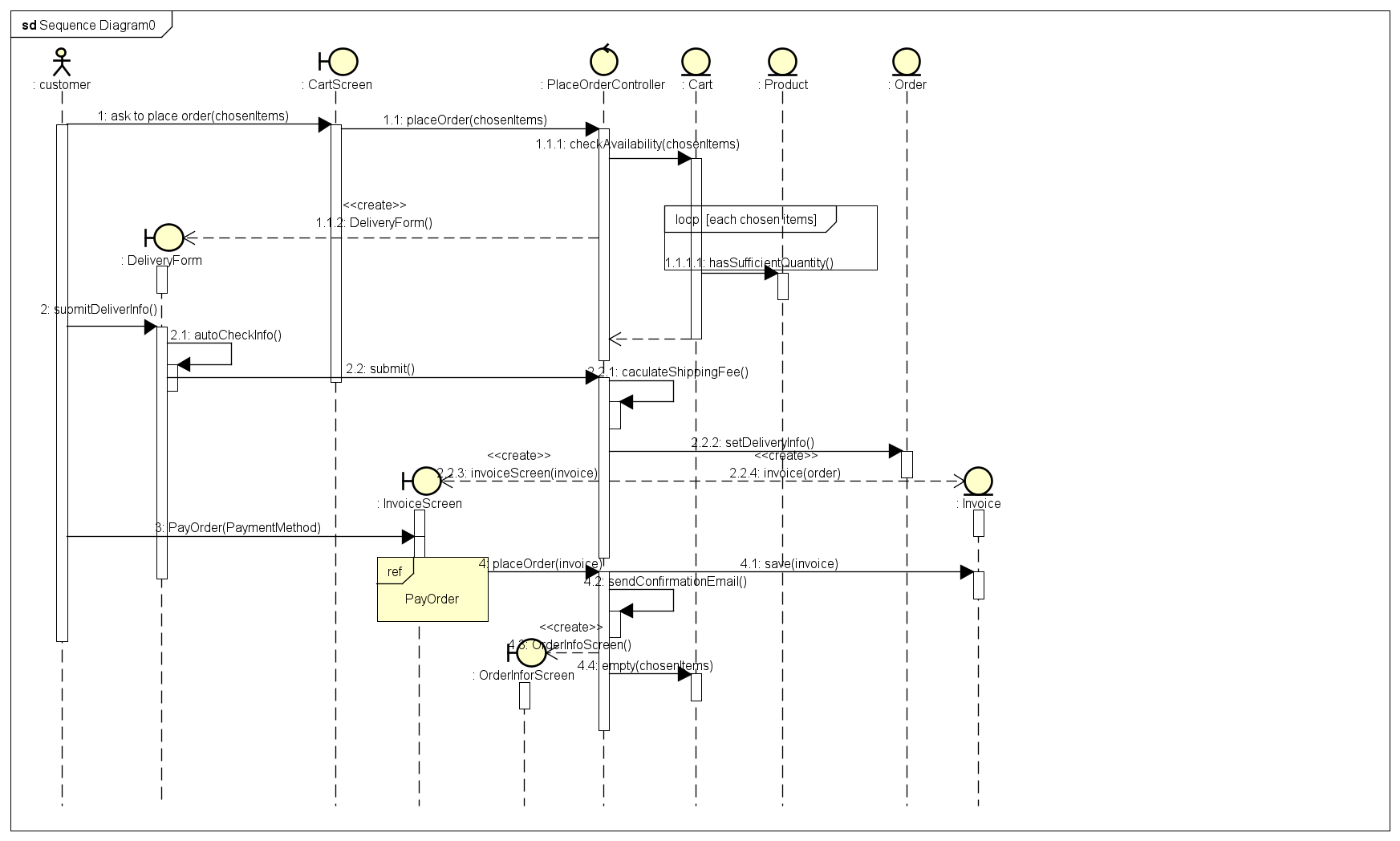
<*Briefly describe the architectural design steps*>

## Architectural Patterns

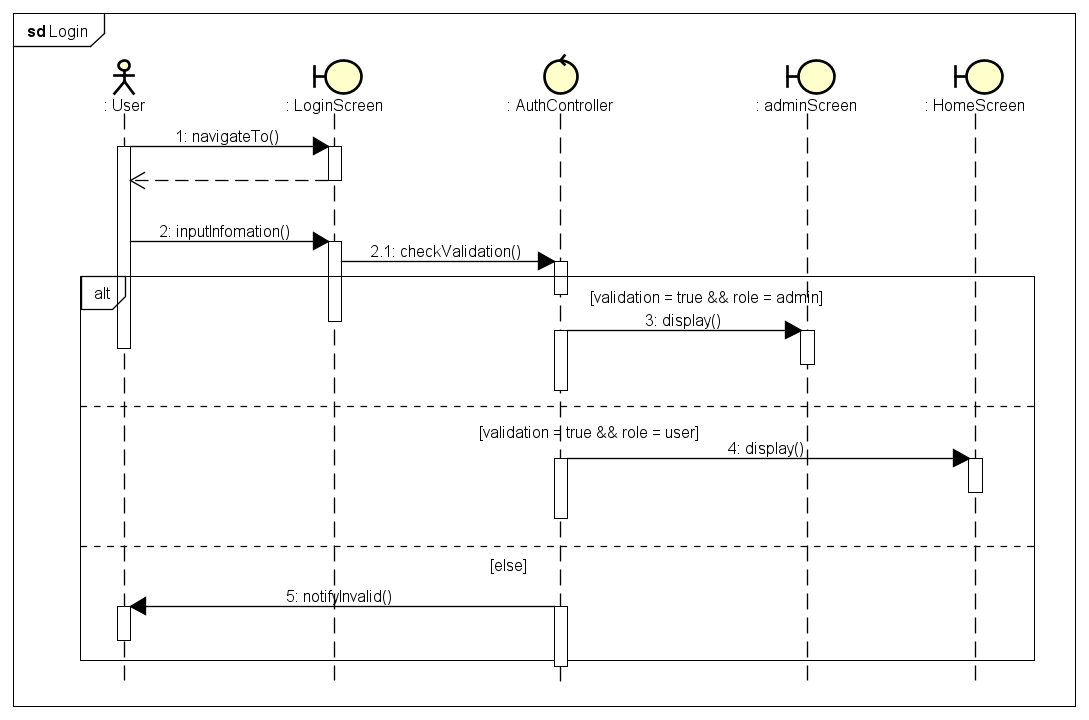
**Architectural design** begins by analyzing system requirements to identify key components. The system is then decomposed into modules based on functionality. A suitable architectural pattern (e.g., MVC) is selected to guide structure. Responsibilities and interactions among components are defined using class and sequence diagrams. Finally, the architecture is evaluated and refined to meet quality attributes.

## Interaction Diagrams

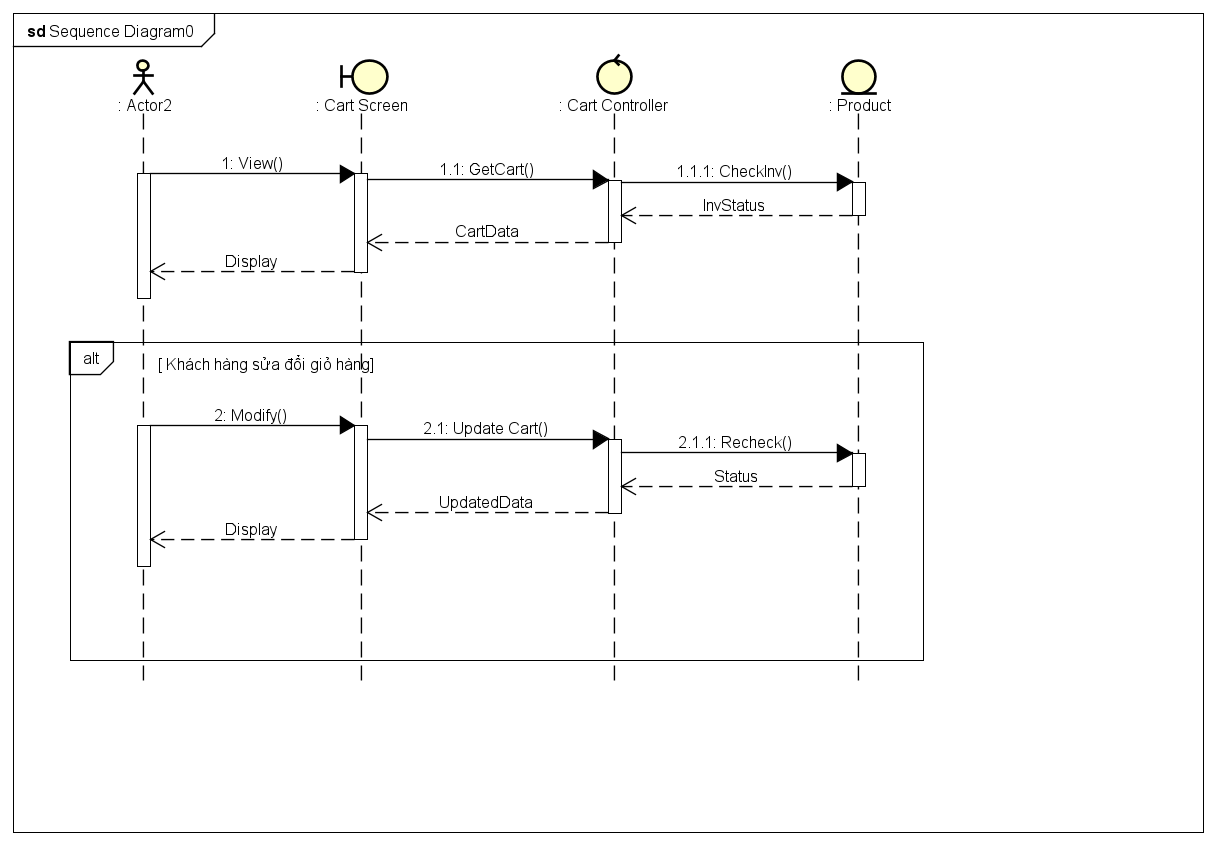
**Sequence Diagram for UC “Place Order"**



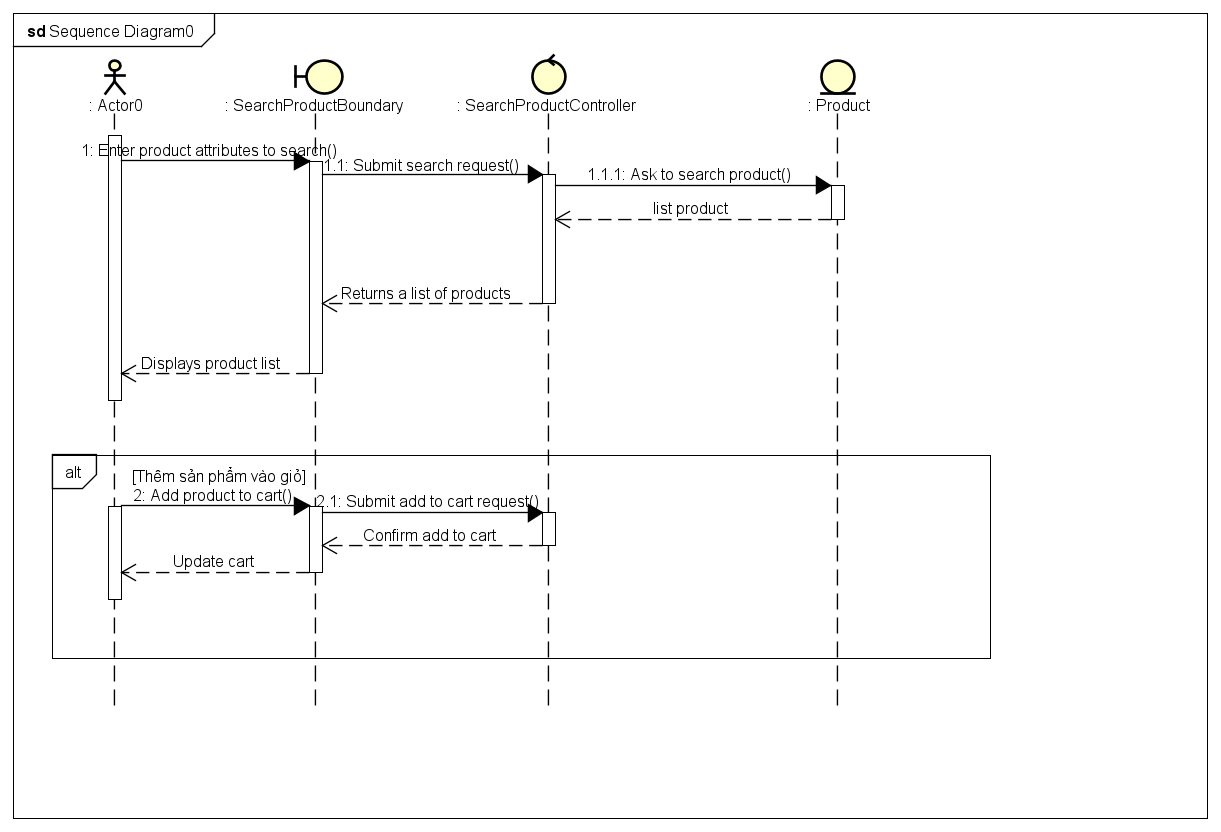
**Sequence Diagram for UC “Login"**



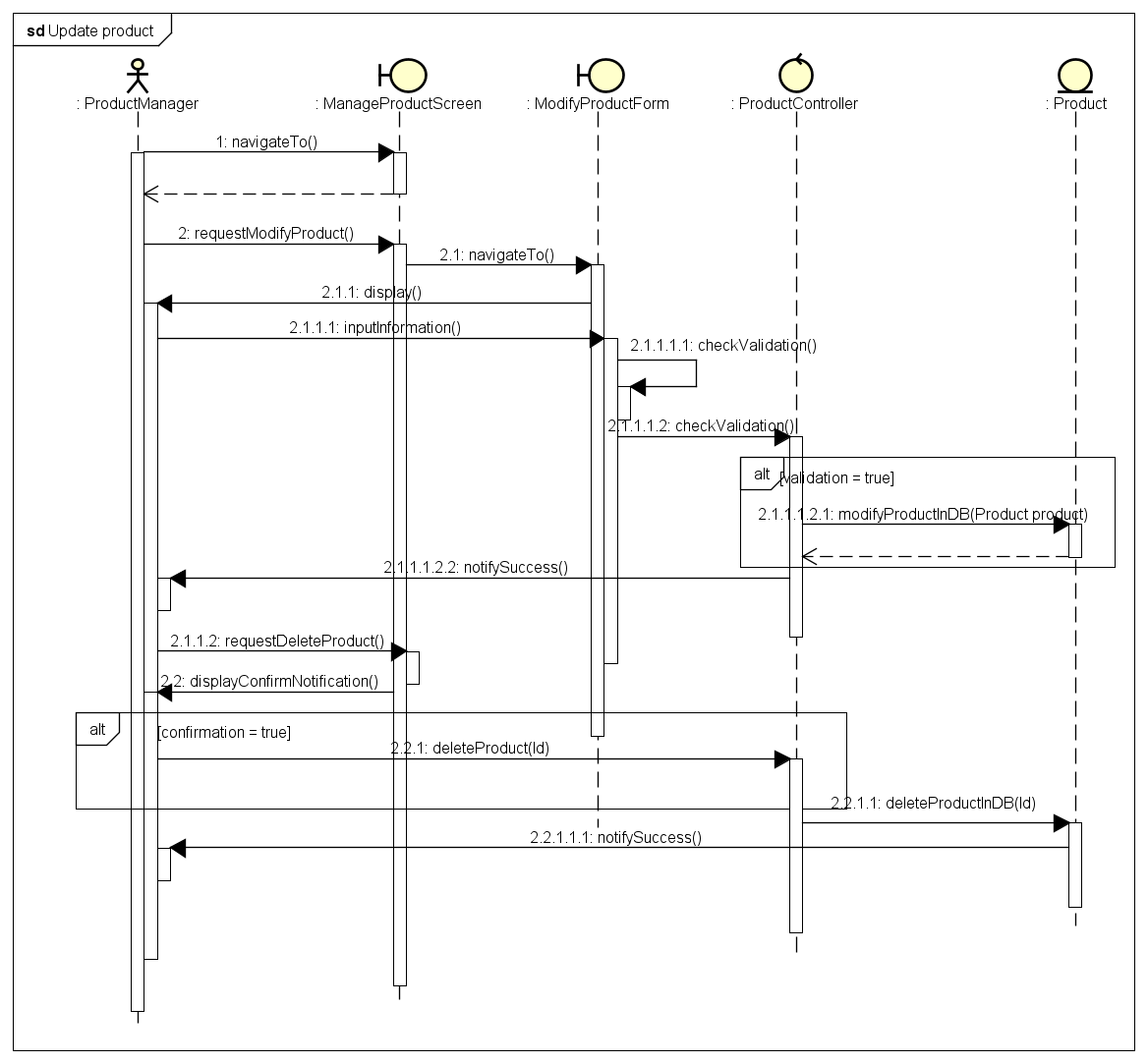
**Sequence Diagram for UC “View Cart"**



**Sequence Diagram for UC “Search Product"**



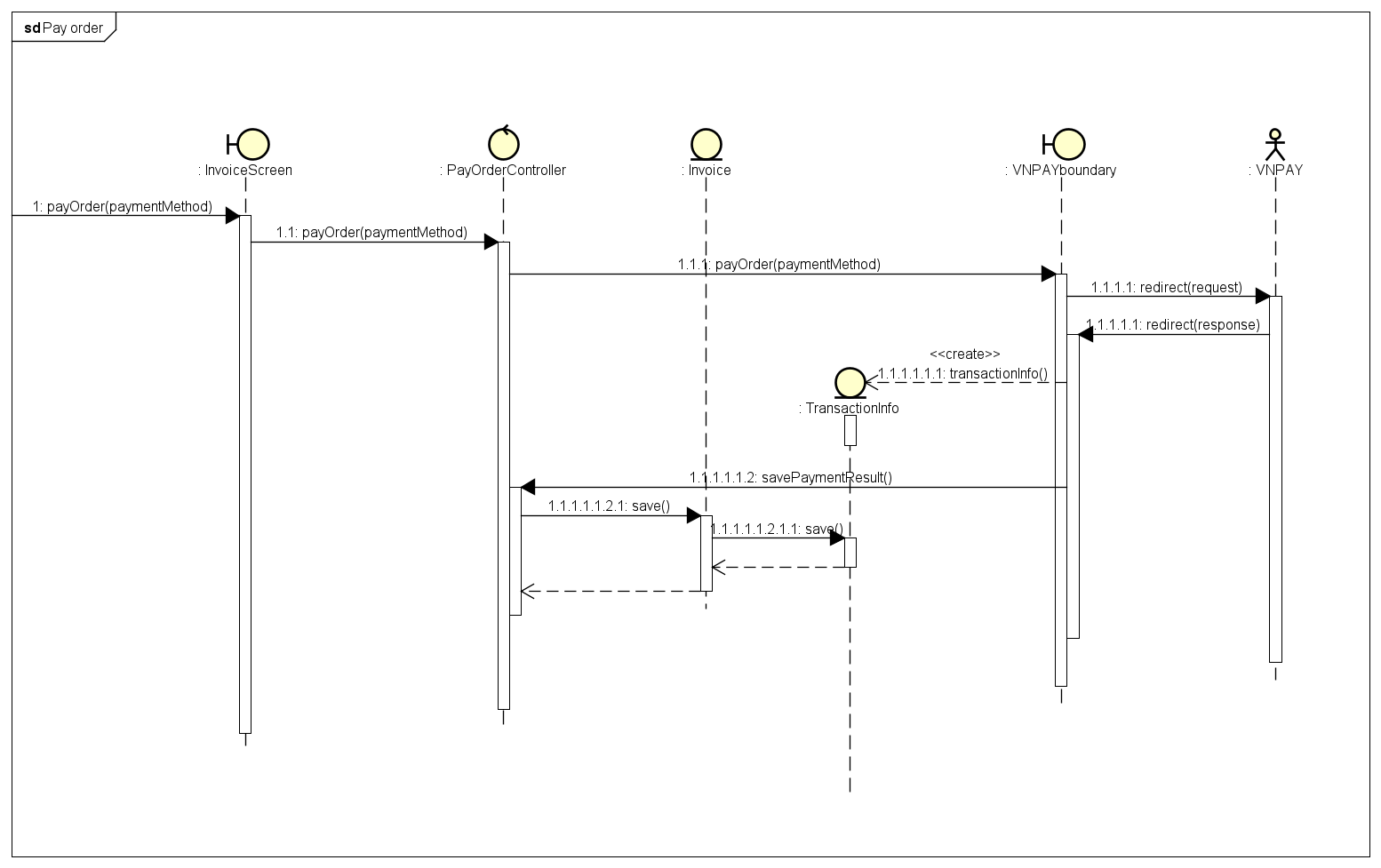
**Sequence Diagram for UC “Update Product"**



**Sequence Diagram for UC “Add Product"**

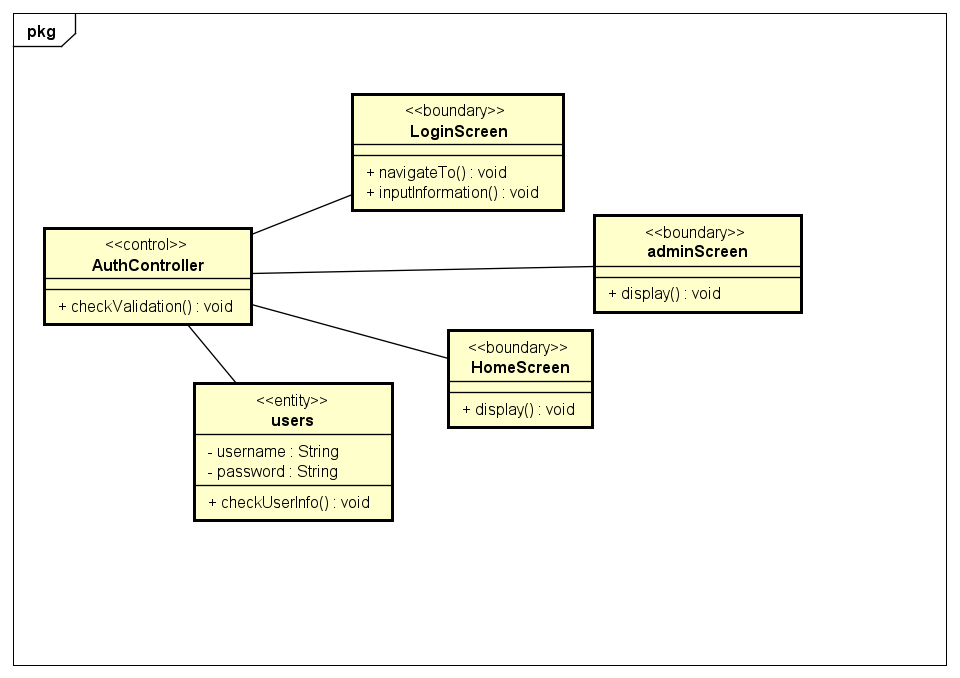


**Sequence Diagram for UC “Pay Order"**

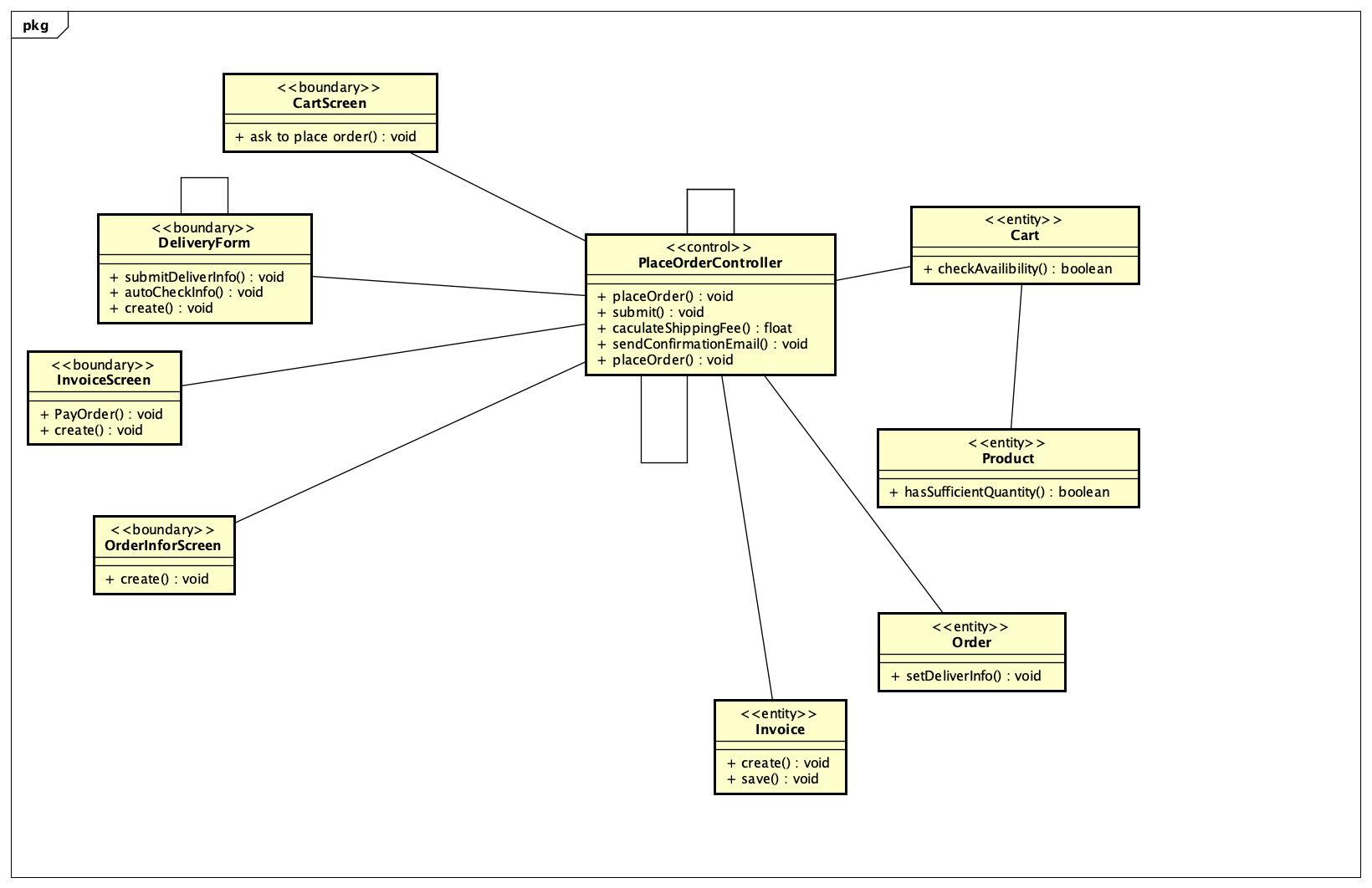


## Analysis Class Diagrams

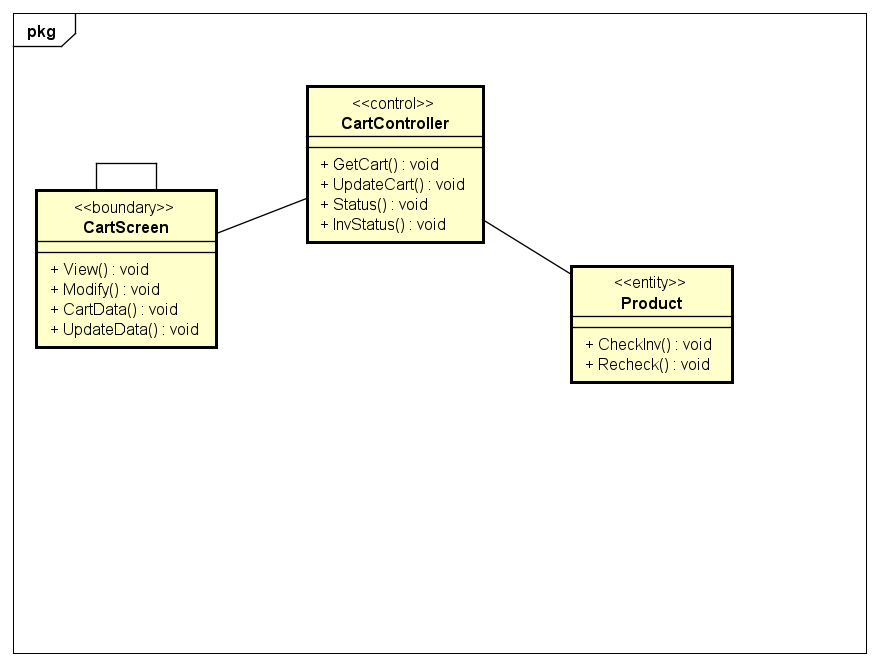
Analysis Class Diagram – Login



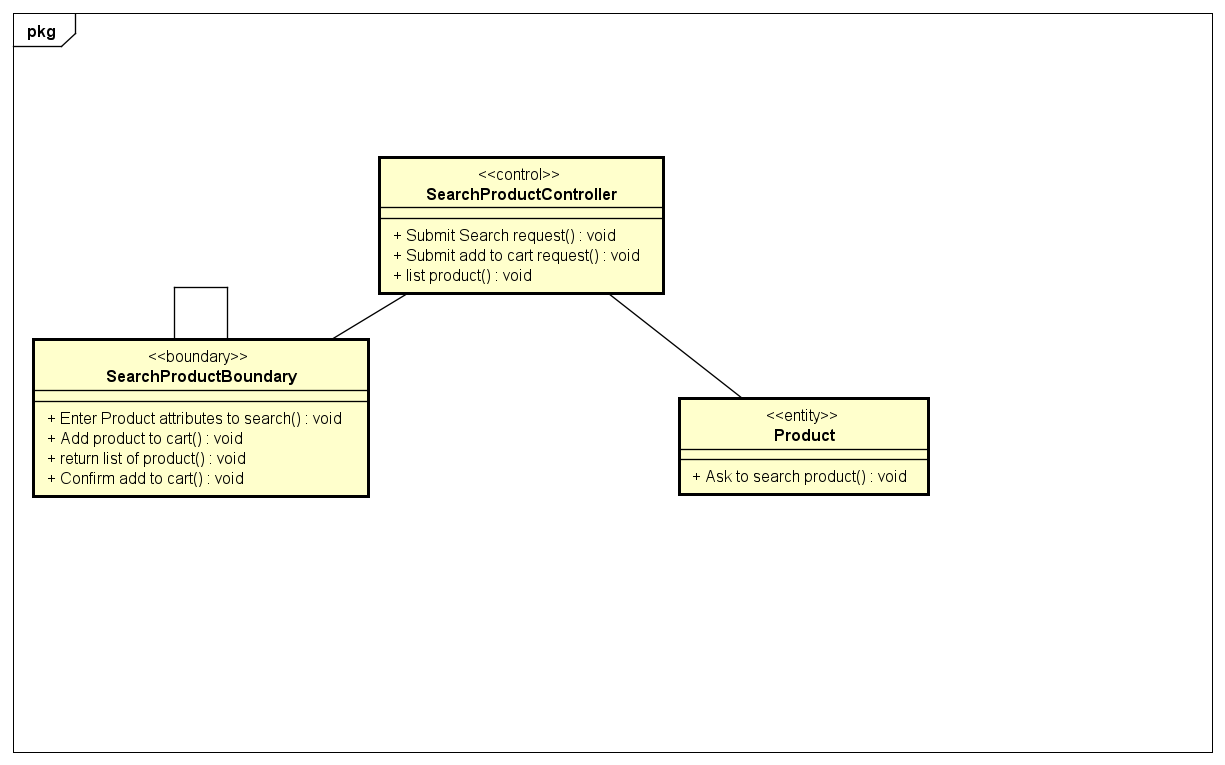
**Analysis Class Diagram - Place Order**



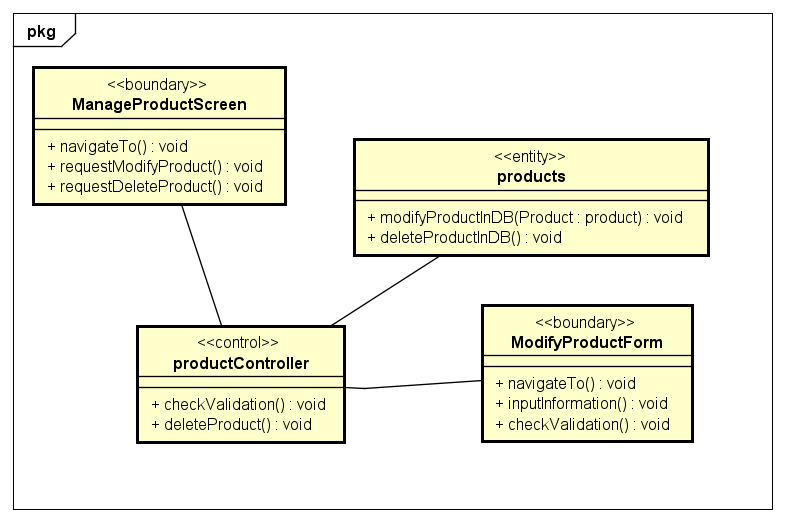
**Analysis Class Diagram - View Cart**



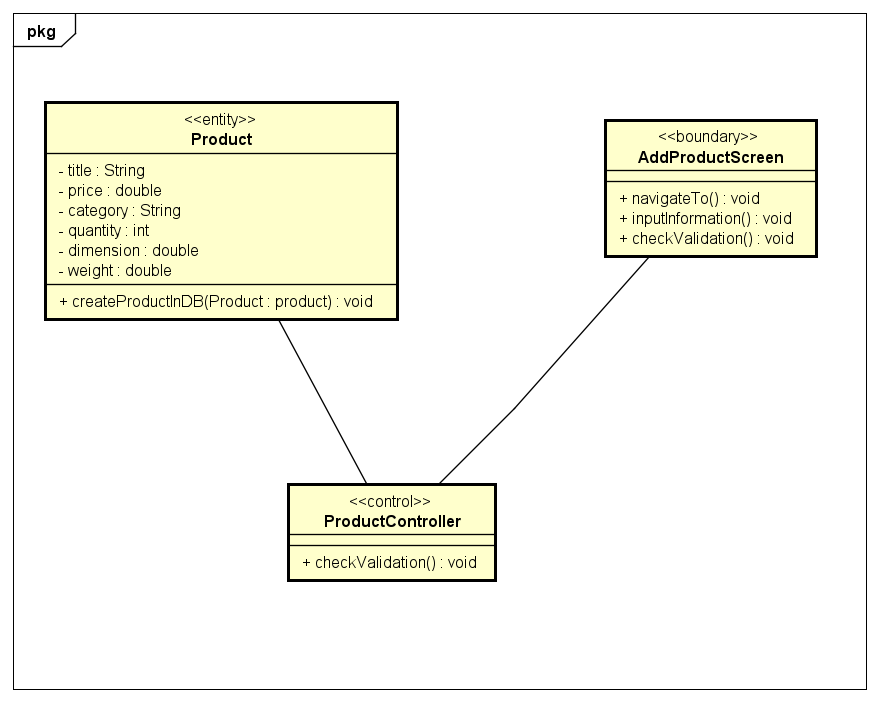
**Analysis Class Diagram - Search Product**



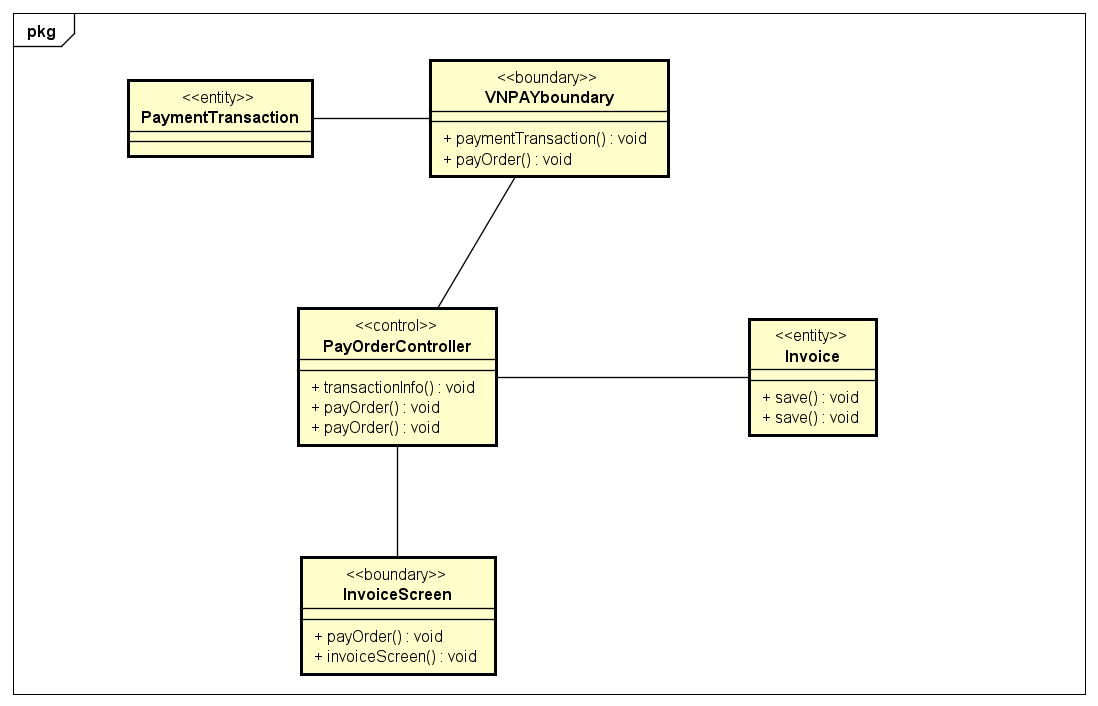
**Analysis Class Diagram - Update Product**



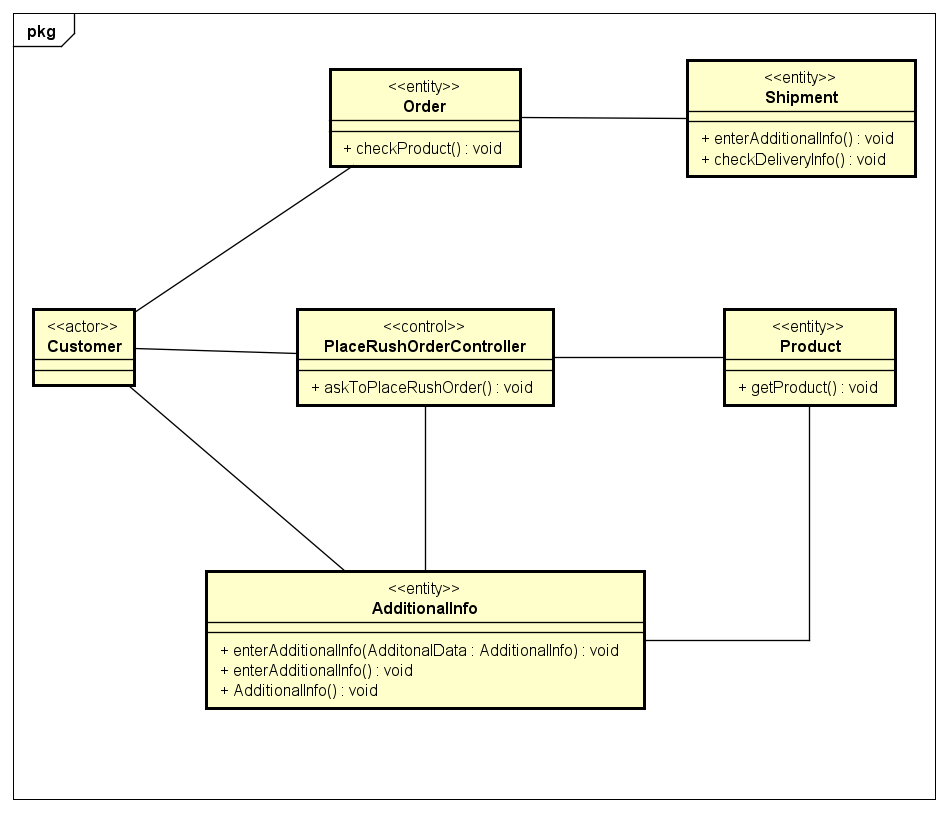
**Analysis Class Diagram - Add Product**



**Analysis Class Diagram - Pay Order**



**Analysis Class Diagram - Place Rush Order**



## Unified Analysis Class Diagram

## Security Software Architecture

*<Describe the software components and configuration supporting the security and privacy of the system. Specify the architecture for (1) authentication to validate user identity before allowing access to the system;(2) authorization of users to perform functional activity once logged into the system, (3) encryption protocol to support the business risks and the nature of information, and (4) logging and auditing design, if required.>*

# Detailed Design

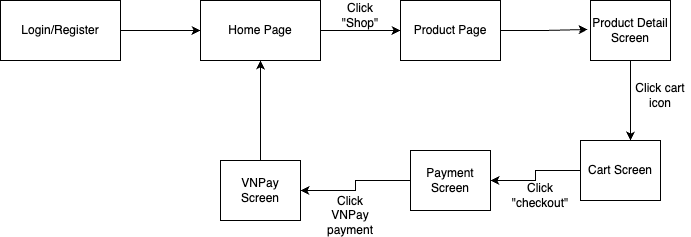
## User Interface Design

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

### Screen Configuration Standardization

### Screen Transition Diagrams

**Screen Transition for Customer**



### Screen Specifications

***Login Screen:***



Specification:

|  |  |  |  |
| --- | --- | --- | --- |
| Aims Software |  | Date of creation | Person in charge |
| Screen Specification | Login Screen | 15/5/2025 | Thai Duc |

|  |  |  |
| --- | --- | --- |
| Control | Operation | Function |
| Area | Initial | Display login form including username, password fields |
| Text Field  Text Field | Input  Input | Input username  Input password |
| Button | Click | Submit credentials to log in |
| Link | Click | Redirect user to account registration screen |

Field Attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen name** | **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Login | Username | 50 | Character | Black | Left justified |
| Login | Password | 50 | Character | Hidden input (black) | Left justified |

***Home Screen:***

Specification:

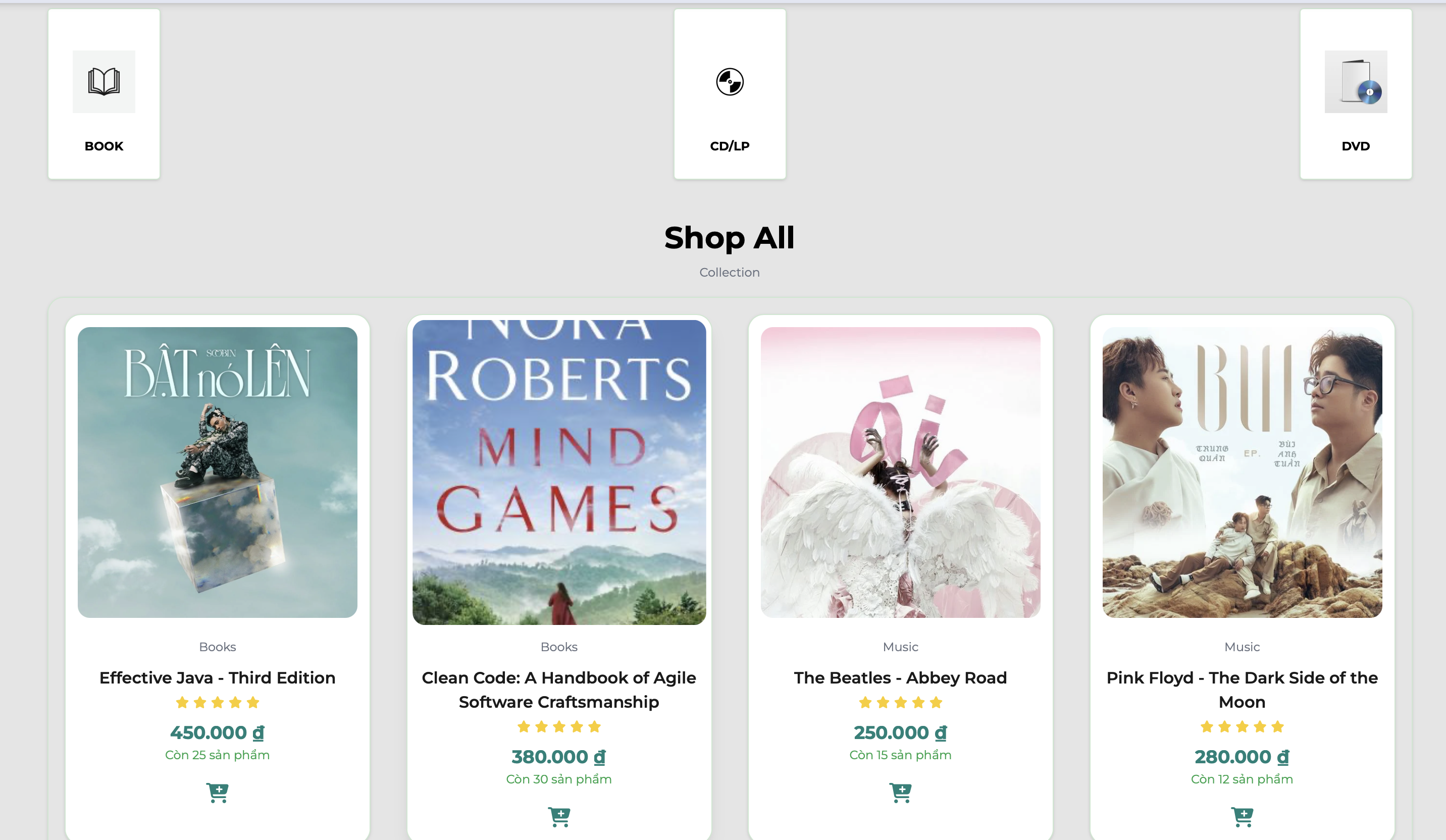
|  |  |  |  |
| --- | --- | --- | --- |
| Aims Software |  | Date of creation | Person in charge |
| Screen Specification | Home Screen | 15/5/2025 | Thai Duc |

|  |  |  |
| --- | --- | --- |
| **Control** | **Operation** | **Function** |
| **Search Bar** | Input | Enter product keywords to search |
| **Navigation Menu** | Click | Navigate to Home, Shop, Cart, Login, Register |
| **Cart Icon** | Click | View shopping cart contents (with item count) |
| **Category Button** | Click | Select a category (Books, Music, Movies) to filter product list |
| **Featured Section** | View only | Display 20 randomly selected featured products on each page reload |

Field Attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen name** | **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Home | Search Text Input | 100 | Character | Black | Center justified |
| Home | Category Label | 30 | Character | Black | Center aligned |
| Home | Cart Count | 3 | Numeral | Red badge | Right aligned |
| Home | Featured Product List | – | Composite | Dynamic display | 20 items per load |

***Product Screen:***



Specification:

|  |  |  |  |
| --- | --- | --- | --- |
| Aims Software |  | Date of creation | Person in charge |
| Screen Specification | Product Screen | 15/5/2025 | Thai Duc |

|  |  |  |
| --- | --- | --- |
| **Control** | **Operation** | **Function** |
| **Category Filter** | Click | Filter products by category: Book, CD/LP, DVD |
| **Product Card** | View | Display product image, title, price, rating, and stock quantity |
| **Add to Cart Icon** | Click | Add the selected product to the cart |
| **Shop All Label** | View | Display label indicating complete collection of items |

Field Attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen name** | **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Product | Product Title | 100 | Character | Black | Center justified |
| Product | Product Price | 10 | Numeric | Green | Currency format (₫) |
| Product | Rating | 1–5 | Star icon | Yellow icon | Display average rating |
| Product | Quantity Available | 10 | Numeric | Green text | “X products available” |
| Product | Add to Cart Icon | – | Button/Icon | Green cart icon | Adds 1 unit to cart |

***Cart Screen:***



Specification:

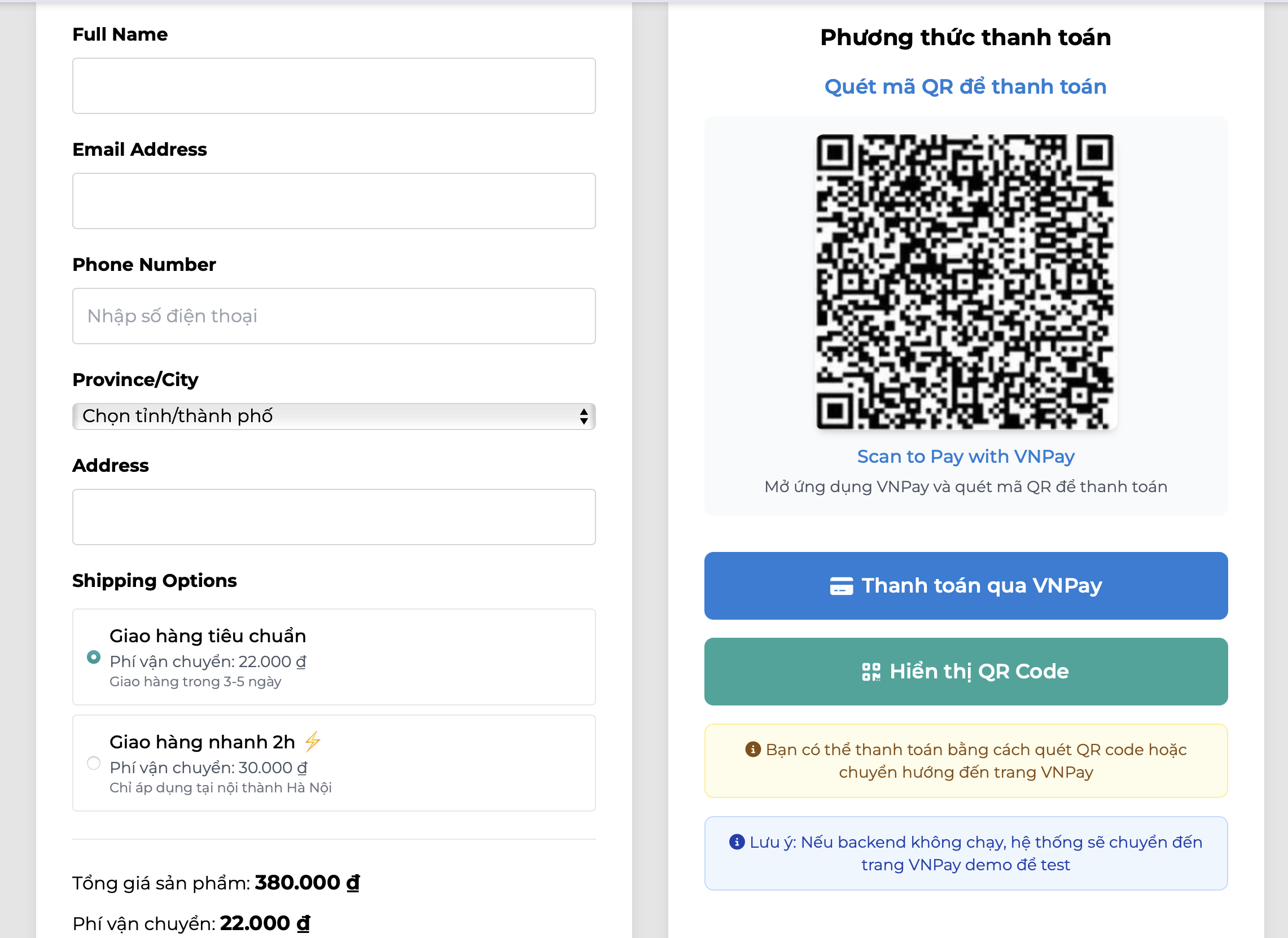
|  |  |  |  |
| --- | --- | --- | --- |
| Aims Software |  | Date of creation | Person in charge |
| Screen Specification | Cart Screen | 15/5/2025 | Thai Duc |

|  |  |  |
| --- | --- | --- |
| **Control** | **Operation** | **Function** |
| **Product Row** | View | Display product image, name, price, quantity, and subtotal |
| **Remove Button** | Click | Remove product from cart |
| **Quantity Buttons (+/-)** | Click | Increase or decrease the quantity of a product |
| **Continue Shopping** | Click | Return to the product browsing screen |
| **Checkout Button** | Click | Proceed to payment screen |
| **Cart Summary Box** | View | Show total price, shipping fee, and grand total |

Field Attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen name** | **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Cart | Product Title | 100 | Character | Black | Left aligned |
| Cart | Product Image | – | Image | Thumbnail | Fixed size |
| Cart | Quantity | 2 | Numeric | Editable | Uses (+/-) buttons |
| Cart | Unit Price | 10 | Currency | Black | Formatted in ₫ |
| Cart | Subtotal | 10 | Currency | Green text | Unit price × quantity |
| Cart | Shipping Fee | 20 | Character | Gray text | “Free” or calculated |
| Cart | Total Amount | 10 | Currency | Green, bold | Final amount to pay |

***Payment Screen:***



Specifications:

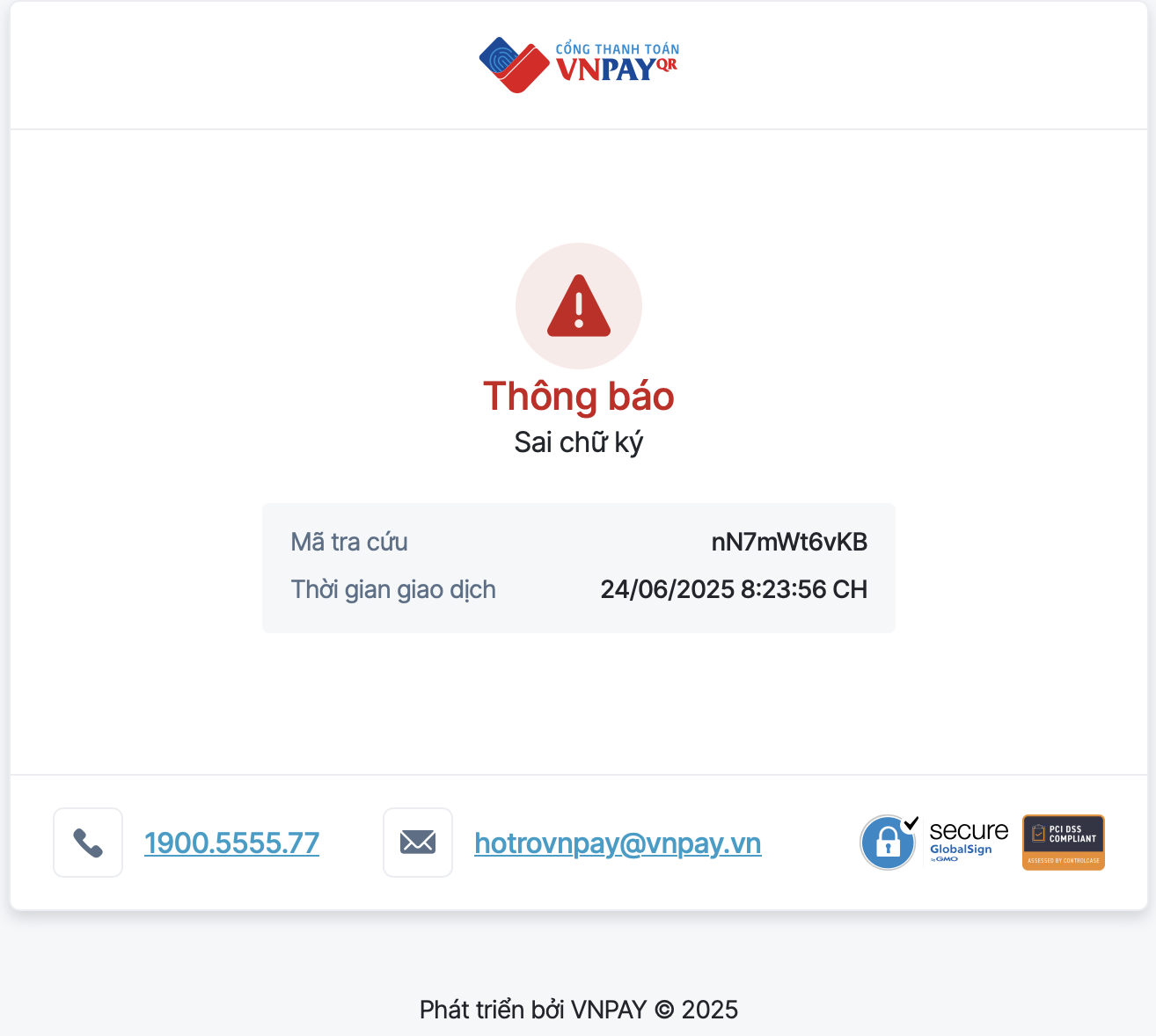
|  |  |  |  |
| --- | --- | --- | --- |
| Aims Software |  | Date of creation | Person in charge |
| Screen Specification | Payment Screen | 15/5/2025 | Thai Duc |

|  |  |  |
| --- | --- | --- |
| **Control** | **Operation** | **Function** |
| **Text Fields** | Input | Input user information: Full Name, Email, Phone Number, Address |
| **Dropdown** | Select | Select Province/City |
| **Shipping Option** | Select | Choose between standard or express delivery |
| **Price Summary** | View | Show product price, shipping fee, and total amount |
| **QR Code Area** | Display | Show QR Code for VNPay payment |
| **VNPay Button** | Click | Redirect to VNPay gateway for online payment |
| **QR Code Button** | Click | Toggle QR code display for manual scan via VNPay app |
| **Notifications** | View | Display warnings or notes regarding payment and backend environment |

Field Attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Screen name** | **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Payment | Full Name | 100 | Character | Required | User’s full name |
| Payment | Email Address | 100 | Character | Required | Valid email format |
| Payment | Phone Number | 20 | Numeric | Required | Vietnamese phone number format |
| Payment | Province/City | – | Dropdown | Required | Predefined list |
| Payment | Address | 200 | Character | Required | User delivery address |
| Payment | Shipping Option | – | Radio Button | Required | Standard or Express |
| Payment | Product Price | 10 | Currency | Black text | Unit total |
| Payment | Shipping Fee | 10 | Currency | Black text | Depends on shipping selection |
| Payment | Total Price | 10 | Currency | Green bold text | Final price to pay |
| Payment | QR Code | – | Image | QR Code | VNPay payment |
| Payment | Pay with VNPay Button | – | Button | Blue | Redirect to gateway |
| Payment | Show QR Button | – | Button | Green | Toggle QR display |

***VNPay Screen:***



Specification:

|  |  |  |  |
| --- | --- | --- | --- |
| Aims Software |  | Date of creation | Person in charge |
| Screen Specification | VNPay Screen | 15/5/2025 | Thai Duc |

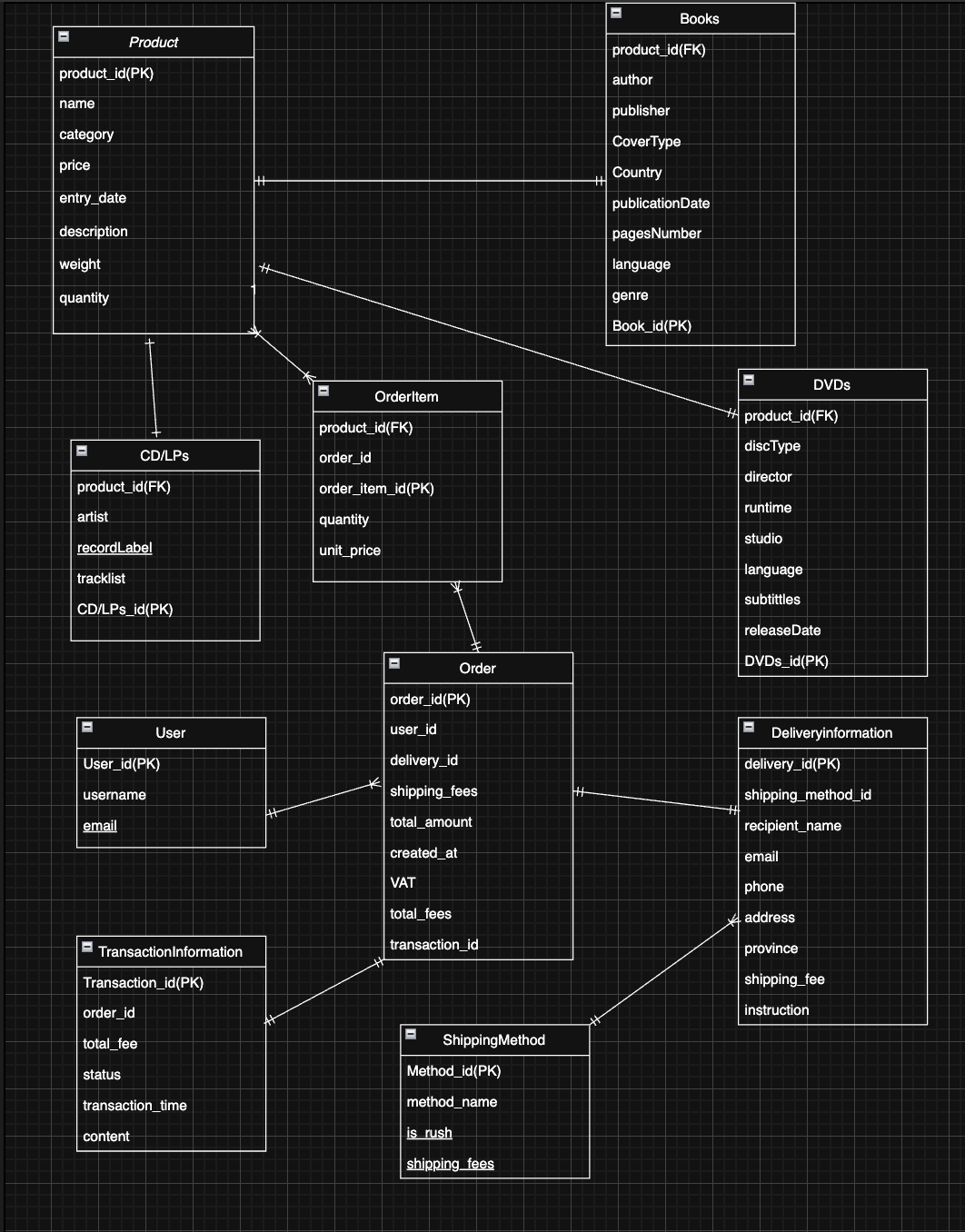
|  |  |  |
| --- | --- | --- |
| **Control** | **Operation** | **Function** |
| **Error Message** | View | Displays the type of error (e.g., “Invalid signature”) |
| **Transaction ID** | View | Shows the transaction reference code for user inquiry |
| **Timestamp** | View | Displays the exact date and time of the failed transaction |
| **Phone Contact** | Click | Allows user to call VNPay support |
| **Email Contact** | Click | Opens default mail client to send a support request to VNPay |

Field Attributes:

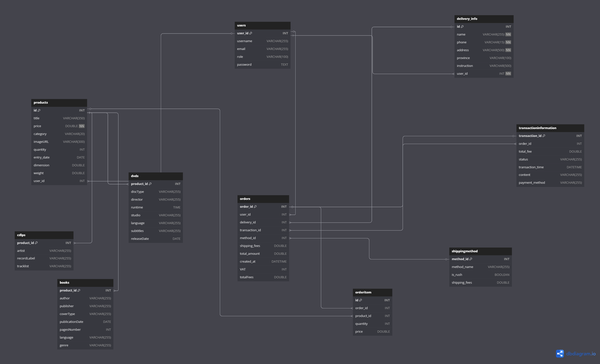
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen name** | **Item name** | **Type** | **Field attribute** | **Remarks** |
| VNPay | Error Message | Character | Red text, bold | “Signature mismatch” or similar |
| VNPay | Transaction ID | Alphanumeric | Gray text box | For customer support reference |
| VNPay | Transaction Time | Datetime | Gray text box | Full timestamp of the error |
| VNPay | Phone Number | Link | Blue clickable text | Support hotline |
| VNPay | Email Address | Link | Blue clickable text | Support email for issues |

## Data Modeling

### Conceptual Data Modeling



### Database Design



#### Database Management System

*<Specify what is the decision of Database Management System (DBMS) and give some description of the DBMS>*

#### Database Diagram

<

* *Show the process to design database from E-R diagram*
* *Show the diagram of DB design*

*>*

#### Database Detail Design

Table 1: Users

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Extra** |
| user\_id | int | PRI | auto\_increment |
| username | varchar(255) |  |  |
| email | varchar(255) |  |  |
| role | varchar(100) |  |  |
| password | text |  |  |

Table 2: Transactioninformation

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Description** |
| transaction\_id | int | PRI | Primary key, unique transaction ID |
| order\_id | int | MUL | Foreign key (assumed to reference orders) |
| total\_fee | double |  | Total amount paid |
| status | varchar(255) |  | Status of the transaction (e.g., “paid”, “pending”) |
| transaction\_time | datetime |  | Time of the transaction |
| content | varchar(255) |  | Description or note about the transaction |
| payment\_method | varchar(255) |  | Payment method (e.g., “VNPay”, “card”) |

Table 3: Shippingmethod

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Description** |
| method\_id | int | PRI | Primary key, unique delivery method ID |
| method\_name | varchar(255) |  | Name of the delivery method (e.g., “Standard”, “Express”) |
| is\_rush | tinyint(1) |  | Whether it’s a rush delivery (0 = No, 1 = Yes) |
| shipping\_fees | double |  |  |

Table 4: products

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Description** |
| id | int | PRI | Primary key, unique product ID |
| title | varchar(350) |  |  |
| price | double |  |  |
| category | varchar(20) |  | Product category (e.g., book, DVD) |
| imageURL | varchar(300) |  | URL to product image |
| quantity | int |  | Quantity in stock |
| entry\_date | date |  | Date the product was added to the system |
| dimension | double |  |  |
| weight | double |  | Product weight |
| user\_id | int | MUL | Foreign key reference to users(user\_id) |

Table 5: Orders

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Description** |
| order\_id | int | PRI | Primary key, unique identifier for the order |
| user\_id | int | MUL | Foreign key to users(user\_id) – the customer placing the order |
| delivery\_id | int | MUL | Foreign key to a delivery record (if using a delivery table) |
| transaction\_id | int | MUL | Foreign key to the transactions table |
| method\_id | int | MUL | Foreign key to delivery\_methods(method\_id) |
| shipping\_fees | double |  | Shipping cost |
| total\_amount | double |  | Total cost of items before taxes/shipping |
| created\_at | datetime |  | Order creation timestamp |
| VAT | int |  | VAT rate or amount (depending on usage) |
| totalFees | double |  | Grand total including all fees (e.g., total\_amount + shipping + VAT) |

Table 6: Orderitems

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Key** | **Description** |
| id | int | PRI | Primary key, auto-increment |
| order\_id | int | MUL | Foreign key linking to orders(order\_id) |
| product\_id | int | MUL | Foreign key linking to products(id) |
| quantity | int |  | Number of units of the product in this order |
| price | double |  | Price per unit (copied at time of order placement) |

Table 7: DVDs

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| product\_id | int | Primary key, also a foreign key to products(id) |
| discType | varchar(255) | Type of disc (e.g., DVD, Blu-ray) |
| director | varchar(255) | Director of the movie |
| runtime | time | Runtime duration (e.g., 01:45:00) |
| studio | varchar(255) | Production studio |
| language | varchar(255) | Language of the main audio track |
| subtitles | varchar(255) | Subtitle languages available |
| releaseDate | date | Date of release |

Tables 8:Books

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| product\_id | int | Primary key, foreign key to products(id) |
| author | varchar(255) | Author name |
| publisher | varchar(255) | Publisher name |
| coverType | varchar(255) | e.g., Paperback, Hardcover |
| publicationDate | date | Date of publication |
| pagesNumber | int | Total number of pages |
| language | varchar(255) | Language the book is written in |
| genre | varchar(255) | Book genre/category |

Tables 9: CDLPs

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| product\_id | int | Primary key, foreign key to products(id) |
| artist | varchar(255) | Main performing artist |
| recordLabel | varchar(255) | Label that published the album |
| tracklist | varchar(255) | Short track list or summary (may be extended later) |

Tables 10: Delivery Info

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Description** |
| id | int | NO | Primary key, unique address ID |
| name | varchar(255) | NO | Full name of the receiver |
| phone | varchar(15) | NO | Receiver’s phone number |
| address | varchar(500) | NO | Full address |
| province | varchar(100) | YES | Optional province/state |
| instruction | varchar(500) | YES | Optional delivery instructions (e.g., “Leave at door”) |
| user\_id | int | NO | Foreign key to users(user\_id) |

## Non-Database Management System Files

*<Provide the detailed description of all non-DBMS files if any and include a narrative description of the usage of each file that identifies if the file is used for input, output, or both, and if the file is a temporary file. Also provide an indication of which modules read and write the file and include file structures (refer to the data dictionary). As appropriate, the file structure information should include the following:*

*• Record structures, record keys or indexes, and data elements referenced within the records*

*• Record length (fixed or maximum variable length) and blocking factors*

*• Access method (e.g., index sequential, virtual sequential, random access, etc.)*

*• Estimate of the file size or volume of data within the file, including overhead resulting from file access methods*

*• Definition of the update frequency of the file (If the file is part of an online transaction-based system, provide the estimated number of transactions per unit of time, and the statistical mean, mode, and distribution of those transactions.)*

*• Backup and recovery specifications>*

## Class Design

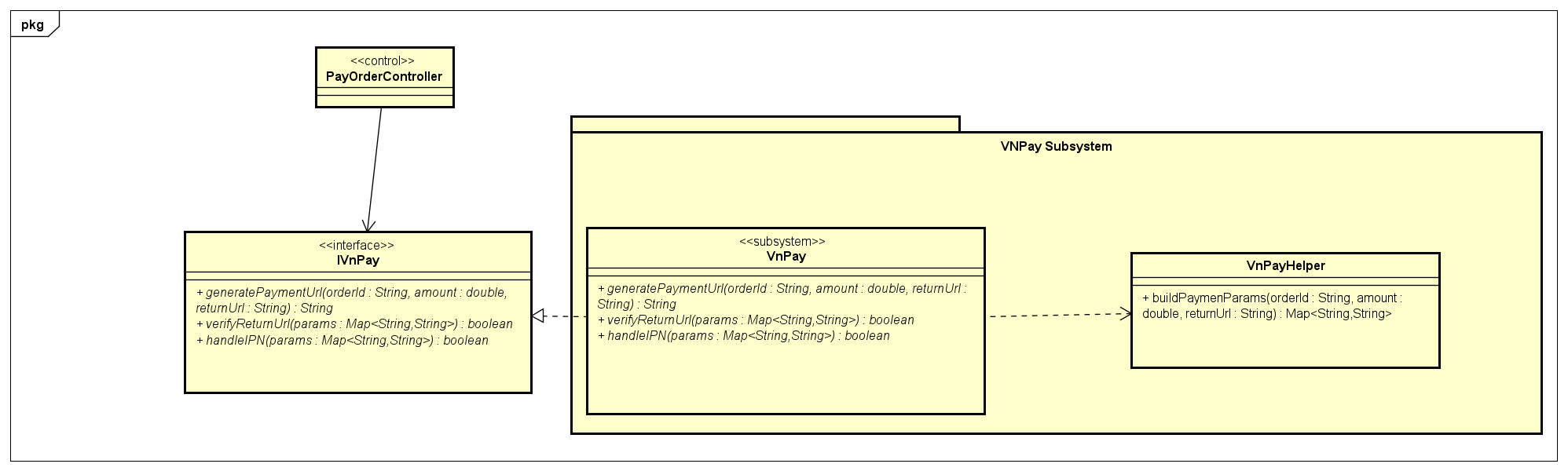
### General Class Diagram

<General class diagram which shows the whole class diagram of the software. This diagram may have packages, subsystems and classes. Classes in this diagram may not have all attributes and operations>

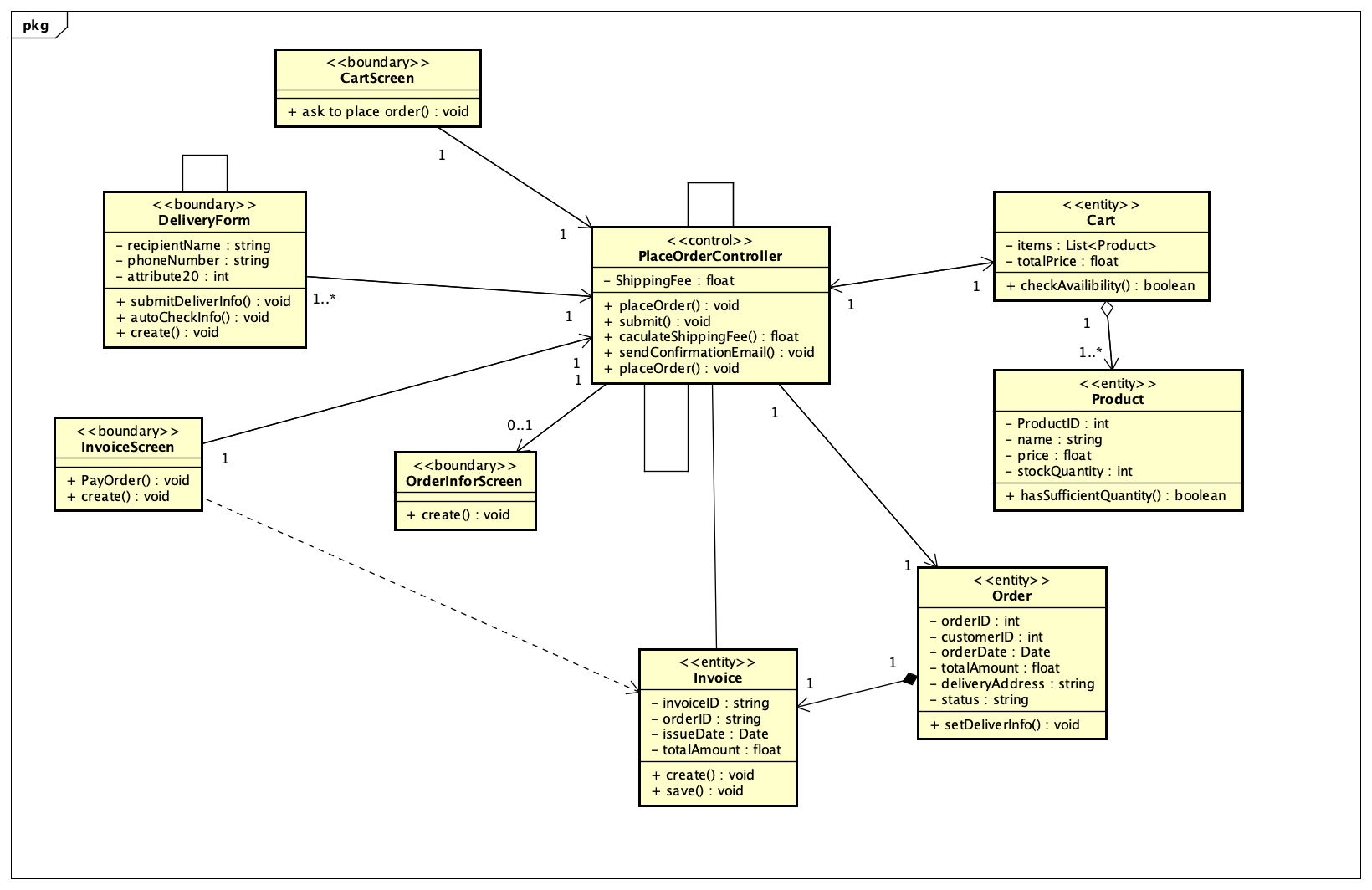
### Class Diagrams

<Detail class diagram with full attributes and operations>

#### Interface&Subsystem Class Diagram



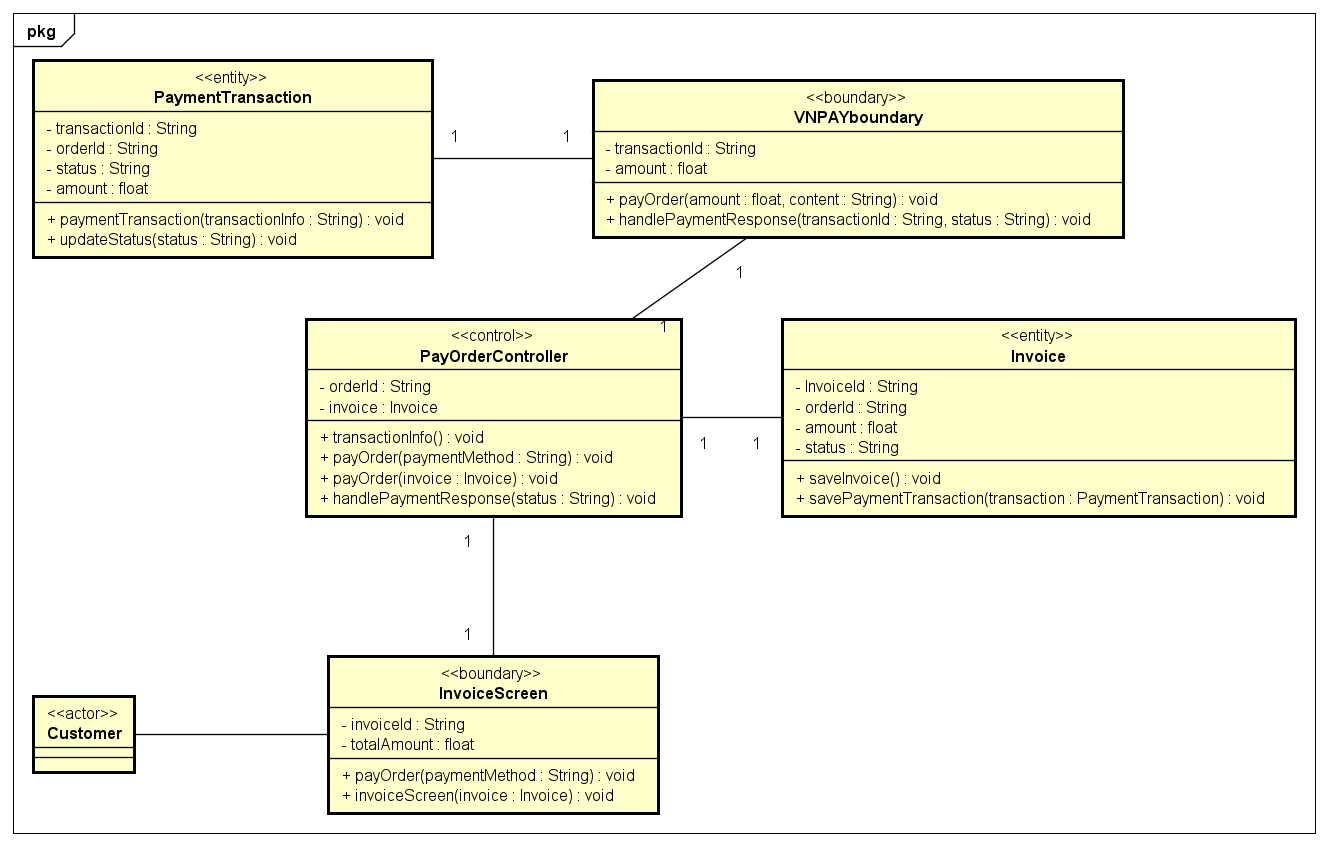
#### Class Diagram PlaceOrder



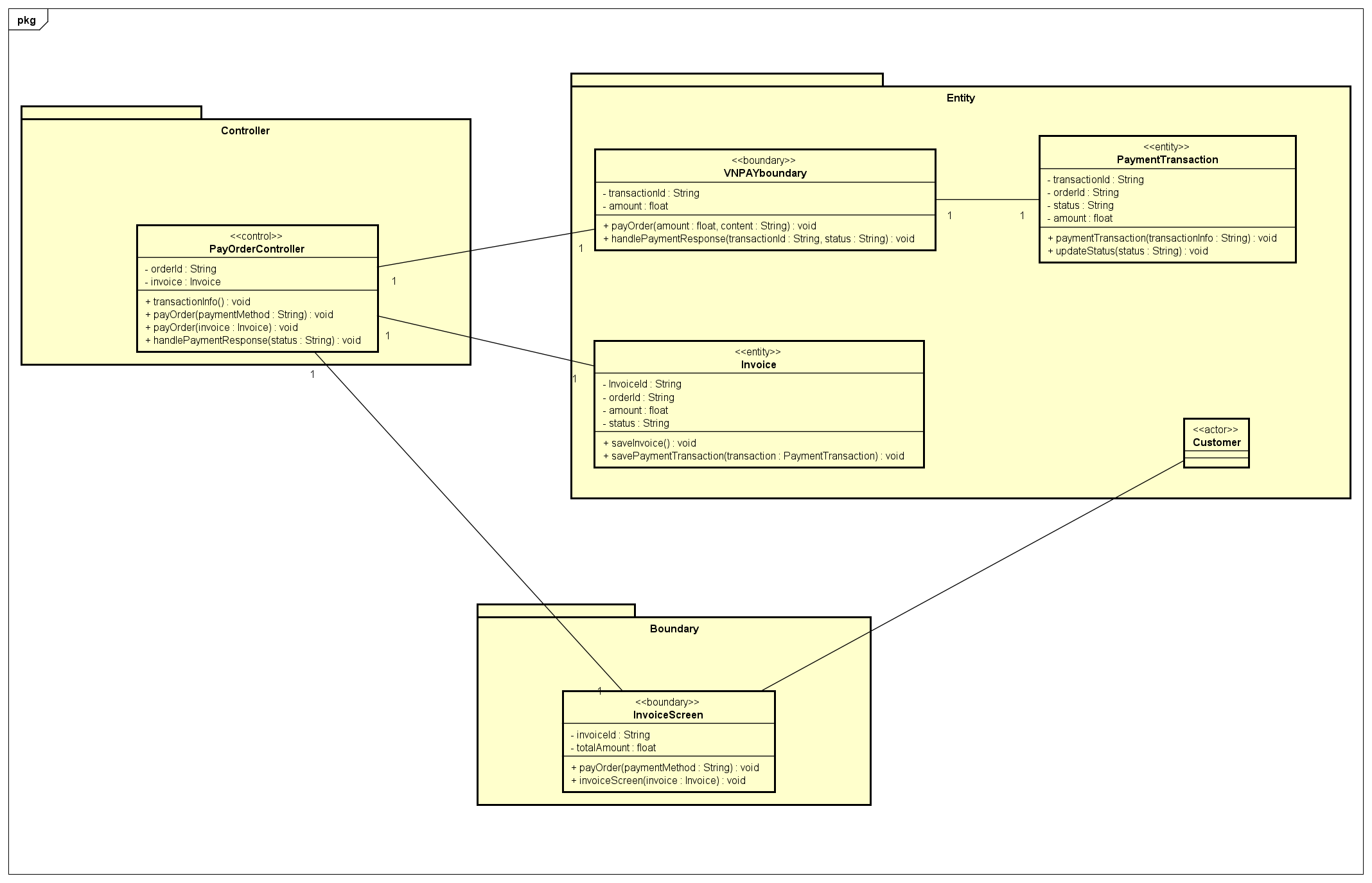
#### PackagePlaceOrder



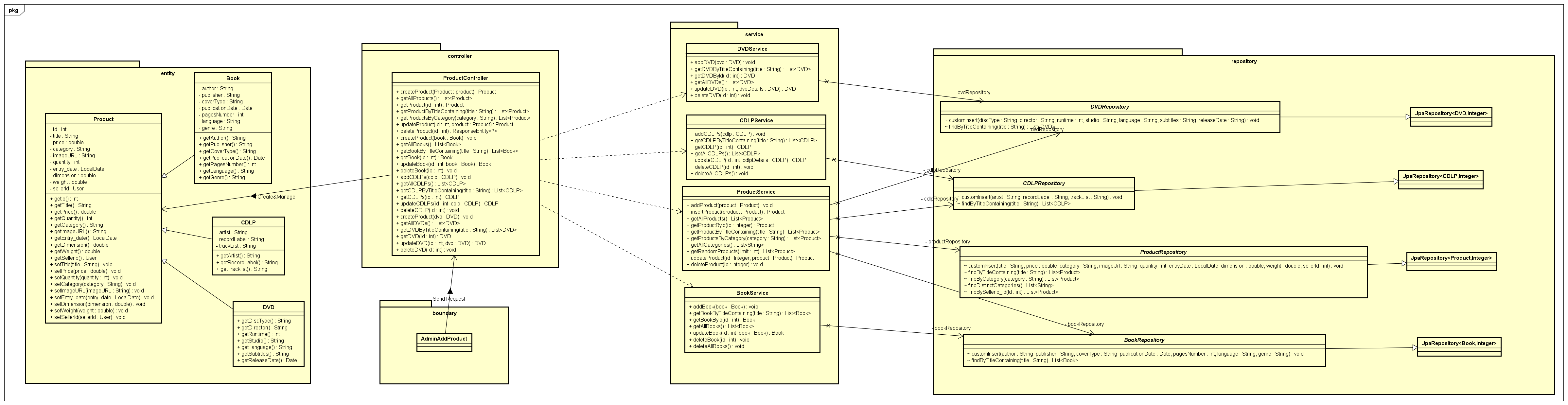
#### Class Diagram PayOrder



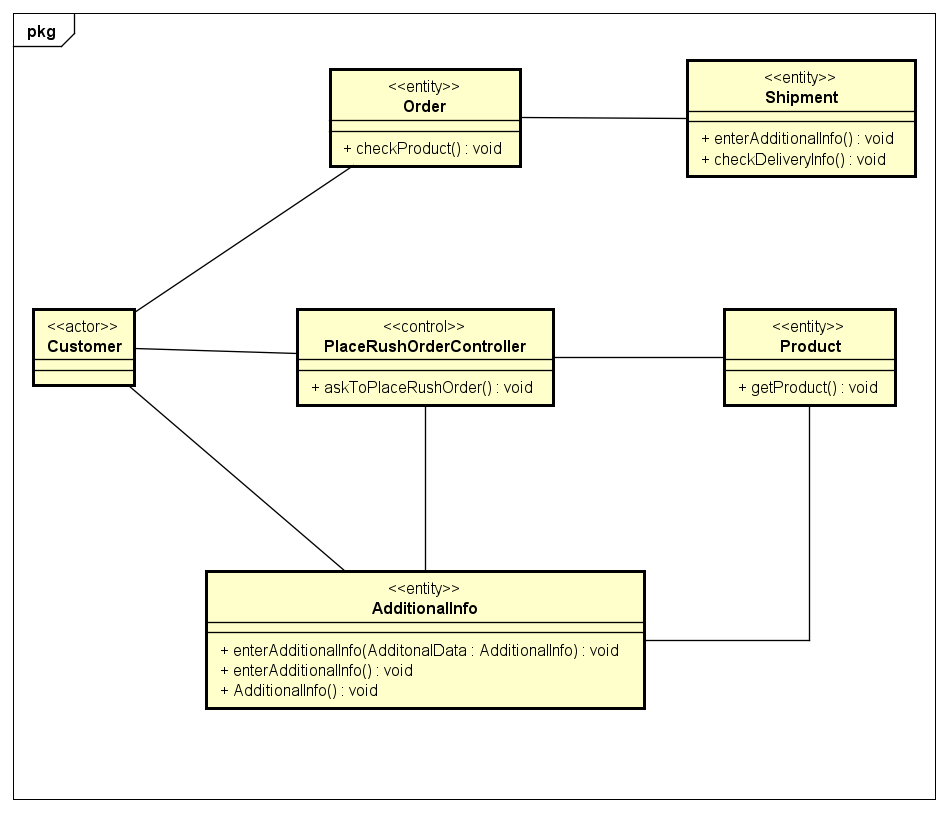
#### Package PayOrder



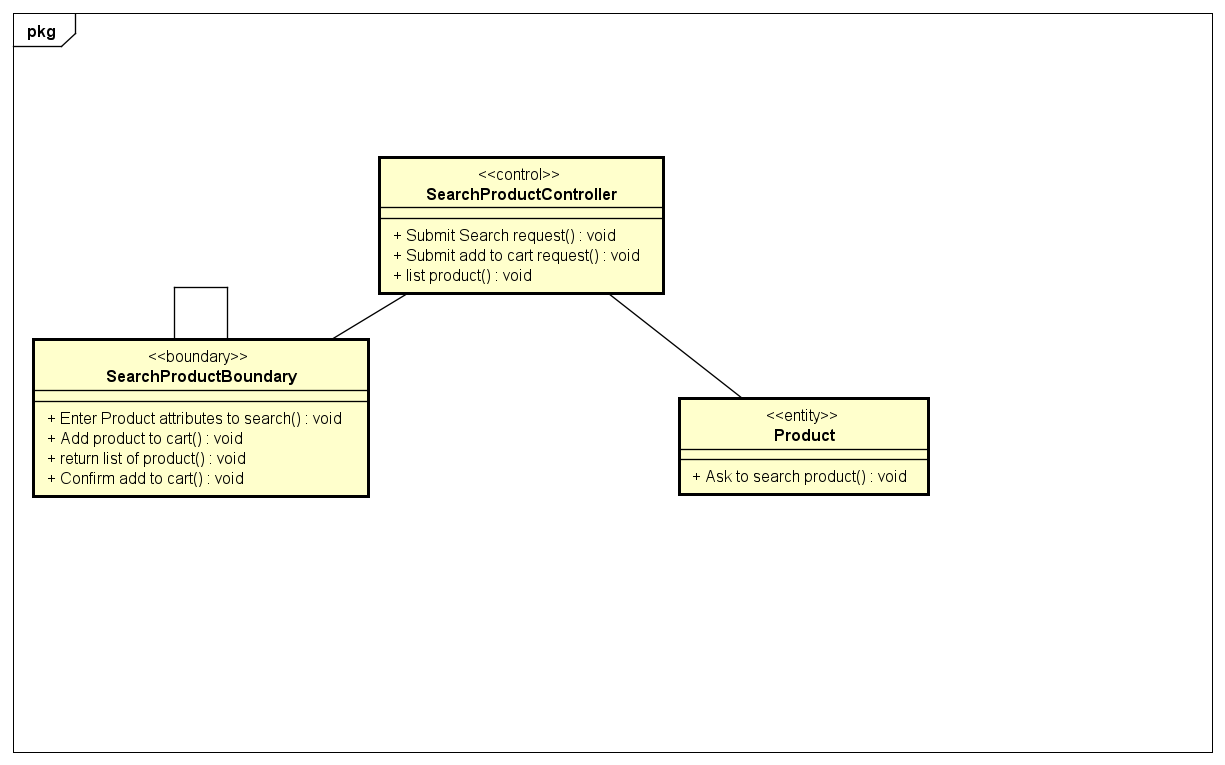
#### Class Diagram Add\_UpdateProduct



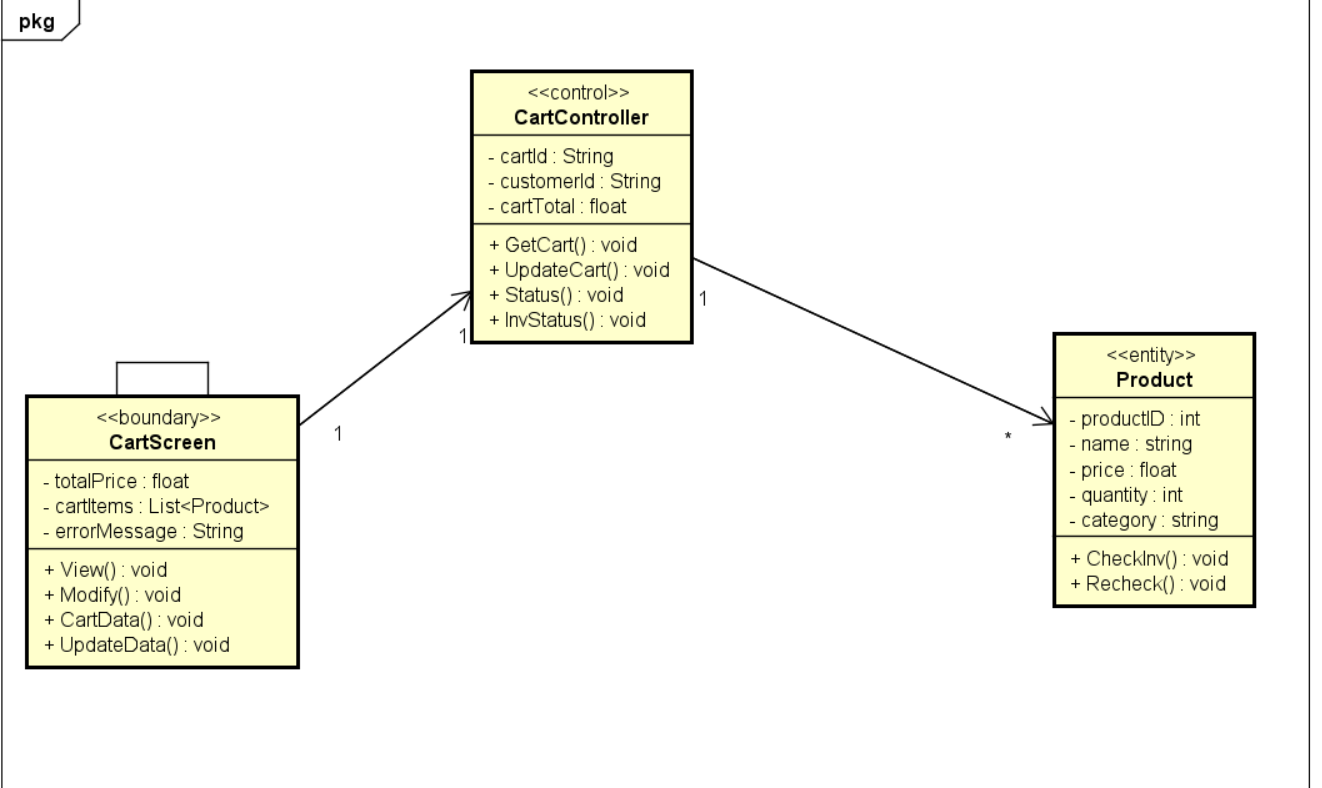
#### Class Diagram PlaceRushOrder



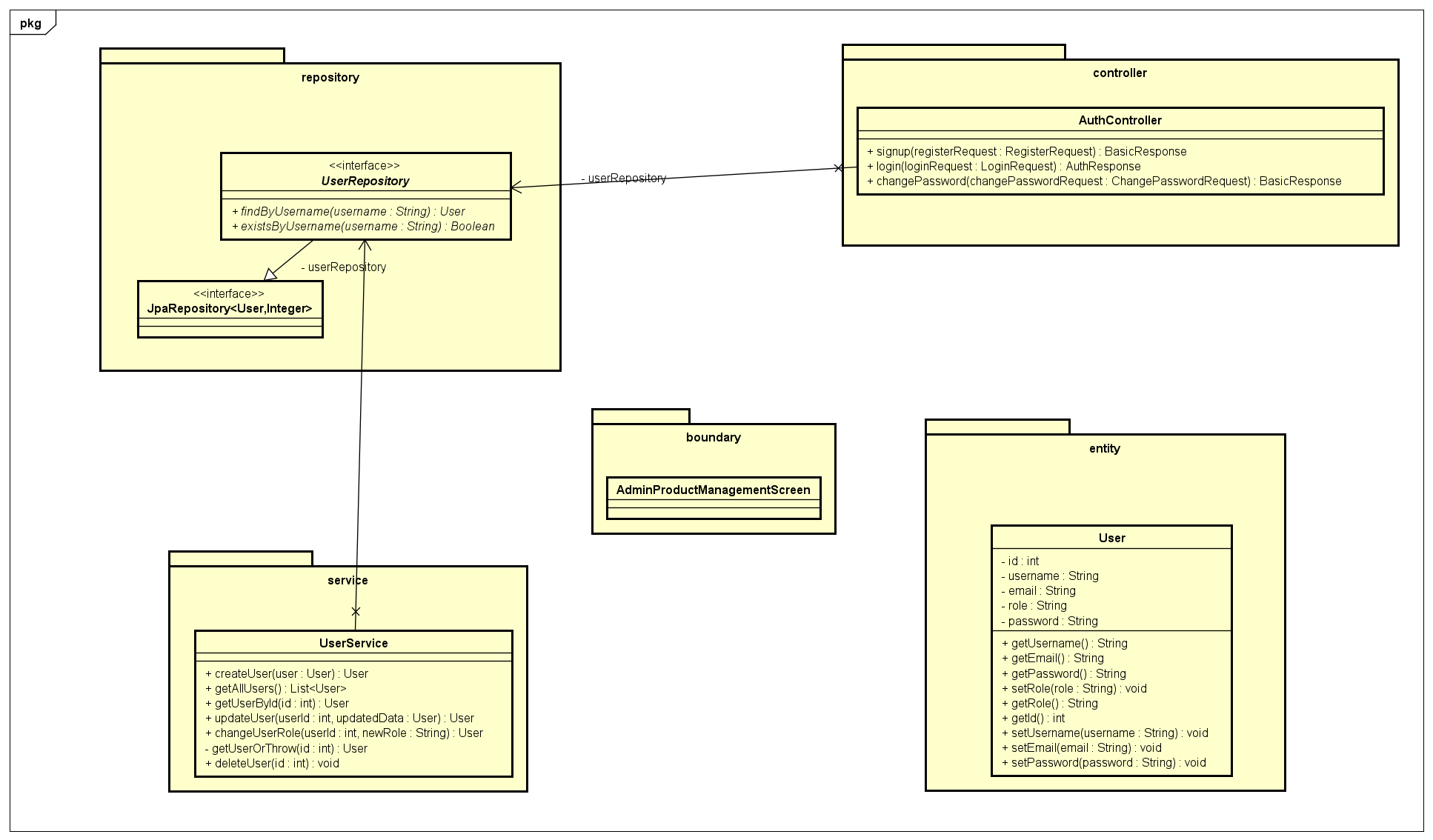
#### Class Diagram SearchProduct



**4.4.2.9 Class Diagram ViewCart**



### 4.4.2.10 Class Diagram Login



### Class Design

<Detail design for each class>

#### Class “SampleClass1”

<SampleClass1 class image in UML>

Table 2. Example of attribute design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 |  |  |  |  |
| 2 |  |  |  |  |

Table 3. Example of operation design

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 |  |  |  |
| 2 |  |  |  |

*Parameter*:

* x: Default value, description
* y: Default value, description

*Exception*:

* AException if …
* BException if …

**Method**

How to use parameters / attributes

Flowchart / activity diagram / sequence diagram if the method has a complex/special algorithm

**State**

State diagram if any

#### Class “SampleClass2”

…

# Design Considerations

***<Describe issues which need to be addressed or resolved before attempting to devise a complete design solution. Remember that, you have to refactor your source code to strictly follow the final design>***

## Goals and Guidelines

*<Describe any goals, guidelines, principles, or priorities which dominate or embody the design of the system and its software.*

*Examples of such goals might be: an emphasis on speed versus memory use; or working, looking, or “feeling” like an existing product.*

*Guidelines include coding guidelines and conventions.*

*For each such goal or guideline, describe the reason for its desirability unless it is implicitly obvious.*

*Describe any design policies and/or tactics that do not have sweeping architectural implications (meaning they would not significantly affect the overall organization of the system and its high-level structures), but which nonetheless affect the details of the interface and/or implementation of various aspects of the system (e.g., choice of which specific product to use)*>

## Architectural Strategies

*<Describe any design decisions and/or strategies that affect the overall organization of the system and its higher-level structures. These strategies should provide insight into the key abstractions and mechanisms used in the system architecture. Describe the reasoning employed for each decision and/or strategy (possibly referring to previously stated design goals and principles) and how any design goals or priorities were balanced or traded-off.*

*Examples of design decisions might concern (but are not limited to) things like the following:*

*• Use of a particular type of product (programming language, database, library, commercial off-the-shelf (COTS) product, etc.)*

*• Reuse of existing software components to implement various parts/features of the system*

*• Future plans for extending or enhancing the software*

*• User interface paradigms (or system input and output models)*

*• Hardware and/or software interface paradigms*

*• Error detection and recovery*

*• Memory management policies*

*• External databases and/or data storage management and persistence*

*• Distributed data or control over a network*

*• Generalized approaches to control*

*• Concurrency and synchronization*

*• Communication mechanisms*

*• Management of other resources*

>

## Design and Program Evaluation

*<Evaluate your design and describe which levels of coupling and cohesion that your design is at. Give proofs for your assumptions. Explain if there is any special design or exceptions>*

*<You may show the previous design from which you made improvements to get better levels of coupling and cohesion. You should clarify how and why you did these improvements>*

*<Does your design follow the SOLID principles if there are new requirements/changing requirements in the future? Give proofs for your assumptions. Explain if there is any special design or exceptions>*

*<You may show the previous design from which you made improvements to get a better design, which follows SOLID principles in spite of additional requirements. You should clarify how and why you did these improvements>*

### Cohesion & SRP

Explain in detail for each class in the table.

Table 4. Cohesion & SRP of AIMS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *#* | *Class Name* | *PIC* | *Cohesion* | *SRP* | *Solution* |
| 1 | Order | TuấnNH | Temporal: m1(), m2() | Yes:  2 responsibilities: r1, r2 | Separate m2() to the class XXX… |
| 2 | <class name> | <person in charge> | <which level, why, where> | <Yes/No, why, where> | <solution if necessary> |

### Coupling & Other SOLID

Explain in detail for each problem in the table. Each problem, you may provide a class diagram before and after the improvement.

Table 5. Coupling & other SOLID of AIMS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Problem* | *PIC* | *Location* | *Solution* |
| 1 | Control coupling | TuấnNH | Class A and class B | Create a new class C as a superclass for A and B… |
| 2 | <Any bad coupling level or any SOLID violation> | <person in charge> | <list of class or you may provide a class diagram if necessary> |  |

## Design Patterns

*<Do you use any design patterns for your design? If yes, describe detailly why you use those design patterns? Describe in detail on the solutions and how to implement each design pattern>*