HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and Communications Technology

Software Requirement Specification

Version 1.1

AIMS.

Môn: Thiết kế và xây dựng phần mềm

Giảng viên: Nguyễn Thị Thu Trang

Nông Hoàng Lâm – 20204575

Nguyễn Ngọc Linh – 20204576

Vũ Văn Lợi - 20204577

*Hanoi,* *<10, 2023r>*

Table of contents

Table of contents 1

1 Introduction 2

1.1 Objective 2

1.2 Scope 2

1.3 Glossary 2

1.4 References 2

2 Overall requirements 3

2.1 Actors 3

2.2 General use case diagram 3

2.3 Business processes 4

3 Detail requirements 5

3.1 Specification of Use case UC001 – Đặt hàng 5

3.2 Specification of Use case UC002 – Thanh toán” 8

3.3 Specification of Use case UC003 – Đặt hàng nhanh” 9

4 Supplementary specification 13

4.1 Functionality 13

4.2 Usability 13

4.3 Reliability 13

4.4 Performance 13

4.5 Maintainability 13

4.6 Design Constraints 13

# Introduction

## Objective

Tài liệu này trình bày miêu tả về chức năng của phần mềm thương mại điện tử AIMS.Tài liệu này dành cho các bên liên quan và phía lập trình viên.

## Scope

Bao gồm các chức năng chính của phần mềm

## Glossary

## References

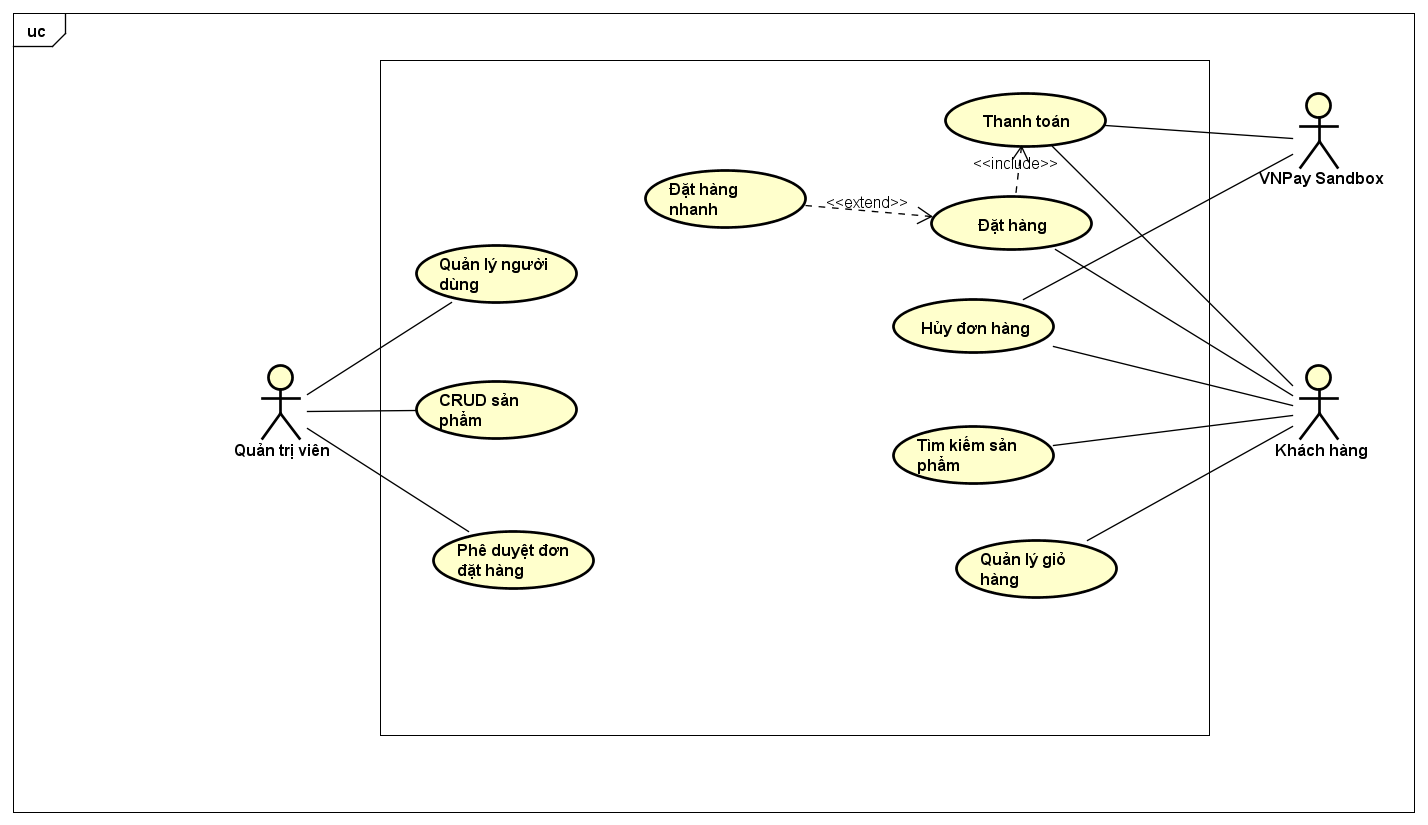
# Overall requirements

## Actors

Khách hàng

VNPay sandbox

## General use case diagram

Phần mềm AIMS giúp khách hàng có thể đặt sản phẩm và thanh toán. 

## Business processes

A diagram of a flowchart

Description automatically generated

# Detail requirements

## Specification of Use case UC001 – Đặt hàng

1. **Use case code**

UC001

1. **Brief Description**

Usecase này giúp khách hàng có thể đặt sản phẩm và thanh toán nội vùng.

1. **Actors**

Khách hàng.

1. **Preconditions.**

Không có.

1. **Basic Flow of Events**
   1. Khách hàng bấm vào xem giỏ hàng.
   2. Hệ thống tính phí tổng giá sản phẩm.
   3. Hệ thống kiểm tra các sản phẩm trong giỏ còn có tồn kho.
   4. Hệ thống sẽ hiện ra danh sách các món hàng khách hàng muốn đặt (tên sản phẩm, số lượng, và giá cả).
   5. Khách hàng bấm vào nút “Đặt hàng”.
   6. Hệ thống kiểm tra các sản phẩm trong giỏ còn có tồn kho.
   7. Hệ thống hiển thị phiếu điền thông tin giao hàng, yêu cầu khách hàng cập nhật thông tin giao hàng.
   8. Khách hàng điền các thông tin cần thiết và không chọn “giao hàng nhanh”. Bấm “Cập nhật”.
   9. Hệ thống kiểm tra thông tin đầu vào.
   10. Hệ thống tính phí giao hàng.
   11. Hệ thống sẽ hiển thị thông tin đơn hàng tạm thời.
   12. Gọi tiếp đến Usecase “Thanh toán”.
   13. Hệ thống làm trống giỏ hàng.

Alternative flows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
|  | Tại bước 3 | Nếu hệ thống kiểm tra trong kho không đủ các sản phẩm | * Hệ thống sẽ thông báo tới khách hàng phải sản phẩm không đủ trong kho và yêu cầu khách hàng cập nhật lại giỏ hàng các sản phẩm thiếu | Tiếp tục ở bước 4. |
|  | Tại bước 8 | nếu khách hàng bỏ trống các trường thông tin bắt buộc hoặc ghi sai định dạng | * hệ thống sẽ yêu cầu khách hàng khách hàng nhập đầy đủ thông tin | Tiếp tục bước 8 |
|  | Tại bước 8 | nếu khách hàng chọn “Giao hàng nhanh” | * Chèn vào usecase “Đặt hàng nhanh” | Tiếp tục usecase “Đặt hàng nhanh” |
|  | Tại bước 1 | nếu trong giỏ hàng không có sản phẩm | * hệ thống sẽ thông báo trong giỏ hàng không có sản phẩm | Tiếp tục bước 1 |

1. **Input data**

Table A-Input data of “phiếu thông tin giao hàng”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
|  | Địa chỉ giao hàng |  | Có | 50 từ tối đa | Số 1, Đường Tạ Quang Bửu, quận Hai Bà Trưng. |
|  | Tên người nhận |  | Có | 30 ký tự tối đa | Nông Hoàng Lâm |
|  | Số điện thoại |  | Có | Từ 9-11 chữ số. | 09123456789 |
|  | Tỉnh | Chọn từ danh sách 64 tỉnh | Có | Chọn 1 tỉnh | Hà Nội |
|  | Số thẻ |  | Có | 16 ký tự | 1234 5678 1234 5678 |
|  | Tên chủ thẻ |  | Có | 30 ký tự tối đa, viết hoa, không dấu | NONG HOANG LAM |

1. **Output data**

Table B-Output data of “Bảng thông tin đơn đặt hàng tạm thời”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
|  | Tiêu đề | Tên của sản phẩm |  | Đĩa nhạc Pop |
|  | Giá | Giá trên một đơn vị sản phẩm | Số nguyên dương >0  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng | 100,000 đồng |
|  | Số lượng | Số lượng của sản phẩm | Số nguyên dương > 0 | 3 |
|  | Tổng số tiền của sản phẩm | Tổng số tiền của loại sản phẩm | Số nguyên dương >0  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng | 300,000 đồng |
|  | Tổng tiền phải trả khi chưa tính VAT | Tổng số tiền của tất cả sản phẩm trong giỏ trước khi tính VAT | 500,000 đồng |
|  | Tổng tiền phải trả khi đã tính VAT | Tổng số tiền của tất cả sản phẩm trong giỏ sau khi tính VAT | Số nguyên dương >0  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng  Hiện %VAT đã tính và thành tiền của VAT | 550,000 đồng (10%VAT = 50,000 đồng) |

## Specification of Use case UC002 – Thanh toán”

1. **Use case code**

UC002

1. **Brief Description**

Khi khách hàng muốn thanh toán đơn hàng của mình với AIMS thông qua cổng thanh toán ngân hàng.

1. **Actors**

Khách hàng, ngân hàng.

1. **Preconditions**
2. **Basic Flow of Events**
   1. Khách hàng chọn phương thức thanh toán sử dụng cổng thanh toán VNPay.
   2. Hệ thống hiển thị phiếu thông tin thanh toán.
   3. Khách hàng điền thông tin giao dịch và nội dung giao dịch và bấm xác nhận.
   4. Hệ thống sẽ hiển thị thông tin giao dịch.
   5. Hệ thống gửi thông tin giao dịch đến hòm thư điện tử của khách hàng.
   6. Hệ thống hiển thị thông báo “Giao dịch thành công”.
3. **Alternative flows**

Table N-Alternative flows of events for UC Place order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
|  | Tại bước 3 | nếu khách hàng đưa thông tin số thẻ sai | * hệ thống sẽ hiển thị thông báo “Không tìm thấy số tài khoản, vui lòng nhập lại thông tin”. | Tiếp tục bước 3. |
|  | Tại bước 3 | nếu tài khoản của khách hàng không đủ số dư | * hệ thống sẽ hiển thị thông báo “Số dư không đủ, vui lòng thanh toán lại.” | Tiếp tục bước 3. |

1. **Input data**

Table A-Input data of form điền thông tin bên ngân hàng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
|  | Số thẻ |  | Có | 16 ký tự | 1234 5678 1234 5678 |
|  | Tên chủ thẻ |  | Có | 20 ký tự tối đa | NONG HOANG LAM |
|  | Tên ngân hàng |  | Có | Chọn từ danh sách các ngân hàng | Ngân hàng BIDV |

1. **Output data**

Table B-Output data of “Bảng thông tin giao dịch”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
|  | mã giao dịch |  | Được format theo template sau: “AIMS” + lần giao dịch | AIMS191103 |
|  | tên chủ thẻ |  | Viết hoa, không dấu | NONG HOANG LAM |
|  | số tiền bị trừ |  | Số nguyên âm, có đơn vị tiền tệ vnđ đằng sau | -108,000 đồng |
|  | nội dung giao dịch |  | Được ghi cùng với mã giao dịch | Giao dịch thanh toán từ AIMS, mã giao dịch: AIMS19103 |
|  | số dư |  | Số nguyên dương >0  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng | 999,000 đồng |
|  | ngày giờ giao dịch |  | Được định dạng theo “DD/MM/YYYY:h:m:s” | 19/11/2023:19:23:56 |

## Specification of Use case UC003 – Đặt hàng nhanh”

1. **Use case code**

UC003

1. **Brief Description**

Khi khách hàng muốn đặt đơn hàng của mình nhanh.

1. **Actors**

Khách hàng

1. **Preconditions**

Usecase mở rộng của usecase đặt hàng, khi người dùng ấn “Đặt hàng nhanh”

1. **Basic Flow of Events**
   1. Hệ thống thêm các trường thông tin đặt hàng nhanh vào phiếu thông tin giao hàng.
   2. Người dùng nhập các trường thông tin đặt hàng nhanh.
   3. Hệ thống kiểm tra địa điểm giao hàng nhanh có hỗ trợ.
   4. Hệ thống kiểm tra các sản phẩm có hỗ trợ giao hàng nhanh.
   5. Hệ thống cập nhật phí giao hàng nhanh.
   6. Hệ thống hiển thị phiếu thông tin “đặt hàng nhanh”, bao gồm chi phí đặt hàng nhanh, các sản phẩm có thể giao hàng nhanh và các sản phẩm không hỗ trợ .
   7. Người dùng bấm nút xác nhận.
   8. Hệ thống cập nhật chi phí giao hàng nhanh, các sản phẩm giao hàng nhanh, các sản phẩm giao hàng thường tại phiếu thông tin “giao hàng”.
2. **Alternative flows**

Table 2-Alternative flows of events for UC Place order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
|  | Tại bước 4 | nếu các sản phẩm đều không hỗ trợ giao hàng nhanh. | * hệ thống sẽ hiển thị thông báo yêu cầu người dùng cập nhật lại đơn hàng. | Tiếp tục tại bước 4 |
|  | Tại bước 3 | nếu địa chỉ không hỗ trợ giao hàng nhanh | * hệ thống sẽ hiển thị thông báo yêu cầu người dùng phải nhập lại thông tin. | Tiếp tục tại bước 3 |
|  | Tại bước 2 | Nếu có trường thông tin bị để trống hoặc ghi sai định dạng | * Hệ thống sẽ hiển thi thông báo yêu cầu người dùng sửa lại thông tin | Tiếp tục tại bước 3. |

1. **Input data**

Table A-Input data of “phiếu thông tin giao hàng nhanh”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
|  | Địa chỉ giao hàng |  | Có | 50 từ tối đa | Số 1, Đường Tạ Quang Bửu, quận Hai Bà Trưng. |
|  | Địa chỉ giao hàng nhanh |  | Có | 50 từ tối đa | Số 1, Đường Tạ Quang Bửu, quận Hai Bà Trưng |
|  | Chỉ dẫn giao hàng |  | Không | 50 từ tối đa | Gửi đến văn phòng vào 11 giờ trưa |
|  | Thời gian nhận hàng |  | Có | Định dạng theo hh:mm | 11:30 |
|  | Số điện thoại |  | Có | Từ 9-11 chữ số. | 09123456789 |
|  | Tỉnh | Chọn từ danh sách 64 tỉnh | Có | Chọn 1 tỉnh | Thành phố Hà Nội |

1. **Output data**

Table B-Output data of “Bảng thông tin đơn đặt hàng nhanh”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
|  | Tiêu đề | Tên của sản phẩm |  | Đĩa nhạc Pop |
|  | Giá | Giá trên một đơn vị sản phẩm | Số nguyên dương >0  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng | 100,000 đồng |
|  | Loại đặt hàng | Sản phẩm là đặt hàng nhanh hay đặt hàng thường | Sẽ có biểu tượng đặt hàng nhanh bên cạnh ảnh sản phẩm. |  |
|  | Số lượng | Số lượng của sản phẩm | Số nguyên dương > 0 | 3 |
|  | Đặt hàng nhanh | Có phải sản phẩmđặt hàng nhanh | Một ô tích, không cho phép người dùng thay đổi | V |
|  | Giá thành đặt hàng nhanh |  | Số nguyên dương >0  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng | 50,000 dồng |
|  | Tổng số tiền của sản phẩm | Tổng số tiền của loại sản phẩm | Số nguyên dương >0  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng | 300,000 đồng |
|  | Tổng tiền phải trả khi chưa tính VAT | Tổng số tiền của tất cả sản phẩm trong giỏ trước khi tính VAT | 500,000 đồng |
|  | Tổng tiền phải trả khi đã tính VAT | Tổng số tiền của tất cả sản phẩm trong giỏ sau khi tính VAT | Số nguyên dương >0  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng  Hiện %VAT đã tính và thành tiền của VAT | 550,000 đồng (10%VAT = 50,000 đồng) |

## Specification of Use case UC004 – Hủy đơn hàng”

1. **Use case code**

UC004

1. **Brief Description**

Khi khách hàng muốn hủy đơn hang mà mình đã thanh toán và đơn hang chưa được kiểm duyệt.

1. **Actors**

Khách hàng

1. **Preconditions**

Đang chạy phần mềm và đang ở tại Screen Home và người dung đang có kết nối với mạng internet.

1. **Basic Flow of Events**
   1. Người dung bấm vào tìm kiếm đơn hàng.
   2. Hệ thống hiển thị trang tìm kiếm đơn hàng.
   3. Người dung nhập số điện thoại của đơn hang đã đặt để tìm kiếm đơn hàng.
   4. Hệ thống hiển thị tất cả các đơn hàng có số điện thoại mà người dùng yêu cầu.
   5. Người dung bấm nút hủy đơn hàng của đơn hàng mình muốn hủy.
   6. Hệ thống xử lý hoàn tiền.
   7. Hệ thống hiển thị thông báo hủy đơn hàng thành công.”.
2. **Alternative flows**

Table 2-Alternative flows of events for UC Place order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
|  | Tại bước 2 | Nếu không tìm thấy bất cứ sản phẩm nào | * hệ thống sẽ hiển thị thông báo không tồn tại sản phẩm. | Tiếp tục tại bước 2 |
|  | Tại bước 2 | nếu số điện thoại không phù hợp. | * hệ thống sẽ hiển thị thông báo số điện thoại không hợp lệ. | Tiếp tục tại bước 2 |

1. **Input data**

Table A-Input data of “phiếu thông tin giao hàng nhanh”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
|  | Số điện thoại | Số điện thoại để tìm kiếm đơn hàng | Có | Từ 9-11 chữ số. | 09123456789 |

1. **Output data**

Table B-Output data of “Bảng thông tin hủy đơn hàng”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
|  | Mã đơn hàng | Mã của đơn hàng đã đặt |  | 1 |
|  | Tổng số tiền của đơn hàng |  | Số nguyên dương >0  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng | 100,000 đồng |
|  | Địa chỉ |  | Chữ | 3 Tạ Quang Bửu |
|  | Tình thành |  | Chữ | Hà Nội |
|  | Email |  | Định dạng email | nhl@gmail.com |
|  | Số điện thoại |  | 9-11 chữ số  Có dấu , tách giữa phần nghìn  Đơn vị nghìn đồng | 082123456 |
|  | Mã giao dịch |  | Chữ số | 47112345 |
|  | Trạng thái đơn hàng |  | Là một trong các trạng thái sau đây:   * Pending: Đang chờ thanh toán * Canceled: Đã hủy đơn hàng do chưa thanh toán. * Refunded: Đã hoàn tiền đơn hàng. * Success: Đơn hàng đã thanh toán và được duyệt thành công. | refunded |

A diagram of a company

Description automatically generated with medium confidence

*Biểu đồ hoạt động “Hủy đơn hàng”*

## Specification of Use case UC005 – “Tìm kiếm sản phẩm”

A screenshot of a diagram

Description automatically generated

1. **Mã use case**

UC003

1. **Mô tả ngắn gọn**

Use case này mô tả tương tác giữa phầm mềm AIMS với khách hàng khi khách hàng muốn tìm kiếm sản phẩm trong hệ thống

1. **Tác nhân**
2. Khách hàng
3. **Tiền điều kiện**

Không có

1. **Luồng sự kiện chính**
2. Phần mềm AIMS hiển thị màn trang chủ với danh sách tất cả các sản phẩm đang được bán.
3. Người dùng chọn hình thức tìm kiếm
4. Người dùng điền thông tin sản phẩm rồi ấn nút search hoặc nút enter.
5. Phần mềm AIMS sẽ tìm kiếm trong DataBase những sản phẩm thỏa mãn từ khóa tìm kiếm. Nếu không có sản phẩm thỏa mãn sẽ hiển thị không tìm thấy sản phẩm.
6. **Luồng sự kiện thay thế**

Không có

1. **Dữ liệu vào**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví Dụ** |
| 1. | Thông tin tìm kiếm | Thông tin vật phẩm cần tìm | Có | Tất cả | cÁ heo  DVD  Harry porter |
|  |  |  |  |  |  |

1. **Dữ liệu ra**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví Dụ** |
| 1. | Tên vật phẩm | Tên vật phẩm tìm thấy | Có | Tất cả | cá heo  DVD  Harry porter |
| **2** | Giá | Giá của vật phẩm | Có | |  | | --- | | Gồm các chữ số từ 0 đến 9, mỗi 3 chữ số từ cuối sẽ ngăn cách bởi dấu phẩy | | 400,000 |

1. **Hậu điều kiện**

Không có

A diagram with text and words

Description automatically generated with medium confidence

*Biểu đồ hoạt động UseCase “Tìm kiếm sản phẩm”*

## Specification of Use case UC004 – “Quản lý giỏ hàng”

1. **Use case code**

UC003

1. **Brief Description**

UseCase mô tả quá trình khách hàng thực hiện những thao tác để quản lý giỏ hàng của mình.

1. **Actors**

Khách hàng

1. **Preconditions**

Không

1. **Basic Flow of Events (Success – Chọn sản phẩm mua)**
2. Khách hàng truy cập vào hệ thống.
3. Hệ thống hiển thị danh sách và thông tin sản phẩm.
4. Khách hàng chọn sản phẩm muốn mua, chọn số lượng sản phẩm cần mua.
5. Khách hàng click vào nút “Thêm vào giỏ hàng”.
6. Hệ thống hiển thị thông báo đã thêm sản phẩm thành công.
7. Khách hàng click vào giỏ hàng.
8. Hệ thống hiển thị danh sách sản phẩm có trong giỏ hàng.
9. Khách hàng lựa chọn sản phẩm muốn thanh toán.
10. Hệ thống hiển thị số tiền tổng sản phẩm đang chọn.
11. **Alternative flows**

Table N-Alternative flows of events for UC Place order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
|  | 4 | Số lượng thêm vào giỏ hàng vượt quá số lượng trong kho | * Hệ thống báo lỗi: Số lượng mặt hàng trong kho không đủ | Tiếp tục lại bước 3 |
|  | 8 | Số lượng mua đang nhiều hơn số lượng trong kho | * Hệ thống báo lỗi: Số lượng hàng hiện đang không đủ | Tiếp tục lại bước 8 |
|  | 8 | Khách hàng chọn Loại bỏ sản phẩm ra khỏi giỏ hàng. | * Hệ thống loại sản phẩm ra khỏi giỏ hàng, nếu sản phẩm đang được chọn sẽ trừ số tiền của sản phẩm đó | Tiếp tục lại bước 8 |

1. **Input data**

Table A-Input data of …

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
|  |  |  |  |  |  |

1. **Output data**

Table B-Output data of …

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
|  |  |  |  |  |

1. **Postconditions**

Không

A diagram of a diagram

Description automatically generated

*Biểu đồ hoạt động UseCase “Quản lý giỏ hàng”*

# Supplementary specification

## Functionality

Đặt hàng, đặt hàng nhanh, hủy đơn hàng, thanh toán, quản lý người dùng, phê duyệt đơn hàng, cập nhật giỏ hàng, tìm kiếm sản phẩm, CRUD sản phẩm.

## Usability

Hệ thống hoạt động 24/7

## Reliability

Hệ thống có thể hoạt động trở lại bình thường trong vòng 1 giờ sau khi xảy ra lỗi.

## Performance

Có thể cho phép phục vụ 1000 khách hàng cùng lúc mà hiệu suất không bị giảm đáng kể, đồng thời có thể hoạt động 300 giờ liên tục không hỏng hóc, Thời gian đáp ứng của hệ thống tối đa là 2 giây khi bình thường hoặc 3 giây lúc cao điểm.

## Maintainability

Hệ thống có thể hoạt động trở lại bình thường trong vòng 1 giờ sau khi xảy ra lỗi

## Design Constraints

Không có

# INTERACTION DIAGRAMS

## Sequence diagram UseCase “Hủy đơn hàng”

A diagram of a company

Description automatically generated with medium confidence

## Sequence diagram UseCase “Tìm kiếm sản phẩm”

A computer screen shot of a diagram

Description automatically generated

## Sequence diagram UseCase “Quản lý giỏ hàng”

A screenshot of a computer screen

Description automatically generated

## Communication diagram UseCase “Quản lý giỏ hàng”

A diagram of a diagram

Description automatically generated

## Communication diagram UseCase “Tìm kiếm sản phẩm”

A diagram of a diagram

Description automatically generated

# ANALYSIS CLASS DIAGRAM

## ANALYSIS CLASS DIAGRAM Tổng quan

A diagram of a computer program

Description automatically generated

## ANALYSIS CLASS DIAGRAM UseCase “Hủy đơn hàng”

A screenshot of a computer program

Description automatically generated

## ANALYSIS CLASS DIAGRAM “Tìm kiếm sản phẩm”

A screenshot of a computer

Description automatically generated

## ANALYSIS CLASS DIAGRAM UseCase “Quản lý giỏ hàng”

A diagram of a diagram

Description automatically generated with medium confidence

# INTERFACE DESIGN

**A screenshot of a computer

Description automatically generated**

**Home Screen Specification**

| **Screen Element** | **Control** | **Operation** | **Function** |
| --- | --- | --- | --- |
| Product List | Interactive List | Scrolling/Clicking | Display products available for purchase |
| Quantity Selector | Input Field/Button | Writing/Clicking | Select the quantity of the product |
| Add to Cart Button | Button | Clicking | Add the selected quantity of product to the cart |

**A screenshot of a computer

Description automatically generated**

**Order Search Screen Specification**

| **Screen Element** | **Control** | **Operation** | **Function** |
| --- | --- | --- | --- |
| Phone Number Field | Text Field | Writing | Enter a phone number to search for orders |
| Search Button | Button | Clicking | Initiate search for orders associated with the phone number |

**A screenshot of a computer

Description automatically generated**

**Cart Screen Specification**

| **Screen Element** | **Control** | **Operation** | **Function** |
| --- | --- | --- | --- |
| Item List | Interactive List | Display | Show items added to the cart |
| Quantity Selector | Input Field/Button | Writing/Clicking | Adjust the quantity of a cart item |
| Remove Item Button | Button | Clicking | Remove an item from the cart |
| Total Price | Label | Display | Show the total price of items in the cart |
| Checkout Button | Button | Clicking | Proceed to the checkout process |

**A screenshot of a computer

Description automatically generated**

**Delivery Screen Specification**

| **Screen Element** | **Control** | **Operation** | **Function** |
| --- | --- | --- | --- |
| Recipient Name Field | Text Field | Writing | Enter the name of the recipient for delivery |
| Address Field | Text Field | Writing | Enter the delivery address |
| Delivery Options | Checkbox/Dropdown | Selecting/Toggling | Choose delivery options and services |

**A screenshot of a computer

Description automatically generated**

**Invoice Screen Specification**

| **Screen Element** | **Control** | **Operation** | **Function** |
| --- | --- | --- | --- |
| Itemized List | Interactive List | Display | Show a list of purchased items with prices |
| Total Amount | Label | Display | Display the total amount due |
| Confirm Payment Button | Button | Clicking | Confirm the purchase and proceed to payment |

**A screen shot of a computer screen

Description automatically generated**

**Payment Screen Specification**

| **Screen Element** | **Control** | **Operation** | **Function** |
| --- | --- | --- | --- |
| Total Price | Label | Display | Show the total price to be paid |
| Payment Button | Button | Clicking | Initiate the payment process |

**A screenshot of a computer

Description automatically generated  
Payment Result Screen Specification**

| **Screen Element** | **Control** | **Operation** | **Function** |
| --- | --- | --- | --- |
| Failure Message | Label | Display | Show that the payment process was canceled |
| Return Home Button | Button | Clicking | Navigate back to the main screen |

# Data Modeling

## CONCEPTUAL DATA MODEL

A diagram of a computer

Description automatically generated

## DATABASE DESIGN

### Logical Data Model

A computer screen shot of a computer

Description automatically generated

### Physical Data Model

-- phpMyAdmin SQL Dump

-- version 5.2.1

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1

-- Generation Time: Dec 31, 2023 at 06:30 PM

-- Server version: 10.4.32-MariaDB

-- PHP Version: 8.0.30

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `aims`

--

-- --------------------------------------------------------

--

-- Table structure for table `book`

--

CREATE TABLE `book` (

`book\_id` int(11) NOT NULL,

`media\_id` int(11) DEFAULT NULL,

`author` varchar(45) DEFAULT NULL,

`cover\_type` varchar(45) DEFAULT NULL,

`publisher` varchar(45) DEFAULT NULL,

`publish\_date` varchar(45) DEFAULT NULL,

`num\_of\_pages` int(11) DEFAULT NULL,

`language` varchar(45) DEFAULT NULL,

`book\_category` varchar(45) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `book`

--

INSERT INTO `book` (`book\_id`, `media\_id`, `author`, `cover\_type`, `publisher`, `publish\_date`, `num\_of\_pages`, `language`, `book\_category`) VALUES

(1, 1, 'John Doe', 'Hardcover', 'Example Publisher', '2022-01-01', 300, 'English', 'Fiction'),

(2, 2, 'Jane Smith', 'Paperback', 'Another Publisher', '2021-12-15', 250, 'Spanish', 'Non-Fiction'),

(3, 3, 'Bob Johnson', 'Hardcover', 'Great Books Publishing', '2022-03-10', 400, 'French', 'Mystery');

-- --------------------------------------------------------

--

-- Table structure for table `cd`

--

CREATE TABLE `cd` (

`cd\_id` int(11) NOT NULL,

`media\_id` int(11) DEFAULT NULL,

`artist` varchar(45) DEFAULT NULL,

`released\_date` varchar(45) DEFAULT NULL,

`record\_lable` varchar(45) DEFAULT NULL,

`music\_type` varchar(45) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `delivery\_info`

--

CREATE TABLE `delivery\_info` (

`delivery\_info\_id` int(11) NOT NULL,

`name` varchar(50) DEFAULT NULL,

`phone\_number` varchar(45) DEFAULT NULL,

`province` varchar(45) DEFAULT NULL,

`instruction` varchar(45) DEFAULT NULL,

`address` varchar(45) DEFAULT NULL,

`email` varchar(45) DEFAULT NULL,

`is\_rush` smallint(6) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `delivery\_info`

--

-- --------------------------------------------------------

--

-- Table structure for table `dvd`

--

CREATE TABLE `dvd` (

`dvd\_id` int(11) NOT NULL,

`media\_id` int(11) DEFAULT NULL,

`disc\_type` varchar(45) DEFAULT NULL,

`director` varchar(45) DEFAULT NULL,

`studio` varchar(45) DEFAULT NULL,

`released\_date` varchar(45) DEFAULT NULL,

`subtitle` varchar(45) DEFAULT NULL,

`runtime` varchar(45) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `invoice`

--

CREATE TABLE `invoice` (

`invoice\_id` int(11) NOT NULL,

`transaction\_id` int(11) DEFAULT NULL,

`order\_id` int(11) DEFAULT NULL,

`amount` float DEFAULT NULL,

`status` char(45) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `media`

--

CREATE TABLE `media` (

`media\_id` int(11) NOT NULL,

`price` float DEFAULT NULL,

`available` int(11) DEFAULT NULL,

`name` varchar(45) DEFAULT NULL,

`imageURL` char(45) DEFAULT NULL,

`category` varchar(45) DEFAULT NULL,

`weight` float DEFAULT NULL,

`support\_rush\_delivery` smallint(6) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `media`

--

INSERT INTO `media` (`media\_id`, `price`, `available`, `name`, `imageURL`, `category`, `weight`, `support\_rush\_delivery`) VALUES

(1, 19000, 50, 'Book Title 1', 'book\_image\_url\_1.jpg', 'Book', 1.5, 1),

(2, 15000, 30, 'Book Title 2', 'book\_image\_url\_2.jpg', 'Book', 1.2, 0),

(3, 23000, 25, 'Book Title 3', 'book\_image\_url\_3.jpg', 'Book', 1.8, 1);

-- --------------------------------------------------------

--

-- Table structure for table `order`

--

CREATE TABLE `order` (

`order\_id` int(11) NOT NULL,

`delivery\_info\_id` int(11) DEFAULT NULL,

`shipping\_amounts` float DEFAULT NULL,

`total\_amounts` float DEFAULT NULL,

`cart\_amounts` float DEFAULT NULL,

`rush\_order` smallint(6) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Dumping data for table `order`

--

-- --------------------------------------------------------

--

-- Table structure for table `order\_media`

--

CREATE TABLE `order\_media` (

`media\_id` int(11) NOT NULL,

`order\_id` int(11) NOT NULL,

`delivery\_info\_id` int(11) NOT NULL,

`quantity` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

CREATE TABLE `rush\_delivery\_info` (

`delivery\_info\_id` int(11) NOT NULL,

`rd\_time` varchar(45) DEFAULT NULL,

`rd\_address` varchar(45) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

-- --------------------------------------------------------

--

-- Table structure for table `transaction`

--

CREATE TABLE `transaction` (

`transaction\_id` int(11) NOT NULL,

`amount` float DEFAULT NULL,

`contents` varchar(45) DEFAULT NULL,

`error\_message` varchar(45) DEFAULT NULL,

`time` timestamp(6) NOT NULL DEFAULT current\_timestamp(6) ON UPDATE current\_timestamp(6),

`ref\_id` int(10) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

--

-- Indexes for dumped tables

--

--

-- Indexes for table `book`

--

ALTER TABLE `book`

ADD PRIMARY KEY (`book\_id`),

ADD KEY `media\_id` (`media\_id`);

--

-- Indexes for table `cd`

--

ALTER TABLE `cd`

ADD PRIMARY KEY (`cd\_id`),

ADD KEY `media\_id` (`media\_id`);

--

-- Indexes for table `delivery\_info`

--

ALTER TABLE `delivery\_info`

ADD PRIMARY KEY (`delivery\_info\_id`);

--

-- Indexes for table `dvd`

--

ALTER TABLE `dvd`

ADD PRIMARY KEY (`dvd\_id`),

ADD KEY `media\_id` (`media\_id`);

--

-- Indexes for table `invoice`

--

ALTER TABLE `invoice`

ADD PRIMARY KEY (`invoice\_id`),

ADD UNIQUE KEY `transaction\_id` (`transaction\_id`),

ADD KEY `order\_id` (`order\_id`);

--

-- Indexes for table `media`

--

ALTER TABLE `media`

ADD PRIMARY KEY (`media\_id`);

--

-- Indexes for table `order`

--

ALTER TABLE `order`

ADD PRIMARY KEY (`order\_id`),

ADD KEY `delivery\_info\_id` (`delivery\_info\_id`);

--

-- Indexes for table `order\_media`

--

ALTER TABLE `order\_media`

ADD PRIMARY KEY (`media\_id`,`order\_id`,`delivery\_info\_id`),

ADD KEY `order\_id` (`order\_id`);

--

-- Indexes for table `rush\_delivery\_info`

--

ALTER TABLE `rush\_delivery\_info`

ADD PRIMARY KEY (`delivery\_info\_id`);

--

-- Indexes for table `transaction`

--

ALTER TABLE `transaction`

ADD PRIMARY KEY (`transaction\_id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `book`

--

ALTER TABLE `book`

MODIFY `book\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

--

-- AUTO\_INCREMENT for table `cd`

--

ALTER TABLE `cd`

MODIFY `cd\_id` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT for table `delivery\_info`

--

ALTER TABLE `delivery\_info`

MODIFY `delivery\_info\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=53;

--

-- AUTO\_INCREMENT for table `dvd`

--

ALTER TABLE `dvd`

MODIFY `dvd\_id` int(11) NOT NULL AUTO\_INCREMENT;

--

-- AUTO\_INCREMENT for table `invoice`

--

ALTER TABLE `invoice`

MODIFY `invoice\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=14;

--

-- AUTO\_INCREMENT for table `media`

--

ALTER TABLE `media`

MODIFY `media\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;

--

-- AUTO\_INCREMENT for table `order`

--

ALTER TABLE `order`

MODIFY `order\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=51;

--

-- AUTO\_INCREMENT for table `transaction`

--

ALTER TABLE `transaction`

MODIFY `transaction\_id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=14266178;

--

-- Constraints for dumped tables

--

--

-- Constraints for table `book`

--

ALTER TABLE `book`

ADD CONSTRAINT `book\_ibfk\_1` FOREIGN KEY (`media\_id`) REFERENCES `media` (`media\_id`);

--

-- Constraints for table `cd`

--

ALTER TABLE `cd`

ADD CONSTRAINT `cd\_ibfk\_1` FOREIGN KEY (`media\_id`) REFERENCES `media` (`media\_id`);

--

-- Constraints for table `dvd`

--

ALTER TABLE `dvd`

ADD CONSTRAINT `dvd\_ibfk\_1` FOREIGN KEY (`media\_id`) REFERENCES `media` (`media\_id`);

--

-- Constraints for table `invoice`

--

ALTER TABLE `invoice`

ADD CONSTRAINT `invoice\_ibfk\_1` FOREIGN KEY (`transaction\_id`) REFERENCES `transaction` (`transaction\_id`) ON UPDATE CASCADE,

ADD CONSTRAINT `invoice\_ibfk\_2` FOREIGN KEY (`order\_id`) REFERENCES `order` (`order\_id`);

--

-- Constraints for table `order`

--

ALTER TABLE `order`

ADD CONSTRAINT `order\_ibfk\_1` FOREIGN KEY (`delivery\_info\_id`) REFERENCES `delivery\_info` (`delivery\_info\_id`);

--

-- Constraints for table `order\_media`

--

ALTER TABLE `order\_media`

ADD CONSTRAINT `order\_media\_ibfk\_1` FOREIGN KEY (`media\_id`) REFERENCES `media` (`media\_id`),

ADD CONSTRAINT `order\_media\_ibfk\_2` FOREIGN KEY (`order\_id`) REFERENCES `order` (`order\_id`);

--

-- Constraints for table `rush\_delivery\_info`

--

ALTER TABLE `rush\_delivery\_info`

ADD CONSTRAINT `rush\_delivery\_info\_ibfk\_1` FOREIGN KEY (`delivery\_info\_id`) REFERENCES `delivery\_info` (`delivery\_info\_id`);

COMMIT;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;