

SOFTWARE DOCUMENT: MICROSHOP E-COMMERCE PLATFORM

Chapter 3: Requirement Specifications

QUICK REFERENCE

| Category | Details |
|-----------------------------|-------------------------------------|
| Total Stakeholders | 5 stakeholder groups |
| Total Use Cases | 20 use cases (12 customer, 8 admin) |
| Functional Requirements | 60 requirements across 5 modules |
| Non-Functional Requirements | 20 requirements across 5 categories |
| Document Version | 1.0 |
| Document Status | Complete - Ready for review |

3. REQUIREMENT SPECIFICATIONS

3.1. Stakeholders for the System

This section identifies all stakeholders who have interest in or will be affected by the Microshop E-Commerce Platform.

3.1.1. Primary Stakeholders

1. End Customers (Vietnamese Consumers)

Profile:

- Age range: 18-45 years old
- Primary need: Purchase mobile phones and laptops online
- Tech-savvy individuals seeking convenient shopping experience
- Located in Vietnam, primarily urban areas

Interests and Needs:

- Easy-to-use interface for browsing and purchasing products
- Detailed product information and specifications
- Secure payment and personal data handling
- Order tracking and management capabilities
- Responsive customer support
- Competitive pricing and product availability

Value Received:

- Convenient 24/7 online shopping
 - Access to wide range of technology products
 - Transparent pricing and product information
 - Time-saving compared to physical store visits
 - Order history and tracking capabilities
-

2. System Administrators (Store Managers and Staff)

Profile:

- Store managers and inventory specialists
- Responsible for daily operations and product management
- Basic to intermediate technical skills
- Work schedule: Regular business hours with some flexibility

Interests and Needs:

- Efficient product catalog management
- Real-time inventory tracking
- Order processing and fulfillment tools
- Customer management capabilities
- Easy-to-understand analytics and reporting
- Reliable system with minimal downtime

Value Received:

- Streamlined inventory management
 - Automated order processing
 - Business insights through analytics
 - Reduced manual workload
 - Centralized control panel for all operations
-

3. Business Owners / Store Managers

Profile:

- Owners of technology retail businesses
- Decision-makers for business strategy
- Focus on profitability and growth
- Need comprehensive business overview

Interests and Needs:

- Sales analytics and business performance metrics
- Revenue tracking and reporting
- Customer behavior insights
- Inventory optimization data
- Return on investment (ROI) visibility
- Scalability for business growth

Value Received:

- Data-driven business decisions
 - Improved operational efficiency
 - Increased revenue potential through online channel
 - Reduced operational costs
 - Competitive advantage in market
-

3.1.2. Secondary Stakeholders

4. Development Team

Profile:

- 4 software developers (full-stack)
- Responsible for building, maintaining, and evolving the system
- Varied experience levels with new technologies

Interests and Needs:

- Clear system requirements and specifications
- Maintainable and scalable codebase
- Comprehensive documentation
- Modern development tools and practices
- Collaborative development environment

Value Received:

- Learning experience with modern tech stack
 - Portfolio-worthy project
 - Practical application of software engineering principles
 - Teamwork and collaboration skills
-

5. Technical Support Staff (Future)

Profile:

- IT staff responsible for system maintenance
- Handle technical issues and user support
- Monitor system performance and health

Interests and Needs:

- System monitoring and logging capabilities
- Error tracking and reporting
- Documentation for troubleshooting
- User activity logs
- System health metrics

Value Received:

- Tools for efficient problem resolution
 - Clear system architecture understanding
 - Monitoring dashboards
-

3.1.3. Stakeholder Summary Matrix

| Stakeholder Group | Primary Goals | System Usage Frequency | Technical Expertise | Priority Level |
|-----------------------|--------------------------------|------------------------------|-----------------------|----------------|
| End Customers | Purchase products conveniently | High (Daily/Weekly) | Basic to Intermediate | Critical |
| System Administrators | Manage operations efficiently | Very High (Daily) | Intermediate | Critical |
| Business Owners | Maximize revenue and insights | Medium (Weekly) | Basic to Intermediate | High |
| Development Team | Build and maintain system | Very High (Daily during dev) | Advanced | High |
| Technical Support | Maintain system health | High (as needed) | Advanced | Medium |

3.2. Use Case Model

This section presents the complete use case model for the Microshop E-Commerce Platform, including graphical representations using Mermaid diagrams and detailed textual descriptions for each use case.

3.2.1. Graphical Use Case Model

System Boundary: Microshop E-Commerce Platform

Actors:

- **Customer:** End user who browses and purchases products
 - **Administrator:** System admin who manages products, orders, and users
 - **System:** Automated processes (e.g., stock reservation, notifications)
-

Use Case Diagram 1: General System Overview (Complete View)

This diagram shows all actors and all use cases in the Microshop E-Commerce Platform.

mermaid

```
%%{init: {'theme':'base'}}%%  
graph TB  
Customer((Customer<br/>Guest/Registered))  
Admin((Administrator))  
System((System))
```

```
subgraph "Microshop E-Commerce Platform"
```

```
subgraph "Authentication Module"
```

```
UC01[UC01: Register]  
UC02[UC02: Login]  
UC16[UC16: Logout]  
UC17[UC17: Admin Login]  
end
```

```
subgraph "Product Module"
```

```
UC03[UC03: Browse Products]  
UC04[UC04: Search Products]  
UC05[UC05: Filter Products]  
UC06[UC06: View Details]  
UC19[UC19: Create Product]  
UC20[UC20: Edit Product]  
UC21[UC21: Delete Product]  
UC22[UC22: Manage Inventory]  
end
```

```
subgraph "Shopping Cart Module"
```

```
UC07[UC07: Add to Cart]  
UC08[UC08: View Cart]  
UC09[UC09: Update Quantity]  
UC10[UC10: Remove Item]  
end
```

```
subgraph "Order Module"
```

```
UC11[UC11: Checkout]  
UC12[UC12: View Orders]  
UC13[UC13: View Order Details]  
UC14[UC14: Track Status]  
UC23[UC23: View All Orders]  
UC24[UC24: Update Status]  
end
```

```
subgraph "Admin Dashboard Module"
```

```
UC18[UC18: Dashboard]  
UC25[UC25: Analytics]  
UC26[UC26: Reports]  
UC27[UC27: Manage Users]  
UC28[UC28: Generate Report]
```

UC28[UC28: System Logs]

end

subgraph "Profile Module"

UC15[UC15: Update Profile]

end

end

Customer --> UC01

Customer --> UC02

Customer --> UC03

Customer --> UC04

Customer --> UC05

Customer --> UC06

Customer --> UC07

Customer --> UC08

Customer --> UC09

Customer --> UC10

Customer --> UC11

Customer --> UC12

Customer --> UC13

Customer --> UC14

Customer --> UC15

Customer --> UC16

Admin --> UC17

Admin --> UC18

Admin --> UC19

Admin --> UC20

Admin --> UC21

Admin --> UC22

Admin --> UC23

Admin --> UC24

Admin --> UC25

Admin --> UC26

Admin --> UC27

Admin --> UC28

System -.monitors.-> UC11

System -.updates.-> UC22

System -.notifies.-> Customer

UC03 -.includes.-> UC05

UC11 -.requires.-> UC02

UC12 -.requires.-> UC02

UC18 -.includes.-> UC25

Legend:

- **Solid lines:** Direct interactions (actor performs use case)
 - **Dashed lines with "includes":** One use case includes another
 - **Dashed lines with "requires":** Precondition relationship
 - **Dashed lines with "monitors/updates/notifies":** System automated actions
-

Use Case Diagram 2: Customer Authentication & Profile Management (Detailed)

This diagram details the authentication and profile management flow for customers.

mermaid

```
%%{init: {'theme':'base'}}%%
```

```
graph TB
```

```
Customer((Customer))  
Guest((Guest User))  
System((System))
```

```
subgraph "Authentication & Profile Management"
```

```
    direction TB
```

```
    subgraph "Registration Process"
```

```
        UC01[UC01: Register Account]  
        UC01_1[Validate Email]  
        UC01_2[Check Password Strength]  
        UC01_3[Create User Record]  
        UC01_4[Generate JWT Token]
```

```
    end
```

```
    subgraph "Login Process"
```

```
        UC02[UC02: Login]  
        UC02_1[Validate Credentials]  
        UC02_2[Generate JWT Token]  
        UC02_3[Load User Session]
```

```
    end
```

```
    subgraph "Profile Management"
```

```
        UC15[UC15: Update Profile]  
        UC15_1[Change Password]  
        UC15_2[Update Contact Info]  
        UC15_3[Manage Addresses]
```

```
    end
```

```
    UC16[UC16: Logout]
```

```
    UC16_1[Invalidate Token]  
    UC16_2[Clear Session]
```

```
end
```

```
Guest --> UC01
```

```
UC01 --> UC01_1
```

```
UC01_1 --> UC01_2
```

```
UC01_2 --> UC01_3
```

```
UC01_3 --> UC01_4
```

```
UC01_4 -.auto login.-> UC02_3
```

```
Guest --> UC02
```

```
Customer --> UC02
```

```
UC02 --> UC02_1
```

```
UC02_1 --> UC02_2
```

```
UC02_2 --> UC02_3
```

UC02_2 --> UC02_3

Customer --> UC15

UC15 --> UC15_1

UC15 --> UC15_2

UC15 --> UC15_3

Customer --> UC16

UC16 --> UC16_1

UC16_1 --> UC16_2

System -.validates.-> UC01_1

System -.hashes password.-> UC01_2

System -.stores.-> UC01_3

System -.authenticates.-> UC02_1

style UC01 fill:#e1f5ff

style UC02 fill:#e1f5ff

style UC15 fill:#fff3e0

style UC16 fill:#fff3e0

Use Case Diagram 3: Product Discovery & Browsing (Detailed)

This diagram shows the complete product discovery flow from browsing to viewing details.

mermaid

```
%%{init: {'theme':'base'}}%%  
graph TB  
Customer((Customer/Guest))  
System((System))  
  
subgraph "Product Discovery Flow"  
direction TB  
  
subgraph "Browse & Search"  
UC03[UC03: Browse Products]  
UC03_1[Load Product List]  
UC03_2[Apply Pagination]  
UC03_3[Display Grid View]  
  
UC04[UC04: Search Products]  
UC04_1[Enter Search Query]  
UC04_2[Search Algorithm]  
UC04_3[Display Results]  
UC04_4[Show Autocomplete]  
end  
  
subgraph "Filter & Sort"  
UC05[UC05: Filter Products]  
UC05_1[Filter by Category]  
UC05_2[Filter by Brand]  
UC05_3[Filter by Price]  
UC05_4[Filter by Stock]  
  
Sort[Sort Products]  
Sort_1[By Price]  
Sort_2[By Name]  
Sort_3[By Date]  
end  
  
subgraph "Product Details"  
UC06[UC06: View Product Details]  
UC06_1[Load Product Info]  
UC06_2[Display Images]  
UC06_3[Show Specifications]  
UC06_4[Check Stock Status]  
UC06_5[Show Similar Products]  
end  
end
```

Customer --> UC03

Customer --> UC04

UC03 --> UC03_1
UC03_1 --> UC03_2
UC03_2 --> UC03_3

UC04 --> UC04_1
UC04_1 --> UC04_2
UC04_2 --> UC04_3
UC04 -.suggests.-> UC04_4

UC03 -.includes.-> UC05
UC04 -.includes.-> UC05

UC05 --> UC05_1
UC05 --> UC05_2
UC05 --> UC05_3
UC05 --> UC05_4

UC03 -.includes.-> Sort
UC04 -.includes.-> Sort
Sort --> Sort_1
Sort --> Sort_2
Sort --> Sort_3

Customer --> UC06
UC03_3 -.click product.-> UC06
UC04_3 -.click product.-> UC06

UC06 --> UC06_1
UC06_1 --> UC06_2
UC06_1 --> UC06_3
UC06_1 --> UC06_4
UC06 -.shows.-> UC06_5

System -.retrieves data.-> UC03_1
System -.executes search.-> UC04_2
System -.applies filters.-> UC05
System -.loads details.-> UC06_1
System -.checks inventory.-> UC06_4

style UC03 fill:#e8f5e9
style UC04 fill:#e8f5e9
style UC05 fill:#fff9c4
style UC06 fill:#e1f5ff

Use Case Diagram 4: Shopping Cart & Checkout Process (Detailed)

This diagram illustrates the complete shopping cart management and checkout flow.

mermaid

```
%%{init: {'theme':'base'}}%%  
graph TB  
Customer((Customer))  
System((System))  
  
subgraph "Shopping Cart & Checkout Flow"  
direction TB  
  
subgraph "Cart Management"  
UC07[UC07: Add to Cart]  
UC07_1[Select Quantity]  
UC07_2[Validate Stock]  
UC07_3[Add/Update Cart Item]  
UC07_4[Update Cart Count]  
  
UC08[UC08: View Cart]  
UC08_1[Retrieve Cart Items]  
UC08_2[Validate Prices]  
UC08_3[Calculate Total]  
UC08_4[Display Cart Summary]  
  
UC09[UC09: Update Quantity]  
UC09_1[Change Quantity]  
UC09_2[Recalculate Totals]  
  
UC10[UC10: Remove from Cart]  
UC10_1[Confirm Removal]  
UC10_2[Update Cart]  
end  
  
subgraph "Checkout Process"  
UC11[UC11: Checkout]  
UC11_1[Verify Login]  
UC11_2[Enter Shipping Info]  
UC11_3[Review Order]  
UC11_4[Validate Order]  
UC11_5[Create Order]  
UC11_6[Reserve Stock]  
UC11_7[Clear Cart]  
UC11_8[Show Confirmation]  
end  
end
```

Customer --> UC07

UC07 --> UC07_1

UC07_1 --> UC07_2

UC07_2 --> UC07_3

UC07_3 --> UC07_4

UC07_3 --> UC07_4

Customer --> UC08

UC08 --> UC08_1

UC08_1 --> UC08_2

UC08_2 --> UC08_3

UC08_3 --> UC08_4

Customer --> UC09

UC09 --> UC09_1

UC09_1 --> UC09_2

Customer --> UC10

UC10 --> UC10_1

UC10_1 --> UC10_2

UC08 --> UC11

Customer --> UC11

UC11 --> UC11_1

UC11_1 --> UC11_2

UC11_2 --> UC11_3

UC11_3 --> UC11_4

UC11_4 --> UC11_5

UC11_5 --> UC11_6

UC11_6 --> UC11_7

UC11_7 --> UC11_8

System -.checks availability.-> UC07_2

System -.stores cart.-> UC07_3

System -.loads cart.-> UC08_1

System -.validates.-> UC08_2

System -.updates totals.-> UC09_2

System -.removes item.-> UC10_2

System -.validates order.-> UC11_4

System -.creates order.-> UC11_5

System -.decrements stock.-> UC11_6

System -.empties cart.-> UC11_7

style UC07 fill:#e1f5ff

style UC08 fill:#e1f5ff

style UC09 fill:#fff3e0

style UC10 fill:#fff3e0

style UC11 fill:#e8f5e9

Use Case Diagram 5: Order Management - Customer View (Detailed)

This diagram shows how customers interact with their orders.

mermaid

```
%%{init: {'theme':'base'}}%%
```

```
graph TB
```

```
Customer((Customer))
```

```
System((System))
```

```
subgraph "Customer Order Management"
```

```
    direction TB
```

```
    UC12[UC12: View Order History]
```

```
    UC12_1[Retrieve User Orders]
```

```
    UC12_2[Display Order List]
```

```
    UC12_3[Filter by Status]
```

```
    UC12_4[Sort by Date]
```

```
    UC13[UC13: View Order Details]
```

```
    UC13_1[Load Order Data]
```

```
    UC13_2[Display Items]
```

```
    UC13_3[Show Shipping Info]
```

```
    UC13_4[Display Payment Info]
```

```
    UC13_5[Show Status Timeline]
```

```
    UC14[UC14: Track Order Status]
```

```
    UC14_1[Display Status Timeline]
```

```
    UC14_2>Show Current Status
```

```
    UC14_3[Estimate Delivery]
```

```
    UC14_4>Show Status History]
```

```
Cancel[Cancel Order]
```

```
Cancel_1[Verify Status Pending]
```

```
Cancel_2[Confirm Cancellation]
```

```
Cancel_3[Restore Stock]
```

```
Cancel_4[Update Status]
```

```
end
```

```
Customer --> UC12
```

```
UC12 --> UC12_1
```

```
UC12_1 --> UC12_2
```

```
UC12 --> UC12_3
```

```
UC12 --> UC12_4
```

```
UC12_2 -.select order.-> UC13
```

```
Customer --> UC13
```

```
UC13 --> UC13_1
```

```
UC13_1 --> UC13_2
```

```
UC13_1 --> UC13_3
```

```
UC13_1 --> UC13_4
```

```
UC13_1 --> UC13_5
```

UC13 --> UC14

Customer --> UC14

UC14 --> UC14_1

UC14_1 --> UC14_2

UC14_1 --> UC14_3

UC14_1 --> UC14_4

UC13 -.if pending.-> Cancel

Cancel --> Cancel_1

Cancel_1 --> Cancel_2

Cancel_2 --> Cancel_3

Cancel_3 --> Cancel_4

System -.fetches orders.-> UC12_1

System -.loads details.-> UC13_1

System -.tracks status.-> UC14_1

System -.restores inventory.-> Cancel_3

style UC12 fill:#e1f5ff

style UC13 fill:#e8f5e9

style UC14 fill:#fff9c4

style Cancel fill:#ffeb3b

Use Case Diagram 6: Admin Product Management (Detailed)

This diagram details the administrator's product management capabilities.

mermaid

```
%%{init: {'theme':'base'}}%%
```

```
graph TB
```

```
    Admin((Administrator))
```

```
    System((System))
```

```
subgraph "Admin Product Management"
```

```
    direction TB
```

```
    UC19[UC19: Create Product]
```

```
    UC19_1[Enter Product Info]
```

```
    UC19_2[Add Specifications]
```

```
    UC19_3[Upload Images]
```

```
    UC19_4[Set Stock & Price]
```

```
    UC19_5[Validate Data]
```

```
    UC19_6[Save Product]
```

```
    UC20[UC20: Edit Product]
```

```
    UC20_1[Load Product Data]
```

```
    UC20_2[Modify Fields]
```

```
    UC20_3[Update Stock]
```

```
    UC20_4[Change Price]
```

```
    UC20_5[Validate Changes]
```

```
    UC20_6[Save Changes]
```

```
    UC20_7[Log Changes]
```

```
    UC21[UC21: Delete Product]
```

```
    UC21_1[Select Product]
```

```
    UC21_2[Check Dependencies]
```

```
    UC21_3[Confirm Deletion]
```

```
    UC21_4[Soft Delete]
```

```
    UC21_5[Log Deletion]
```

```
    UC22[UC22: Manage Inventory]
```

```
    UC22_1[View Stock Levels]
```

```
    UC22_2[Identify Low Stock]
```

```
    UC22_3[Restock Items]
```

```
    UC22_4[Set Alerts]
```

```
    UC22_5[Export Report]
```

```
end
```

```
Admin --> UC19
```

```
UC19 --> UC19_1
```

```
UC19_1 --> UC19_2
```

```
UC19_2 --> UC19_3
```

```
UC19_3 --> UC19_4
```

```
UC19_4 --> UC19_5
```

```
UC19_5 --> UC19_6
```

Admin --> UC20
UC20 --> UC20_1
UC20_1 --> UC20_2
UC20_2 --> UC20_3
UC20_2 --> UC20_4
UC20_3 --> UC20_5
UC20_4 --> UC20_5
UC20_5 --> UC20_6
UC20_6 --> UC20_7

Admin --> UC21
UC21 --> UC21_1
UC21_1 --> UC21_2
UC21_2 --> UC21_3
UC21_3 --> UC21_4
UC21_4 --> UC21_5

Admin --> UC22
UC22 --> UC22_1
UC22_1 --> UC22_2
UC22_2 --> UC22_3
UC22 --> UC22_4
UC22 --> UC22_5

UC22 -.includes.-> UC19
UC22 -.includes.-> UC20

System -.validates.-> UC19_5
System -.stores.-> UC19_6
System -.loads.-> UC20_1
System -.validates.-> UC20_5
System -.updates.-> UC20_6
System -.checks orders.-> UC21_2
System -.marks deleted.-> UC21_4
System -.monitors.-> UC22_2
System -.generates.-> UC22_5

style UC19 fill:#e8f5e9
style UC20 fill:#fff9c4
style UC21 fill:#ffebee
style UC22 fill:#e1f5ff

Use Case Diagram 7: Admin Order & Analytics Management (Detailed)

This diagram shows administrator's order management and analytics capabilities.

mermaid

```
%%{init: {'theme':'base'}}%%  
graph TB  
    Admin((Administrator))  
    System((System))  
  
    subgraph "Admin Order & Analytics Management"  
        direction TB  
  
        subgraph "Dashboard & Analytics"  
            UC18[UC18: View Dashboard]  
            UC18_1[Load KPIs]  
            UC18_2[Display Charts]  
            UC18_3>Show Recent Orders  
            UC18_4[Display Alerts]  
  
            UC25[UC25: View Analytics]  
            UC25_1>Select Time Period  
            UC25_2[Calculate Metrics]  
            UC25_3[Generate Charts]  
            UC25_4>Show Trends  
  
            UC26[UC26: Generate Reports]  
            UC26_1>Select Report Type  
            UC26_2[Set Parameters]  
            UC26_3[Collect Data]  
            UC26_4[Format Report]  
            UC26_5[Export Report]  
        end  
  
        subgraph "Order Management"  
            UC23[UC23: View All Orders]  
            UC23_1[Load Order List]  
            UC23_2[Apply Filters]  
            UC23_3[Search Orders]  
            UC23_4[Sort Orders]  
  
            UC24[UC24: Update Order Status]  
            UC24_1>Select Order  
            UC24_2[Choose New Status]  
            UC24_3[Validate Transition]  
            UC24_4[Update Status]  
            UC24_5[Log Change]  
            UC24_6[Notify Customer]  
            UC24_7[Restore Stock if Cancel]  
        end  
    end
```

Admin --> UC18
UC18 --> UC18_1
UC18_1 --> UC18_2
UC18_1 --> UC18_3
UC18_1 --> UC18_4
UC18 -.includes.-> UC25

Admin --> UC25
UC25 --> UC25_1
UC25_1 --> UC25_2
UC25_2 --> UC25_3
UC25_3 --> UC25_4

Admin --> UC26
UC26 --> UC26_1
UC26_1 --> UC26_2
UC26_2 --> UC26_3
UC26_3 --> UC26_4
UC26_4 --> UC26_5

Admin --> UC23
UC23 --> UC23_1
UC23 --> UC23_2
UC23 --> UC23_3
UC23 --> UC23_4

UC23 -.select order.-> UC24
Admin --> UC24
UC24 --> UC24_1
UC24_1 --> UC24_2
UC24_2 --> UC24_3
UC24_3 --> UC24_4
UC24_4 --> UC24_5
UC24_5 --> UC24_6
UC24_4 -.if cancelled.-> UC24_7

System -.retrieves.-> UC18_1
System -.calculates.-> UC25_2
System -.generates.-> UC25_3
System -.compiles.-> UC26_3
System -.formats.-> UC26_4
System -.fetches.-> UC23_1
System -.validates.-> UC24_3
System -.updates.-> UC24_4
System -.logs.-> UC24_5
System -.sends notification.-> UC24_6
System -.increments stock.-> UC24_7

style UC18 fill:#e1f5ff

style UC23 fill:#e1f5ff

style UC23 fill:#fff9c4

style UC24 fill:#e8f5e9

style UC25 fill:#fff9c4

style UC26 fill:#fff3e0

Use Case Diagram 8: System Relationships & Dependencies

This diagram shows the relationships and dependencies between different use cases.

mermaid

```
%%{init: {'theme':'base'}}%%  
graph TB  
    subgraph "Use Case Relationships"  
        direction TB  
  
        subgraph "Authentication Flow"  
            UC01[Register Account]  
            UC02[Login]  
            UC16[Logout]  
        end  
  
        subgraph "Product Flow"  
            UC03[Browse Products]  
            UC04[Search Products]  
            UC05[Filter Products]  
            UC06[View Details]  
        end  
  
        subgraph "Purchase Flow"  
            UC07[Add to Cart]  
            UC08[View Cart]  
            UC09[Update Quantity]  
            UC10[Remove Item]  
            UC11[Checkout]  
        end  
  
        subgraph "Post-Purchase Flow"  
            UC12[View Orders]  
            UC13[View Order Details]  
            UC14[Track Status]  
        end  
  
        subgraph "Admin Flow"  
            UC17[Admin Login]  
            UC18[Dashboard]  
            UC19[Create Product]  
            UC20[Edit Product]  
            UC22[Manage Inventory]  
            UC23[View All Orders]  
            UC24[Update Status]  
        end  
    end
```

UC01 -.auto login.-> UC02
UC02 -.required for.-> UC11
UC02 -.required for.-> UC12

UC03 -.includes.-> UC05

UC04 -.includes.-> UC05

UC03 -.leads to.-> UC06

UC04 -.leads to.-> UC06

UC06 -.enables.-> UC07

UC07 -.updates.-> UC08

UC08 -.allows.-> UC09

UC08 -.allows.-> UC10

UC08 -.proceeds to.-> UC11

UC11 -.creates.-> UC12

UC12 -.shows.-> UC13

UC13 -.enables.-> UC14

UC17 -.grants access to.-> UC18

UC18 -.includes.-> UC19

UC18 -.includes.-> UC20

UC18 -.includes.-> UC22

UC22 -.includes.-> UC19

UC22 -.includes.-> UC20

UC23 -.enables.-> UC24

style UC02 fill:#ffccdd2

style UC11 fill:#ffccdd2

style UC17 fill:#ffccdd2

style UC18 fill:#c8e6c9

style UC08 fill:#c8e6c9

3.2.2. Textual Description for Each Use Case

This section provides detailed textual descriptions for each use case identified in the graphical models.

Use Case UC01: Register Account

| Element | Description |
|------------------|----------------------------------------------------------------------------------------------------|
| Use Case ID | UC01 |
| Use Case Name | Register Account |
| Actor(s) | Customer (Primary) |
| Description | Allows a new customer to create an account on the platform |
| Preconditions | - Customer is not logged in - Customer has valid email address |
| Postconditions | - New user account is created - User receives confirmation - User is logged in automatically |
| Priority | High |
| Frequency of Use | Medium (New users only) |

Main Flow:

1. Customer navigates to registration page
2. System displays registration form
3. Customer enters required information:
 - Full name
 - Email address
 - Password
 - Confirm password
 - Phone number (optional)
4. Customer clicks "Register" button
5. System validates input data:
 - Checks email format
 - Verifies email is not already registered
 - Checks password strength (minimum 6 characters)
 - Confirms password match
6. System creates new user account
7. System sends welcome email (optional)
8. System automatically logs in the user
9. System redirects to homepage or previous page
10. System displays success message

Alternative Flows:

AF1: Email Already Exists

- At step 5, if email is already registered:
 - System displays error message "Email already exists"
 - System suggests login or password recovery
 - Customer can modify email or go to login page

AF2: Weak Password

- At step 5, if password doesn't meet requirements:
 - System displays error message with password requirements
 - Customer must enter a stronger password
 - Flow returns to step 3

AF3: Validation Errors

- At step 5, if any validation fails:
 - System highlights fields with errors
 - System displays specific error messages for each field
 - Customer corrects errors
 - Flow returns to step 3

Exception Flows:

EF1: Server Error

- System encounters database error during account creation
- System displays error message "Registration failed, please try again"
- System logs error for technical team
- Customer can retry registration

Business Rules:

- BR001: Email must be unique across all users
- BR002: Password must be at least 6 characters
- BR003: Email format must be valid (contain @ and domain)
- BR004: All required fields must be filled

Use Case UC02: Login

| Element | Description |
|------------------|----------------------------------------------------------------------------------------------------------------|
| Use Case ID | UC02 |
| Use Case Name | Login |
| Actor(s) | Customer (Primary) |
| Description | Allows registered customer to authenticate and access their account |
| Preconditions | - Customer has registered account - Customer is not currently logged in |
| Postconditions | - Customer is authenticated - JWT token is issued - Customer is redirected to requested page or homepage |
| Priority | Critical |
| Frequency of Use | Very High (Every session) |

Main Flow:

1. Customer navigates to login page
2. System displays login form
3. Customer enters credentials:
 - Email address
 - Password
4. Customer clicks "Login" button
5. System validates credentials:
 - Checks email exists
 - Verifies password matches
6. System generates JWT token
7. System stores token in browser (localStorage or cookie)
8. System redirects customer to:
 - Previous page (if navigated from specific page)
 - Homepage (if direct login)
9. System displays welcome message with customer name

Alternative Flows:

AF1: Invalid Credentials

- At step 5, if email not found or password incorrect:
 - System displays error message "Invalid email or password"
 - System increments failed login attempt counter
 - Customer can retry login
 - Flow returns to step 3

AF2: Account Locked

- At step 5, if account has too many failed attempts:
 - System displays message "Account temporarily locked"
 - System suggests password reset
 - Customer must wait or reset password

AF3: Remember Me

- At step 3, customer checks "Remember Me" option
- At step 7, system stores token with extended expiration
- User remains logged in for 30 days

Exception Flows:

EF1: Server Connection Error

- System cannot connect to authentication service
- System displays error message
- Customer can retry login

Business Rules:

- BR005: Account locks after 5 failed login attempts
- BR006: JWT token expires after 24 hours (or 30 days if "Remember Me")
- BR007: Password is encrypted using bcrypt

Use Case UC03: Browse Products

| Element | Description |
|------------------|------------------------------------------------------------------------------------------------|
| Use Case ID | UC03 |
| Use Case Name | Browse Products |
| Actor(s) | Customer (Primary), Guest (Secondary) |
| Description | Allows users to view product catalog with pagination |
| Preconditions | - System has products in database |
| Postconditions | - Customer views list of available products - Products are displayed with basic information |
| Priority | Critical |
| Frequency of Use | Very High |

Main Flow:

1. Customer navigates to products page or homepage
2. System retrieves products from database (default: newest first)
3. System displays products in grid layout (12 per page):
 - Product image
 - Product name
 - Price in VND
 - Category (Phone/Laptop)
 - Stock status indicator
4. Customer scrolls through products
5. System loads additional products (pagination)
6. Customer can click on product to view details (UC06)

Alternative Flows:

AF1: Filter Products (includes UC05)

- At step 2, customer applies filters:
 - Category (Phone/Laptop)
 - Brand
 - Price range
- System retrieves filtered products
- System displays filtered results
- Flow continues from step 3

AF2: Sort Products

- At step 2, customer selects sort option:
 - Price: Low to High
 - Price: High to Low
 - Newest
 - Name A-Z
- System sorts products accordingly
- Flow continues from step 3

AF3: No Products Available

- At step 2, if no products match criteria:
 - System displays message "No products found"
 - System suggests removing filters or browsing all products

Exception Flows:

EF1: Loading Error

- System fails to retrieve products
- System displays error message
- System suggests refresh or contact support

Business Rules:

- BR008: Display 12 products per page
- BR009: Out-of-stock products shown but marked clearly
- BR010: Products sorted by creation date (newest first) by default

Use Case UC04: Search Products

| Element | Description |
|-------------------------|---------------------------------------------------------------------------|
| Use Case ID | UC04 |
| Use Case Name | Search Products |
| Actor(s) | Customer (Primary), Guest (Secondary) |
| Description | Allows users to search for products using keywords |
| Preconditions | - System has products in database |
| Postconditions | - Customer views search results - Search query is logged for analytics |
| Priority | High |
| Frequency of Use | High |

Main Flow:

1. Customer enters search query in search bar
2. Customer presses Enter or clicks search icon
3. System processes search query:
 - Searches in product name
 - Searches in product description
 - Searches in product specifications
4. System retrieves matching products
5. System displays search results:
 - Number of results found
 - Matching products in grid layout
 - Highlighted search terms (optional)
6. System logs search query for analytics
7. Customer can view product details or refine search

Alternative Flows:

AF1: No Results Found

- At step 4, if no products match search:
 - System displays "No results found for '[query]'"
 - System suggests:
 - Check spelling
 - Use different keywords
 - Browse all products
 - System may show "Related Products" based on category

AF2: Autocomplete Suggestions

- At step 1, as customer types:
 - System shows autocomplete suggestions
 - Suggestions based on:
 - Popular searches
 - Product names
 - Categories
 - Customer can select suggestion to search immediately

AF3: Search with Filters

- At step 2, customer applies additional filters
- System combines search query with filters
- Flow continues from step 3

Exception Flows:

EF1: Search Service Unavailable

- System search service is down
- System displays message
- System falls back to basic product browsing

Business Rules:

- BR011: Search is case-insensitive
 - BR012: Minimum 2 characters required for search
 - BR013: Search includes product name, description, brand, specs
 - BR014: Search queries logged for analytics (privacy-compliant)
-

Use Case UC05: Filter Products

| Element | Description |
|-------------------------|---------------------------------------------------------------------------|
| Use Case ID | UC05 |
| Use Case Name | Filter Products |
| Actor(s) | Customer (Primary), Guest (Secondary) |
| Description | Allows users to filter products by various criteria |
| Preconditions | - Customer is on products page or search results page |
| Postconditions | - Filtered product list is displayed - Filter selections are preserved |
| Priority | Medium |
| Frequency of Use | High |

Main Flow:

1. Customer views products page with filter options sidebar
2. System displays available filters:
 - Category (Phone/Laptop)
 - Brand (Apple, Samsung, Dell, HP, Lenovo, etc.)
 - Price Range (with slider or predefined ranges)
 - Stock Status (In Stock / Out of Stock)
3. Customer selects one or more filters
4. System applies filters in real-time or after clicking "Apply"
5. System retrieves filtered products from database
6. System displays filtered results
7. System shows active filters with option to remove individual filters
8. System displays count of results

Alternative Flows:

AF1: Clear All Filters

- Customer clicks "Clear All Filters" button
- System removes all active filters
- System displays all products
- Flow returns to UC03

AF2: Multiple Filter Combination

- Customer applies multiple filters across categories
- System uses AND logic between filter categories
- System uses OR logic within same category
- Example: (Category: Phone OR Laptop) AND (Brand: Apple) AND (Price: 10M-20M)

AF3: No Results After Filtering

- At step 5, no products match filter criteria
- System displays "No products match your filters"
- System suggests:
 - Remove some filters
 - Adjust price range
 - Clear all and browse all products
- System shows number of products for each filter option

Exception Flows:

EF1: Filter Loading Error

- System fails to apply filters
- System displays error message
- System reverts to previous state

Business Rules:

- BR015: Filters are ANDed across categories, ORed within category
 - BR016: Filter state preserved during session
 - BR017: Price ranges in VND (Vietnamese Dong)
 - BR018: Out-of-stock products can be filtered out
-

Use Case UC06: View Product Details

| Element | Description |
|------------------|-----------------------------------------------------------------------------------|
| Use Case ID | UC06 |
| Use Case Name | View Product Details |
| Actor(s) | Customer (Primary), Guest (Secondary) |
| Description | Allows users to view comprehensive information about a specific product |
| Preconditions | - Product exists in database - Customer clicked on product from listing |
| Postconditions | - Detailed product information is displayed - Customer can add product to cart |
| Priority | Critical |
| Frequency of Use | Very High |

Main Flow:

1. Customer clicks on product from product listing
2. System retrieves product details from database
3. System displays product detail page with:
 - Product images (multiple if available)
 - Product name
 - Price in VND
 - Stock status (In Stock / Out of Stock / Low Stock)
 - Category (Phone/Laptop)
 - Brand
 - Detailed specifications:
 - For Phones: Screen size, processor, RAM, storage, camera, battery, OS
 - For Laptops: Screen size, processor, RAM, storage, graphics, battery, OS
 - Product description
 - Quantity selector
 - "Add to Cart" button
 - Stock availability indicator

4. Customer can:
 - View images (click to enlarge)
 - Read specifications
 - Select quantity
 - Add to cart (UC07)

5. System tracks product view for analytics

Alternative Flows:

AF1: Out of Stock Product

- At step 3, if product is out of stock:
 - System displays "Out of Stock" message
 - "Add to Cart" button is disabled
 - System may show "Notify Me" option (future feature)
 - Customer can still view all details

AF2: Low Stock Warning

- At step 3, if product stock < 5:
 - System displays "Only X items left!" warning
 - Adds urgency indicator
 - Customer can proceed normally

AF3: Similar Products Suggestion

- At step 3, system displays "Similar Products" section
- Shows 4-6 products from same category or brand
- Customer can click to view those products

Exception Flows:

EF1: Product Not Found

- Product ID invalid or product deleted
- System displays "Product not found" error
- System redirects to products page after 3 seconds

EF2: Loading Error

- System fails to load product details
- System displays error message
- System offers retry option or return to products page

Business Rules:

- BR019: Product views tracked for analytics
- BR020: Maximum quantity per order: 10 units
- BR021: Low stock threshold: 5 units
- BR022: All prices displayed in VND

Use Case UC07: Add to Cart

| Element | Description |
|-------------------------|-----------------------------------------------------------------------------------|
| Use Case ID | UC07 |
| Use Case Name | Add to Cart |
| Actor(s) | Customer (Primary) |
| Description | Allows customer to add a product to their shopping cart |
| Preconditions | - Product is in stock - Customer is on product details page or product listing |
| Postconditions | - Product added to cart - Cart count updated - Stock temporarily reserved |
| Priority | Critical |
| Frequency of Use | Very High |

Main Flow:

1. Customer views product details (UC06)
2. Customer selects quantity (default: 1)
3. Customer clicks "Add to Cart" button
4. System validates:
 - Product is still in stock
 - Requested quantity is available
 - Quantity doesn't exceed limit (max 10 per product)
5. System adds product to cart:
 - If product already in cart: increases quantity
 - If new product: adds new cart item
6. System updates cart count in header
7. System displays success notification:
 - "Product added to cart"
 - Options: "View Cart" or "Continue Shopping"
8. System saves cart to:
 - Backend database (if logged in)
 - Browser localStorage (if guest)

Alternative Flows:

AF1: Product Already in Cart

- At step 5, if product exists in cart:
 - System increases existing quantity instead of adding duplicate
 - System displays message "Quantity updated in cart"

AF2: Insufficient Stock

- At step 4, if requested quantity > available stock:
 - System displays error message "Only X items available"
 - System suggests maximum available quantity
 - Customer can adjust quantity or cancel

AF3: Guest User (Not Logged In)

- At step 8:
 - System stores cart in browser localStorage
 - System displays reminder to login/register for checkout
 - Cart persists during browser session

AF4: Maximum Quantity Reached

- At step 4, if quantity would exceed limit:
 - System displays "Maximum 10 items per product"
 - System sets quantity to maximum allowed
 - Customer can proceed or adjust

Exception Flows:

EF1: Product Became Out of Stock

- Between step 3 and 4, product went out of stock
- System displays "Product is no longer available"
- System suggests similar products
- Cart operation cancelled

EF2: Cart Service Error

- System fails to add to cart
- System displays error message
- System suggests retry
- Cart state remains unchanged

Business Rules:

- BR023: Maximum 10 units per product per customer
- BR024: Cart stored in database for logged-in users
- BR025: Guest cart stored in localStorage for 7 days
- BR026: Cart merged when guest logs in
- BR027: Stock not reserved until checkout starts

Use Case UC08: View Cart

| Element | Description |
|------------------|--------------------------------------------------------------|
| Use Case ID | UC08 |
| Use Case Name | View Cart |
| Actor(s) | Customer (Primary) |
| Description | Allows customer to view all items currently in shopping cart |
| Preconditions | - Customer has added at least one item to cart |
| Postconditions | - Cart contents are displayed - Total price calculated |
| Priority | Critical |
| Frequency of Use | High |

Main Flow:

1. Customer clicks cart icon in header or "View Cart" button

2. System retrieves cart items:

- From database (logged-in users)
- From localStorage (guest users)

3. System validates cart items:

- Checks products still exist
- Verifies stock availability
- Updates prices if changed

4. System displays cart page with:

- List of cart items (each showing):
 - Product image (thumbnail)
 - Product name
 - Unit price
 - Quantity selector
 - Subtotal (price × quantity)
 - Remove button
- Cart summary:
 - Subtotal
 - Tax (if applicable)
 - Shipping (if applicable)
 - Total amount in VND
- "Continue Shopping" button
- "Proceed to Checkout" button

5. System calculates and displays total

Alternative Flows:

AF1: Empty Cart

- At step 2, if cart is empty:
 - System displays "Your cart is empty"
 - System shows "Start Shopping" button
 - System may show recommended products

AF2: Product No Longer Available

- At step 3, if product deleted or out of stock:
 - System displays warning message
 - System highlights unavailable products
 - System provides option to remove from cart
 - System updates total excluding unavailable items

AF3: Price Changed

- At step 3, if product price changed:
 - System displays notification "Price updated"
 - System shows old and new price
 - Cart total updated automatically

AF4: Stock Insufficient

- At step 3, if requested quantity > available stock:
 - System displays warning "Only X available"
 - System suggests reducing quantity
 - "Checkout" button may be disabled until fixed

Exception Flows:

EF1: Cart Loading Error

- System fails to retrieve cart
- System displays error message
- System offers retry option

Business Rules:

- BR028: Cart validates stock availability on view
- BR029: Prices updated in real-time
- BR030: Removed items can't be recovered
- BR031: Cart persists across sessions for logged-in users

Use Case UC09: Update Cart Quantity

| Element | Description |
|------------------|------------------------------------------------------------|
| Use Case ID | UC09 |
| Use Case Name | Update Cart Quantity |
| Actor(s) | Customer (Primary) |
| Description | Allows customer to change quantity of items in cart |
| Preconditions | - Customer is viewing cart - Cart has at least one item |
| Postconditions | - Cart quantity updated - Total price recalculated |
| Priority | High |
| Frequency of Use | Medium |

Main Flow:

1. Customer is on cart page (UC08)
2. Customer changes quantity using:
 - Plus/Minus buttons
 - Direct input in quantity field
3. System validates new quantity:
 - Quantity > 0
 - Quantity ≤ available stock
 - Quantity ≤ maximum limit (10)
4. System updates cart item quantity in real-time
5. System recalculates:
 - Item subtotal
 - Cart total
6. System updates backend/localStorage
7. System displays updated totals

Alternative Flows:

AF1: Reduce to Zero

- At step 2, if customer sets quantity to 0:
 - System prompts "Remove this item?"
 - If confirmed: removes item (UC10)
 - If cancelled: reverts to previous quantity

AF2: Exceed Available Stock

- At step 3, if new quantity > available stock:
 - System displays error "Only X available"
 - System sets quantity to maximum available
 - System displays warning message

AF3: Exceed Maximum Limit

- At step 3, if quantity > 10:
 - System displays "Maximum 10 items allowed"
 - System sets quantity to 10
 - Customer can accept or cancel

Exception Flows:

EF1: Update Failed

- System fails to update quantity
- System reverts to previous quantity
- System displays error message
- Customer can retry

Business Rules:

- BR032: Minimum quantity: 1 (or remove item)
- BR033: Maximum quantity: 10 per product
- BR034: Quantity updates trigger immediate total recalculation
- BR035: Stock validated on each update

Use Case UC10: Remove from Cart

| Element | Description |
|-------------------------|------------------------------------------------------------|
| Use Case ID | UC10 |
| Use Case Name | Remove from Cart |
| Actor(s) | Customer (Primary) |
| Description | Allows customer to remove an item from shopping cart |
| Preconditions | - Customer is viewing cart - Cart has at least one item |
| Postconditions | - Item removed from cart - Cart total recalculated |
| Priority | High |
| Frequency of Use | Medium |

Main Flow:

1. Customer is on cart page (UC08)
2. Customer clicks "Remove" button/icon for specific item
3. System displays confirmation dialog:
 - "Remove [Product Name] from cart?"
 - "Cancel" and "Remove" buttons
4. Customer clicks "Remove"
5. System removes item from cart
6. System recalculates cart total
7. System updates cart count in header
8. System displays notification "Item removed"
9. System updates backend/localStorage

Alternative Flows:

AF1: Cancel Removal

- At step 4, customer clicks "Cancel"
- System closes dialog
- Cart remains unchanged

AF2: Last Item Removed

- At step 5, if removing last item in cart:
 - System displays "Cart is now empty"
 - System shows "Start Shopping" button
 - Cart page shows empty state

AF3: Quick Remove Without Confirmation

- System may offer setting to skip confirmation
- If enabled, item removed immediately
- System shows undo notification for 5 seconds

Exception Flows:

EF1: Removal Failed

- System fails to remove item
- System displays error message
- Cart state unchanged
- Customer can retry

Business Rules:

- BR036: Removed items cannot be recovered (customer must re-add)
 - BR037: Removal triggers immediate total recalculation
 - BR038: Empty cart redirects to empty state view
-

Use Case UC11: Checkout

| Element | Description |
|-------------------------|----------------------------------------------------------------------------------------------------------|
| Use Case ID | UC11 |
| Use Case Name | Checkout |
| Actor(s) | Customer (Primary), System (Secondary) |
| Description | Allows customer to complete purchase and place order |
| Preconditions | - Customer is logged in - Cart has at least one item - All cart items are available |
| Postconditions | - Order created in system - Stock reserved/decremented - Order confirmation sent - Cart cleared |
| Priority | Critical |
| Frequency of Use | High |

Main Flow:

1. Customer clicks "Proceed to Checkout" from cart

2. System validates:

- Customer is logged in
- Cart is not empty
- All products still available
- Stock sufficient for all items

3. System displays checkout page with:

- Order summary (items, quantities, prices)
- Shipping information form:
 - Full name
 - Phone number
 - Delivery address (street, city, district)
- Payment method selection (COD - Cash on Delivery)
- Terms and conditions checkbox

4. Customer fills in shipping information

5. Customer reviews order details

6. Customer checks "Accept terms" checkbox

7. Customer clicks "Place Order" button

8. System validates all required fields

9. System creates order:

- Generates unique order ID
- Sets status to "Pending"
- Records order date/time
- Stores shipping information
- Calculates total amount

10. System reserves/decrements stock for ordered items

11. System clears customer's cart

12. System displays order confirmation page:

- Order ID
- Order details
- Estimated delivery time
- "Track Order" button

13. System sends order confirmation email (optional feature)

14. System redirects to order confirmation page

Alternative Flows:

AF1: Customer Not Logged In

- At step 2, if customer not logged in:
 - System redirects to login page
 - System saves cart state
 - After login, system redirects back to checkout
 - Flow continues from step 3

AF2: Product Became Unavailable

- At step 2, if any product out of stock:
 - System displays error message
 - System highlights unavailable products
 - System redirects to cart
 - Customer must remove unavailable items

AF3: Validation Errors

- At step 8, if required fields missing or invalid:
 - System highlights error fields
 - System displays specific error messages
 - Customer corrects errors
 - Flow returns to step 7

AF4: Use Saved Address

- At step 4, if customer has previous orders:
 - System offers to use previous shipping address
 - Customer can select or enter new address

Exception Flows:

EF1: Order Creation Failed

- At step 9, system fails to create order:
 - System displays error message
 - System does not modify cart or stock
 - Customer can retry checkout
 - System logs error for investigation

EF2: Stock Reservation Failed

- At step 10, stock update fails:
 - System cancels order
 - System displays error message
 - Cart remains intact
 - Customer can retry or contact support

EF3: Payment Processing Error (Future: when payment gateway integrated)

- Payment service unavailable
- System handles gracefully
- Order saved as "Payment Pending"

Business Rules:

- BR039: Customer must be logged in to checkout
 - BR040: Order ID format: ORD-YYYYMMDD-XXXXXX
 - BR041: Default payment method: Cash on Delivery (COD)
 - BR042: Stock decremented immediately upon order placement
 - BR043: Orders cannot be edited after placement
 - BR044: Minimum order value: 0 VND (no minimum)
 - BR045: Maximum order value: 100,000,000 VND
-

Use Case UC12: View Order History

| Element | Description |
|------------------|---------------------------------------------------------------------|
| Use Case ID | UC12 |
| Use Case Name | View Order History |
| Actor(s) | Customer (Primary) |
| Description | Allows customer to view all their past and current orders |
| Preconditions | - Customer is logged in - Customer has placed at least one order |
| Postconditions | - List of customer's orders displayed |
| Priority | High |
| Frequency of Use | Medium |

Main Flow:

1. Customer navigates to "My Orders" or "Order History" page
2. System retrieves all orders for logged-in customer
3. System displays orders in reverse chronological order (newest first)
4. For each order, system displays:
 - Order ID
 - Order date
 - Total amount
 - Number of items
 - Current status (Pending, Processing, Shipped, Delivered, Cancelled)
 - Status indicator (color-coded)
 - "View Details" button
5. Customer can click on any order to view details (UC13)
6. System provides filter/sort options:
 - Filter by status
 - Sort by date
 - Search by order ID

Alternative Flows:

AF1: No Orders Yet

- At step 2, if customer has no orders:
 - System displays "No orders yet"
 - System shows "Start Shopping" button
 - System may show popular products

AF2: Filter by Status

- Customer selects status filter
- System displays only orders matching selected status
- Customer can clear filter to see all

AF3: Pagination

- If customer has many orders (> 10):
 - System displays 10 orders per page
 - System provides pagination controls
 - Customer can navigate pages

Exception Flows:

EF1: Loading Error

- System fails to retrieve orders
- System displays error message
- System offers retry option

Business Rules:

- BR046: Orders displayed newest first
- BR047: All order statuses visible to customer
- BR048: 10 orders per page (pagination)
- BR049: Order history retained indefinitely

Use Case UC13: View Order Details

| Element | Description |
|------------------|----------------------------------------------------------------------------------|
| Use Case ID | UC13 |
| Use Case Name | View Order Details |
| Actor(s) | Customer (Primary) |
| Description | Allows customer to view complete details of a specific order |
| Preconditions | - Customer is logged in - Customer owns the order - Order exists in system |
| Postconditions | - Detailed order information displayed |
| Priority | High |
| Frequency of Use | Medium |

Main Flow:

1. Customer clicks on specific order from order history (UC12)

2. System validates:

- Order belongs to logged-in customer
- Order exists

3. System retrieves complete order details

4. System displays order details page with:

- **Order Information:**

- Order ID
- Order date and time
- Current status with timeline
- Status history (if available)

- **Items Ordered:**

- Product image (thumbnail)
- Product name
- Quantity
- Unit price at time of order
- Subtotal

- **Shipping Information:**

- Recipient name
- Phone number
- Delivery address

- **Payment Information:**

- Payment method (COD)
- Total amount

- **Actions:**

- "Track Order" button (UC14)
- "Contact Support" button (future)
- "Cancel Order" button (if status allows)

5. Customer can track current status

Alternative Flows:

AF1: Order Not Found

- At step 2, if order doesn't exist:
 - System displays "Order not found" error
 - System redirects to order history

AF2: Unauthorized Access

- At step 2, if order belongs to different customer:
 - System displays "Access denied" error
 - System redirects to order history
 - System logs suspicious activity

AF3: Cancel Order Request

- If order status is "Pending":
 - Customer can click "Cancel Order"
 - System prompts confirmation
 - If confirmed, system updates status to "Cancelled"
 - System restores stock
 - System displays confirmation message

Exception Flows:

EF1: Loading Error

- System fails to retrieve order details
- System displays error message
- System offers retry or return to history

Business Rules:

- BR050: Customer can only view their own orders
- BR051: Order details immutable (cannot be edited)
- BR052: Orders can be cancelled only if status is "Pending"
- BR053: Cancelled orders remain in history
- BR054: All prices shown as they were at time of order

Use Case UC14: Track Order Status

| Element | Description |
|------------------|------------------------------------------------------------------|
| Use Case ID | UC14 |
| Use Case Name | Track Order Status |
| Actor(s) | Customer (Primary) |
| Description | Allows customer to see current status and history of their order |
| Preconditions | - Customer is logged in - Customer has placed an order |
| Postconditions | - Order status timeline displayed |
| Priority | Medium |
| Frequency of Use | Medium |

Main Flow:

1. Customer navigates to order tracking (from UC12 or UC13)
2. System retrieves order status information
3. System displays status timeline with:
 - Order Placed ✓ (date/time)
 - Processing (date/time or "In Progress")
 - Shipped (date/time or "Not Yet")
 - Out for Delivery (date/time or "Not Yet")
 - Delivered (date/time or "Not Yet")
4. System highlights current status
5. System displays estimated delivery date (if available)
6. System shows status description:
 - What happens at this stage
 - Expected next step
 - Estimated time to next stage

Alternative Flows:

AF1: Order Cancelled

- If order status is "Cancelled":
 - System displays cancelled status
 - System shows cancellation reason
 - System shows cancellation date
 - Timeline stops at cancellation point

AF2: Delivery Completed

- If order delivered:
 - System shows complete timeline
 - System displays actual delivery date
 - System may request feedback/review

AF3: Delayed Order

- If order delayed beyond estimated delivery:
 - System displays delay notification
 - System shows new estimated date
 - System may provide reason

Exception Flows:

EF1: Status Update Failed

- System cannot retrieve current status
- System displays last known status
- System shows "Unable to update" message
- Customer can retry

Business Rules:

- BR055: Status updates in real-time (when admin updates)
 - BR056: Estimated delivery: 3-7 business days from order date
 - BR057: Status timeline preserved in order history
 - BR058: Customer receives notifications on major status changes (optional)
-

Use Case UC15: Update Profile

| Element | Description |
|-------------------------|-----------------------------------------------------|
| Use Case ID | UC15 |
| Use Case Name | Update Profile |
| Actor(s) | Customer (Primary) |
| Description | Allows customer to update their account information |
| Preconditions | - Customer is logged in |
| Postconditions | - Profile information updated in database |
| Priority | Low |
| Frequency of Use | Low |

Main Flow:

1. Customer navigates to profile/account settings page
2. System displays current profile information:
 - Full name
 - Email (read-only)
 - Phone number
 - Default shipping address (optional)
3. Customer modifies editable fields
4. Customer clicks "Save Changes" button
5. System validates input:
 - Required fields filled
 - Phone format valid
 - Address format valid (if provided)
6. System updates profile in database
7. System displays success message
8. System refreshes profile page with updated information

Alternative Flows:

AF1: Change Password

- Customer clicks "Change Password" link
- System displays password change form:
 - Current password
 - New password
 - Confirm new password
- Customer fills form
- System validates:
 - Current password correct
 - New password meets requirements
 - Passwords match
- System updates password (hashed)
- System displays success message
- System may logout user and require re-login

AF2: Validation Errors

- At step 5, if validation fails:
 - System highlights error fields
 - System displays error messages
 - Customer corrects errors
 - Flow returns to step 3

AF3: No Changes Made

- If customer submits without changes:
 - System displays "No changes to save"
 - Profile page remains as is

Exception Flows:

EF1: Update Failed

- System fails to update database
- System displays error message
- Changes not saved
- Customer can retry

Business Rules:

- BR059: Email cannot be changed (unique identifier)
- BR060: Phone number optional but recommended
- BR061: Password changes require current password verification
- BR062: Profile updates logged for security

Use Case UC16: Logout

| Element | Description |
|-------------------------|-------------------------------------------------------------------------------------------|
| Use Case ID | UC16 |
| Use Case Name | Logout |
| Actor(s) | Customer (Primary) |
| Description | Allows customer to end their session and logout |
| Preconditions | - Customer is logged in |
| Postconditions | - Customer logged out - Session token invalidated - Customer redirected to homepage |
| Priority | Medium |
| Frequency of Use | High |

Main Flow:

1. Customer clicks "Logout" button/link
2. System displays confirmation (optional):
 - "Are you sure you want to logout?"
3. Customer confirms logout
4. System invalidates JWT token
5. System clears authentication state:
 - Removes token from localStorage/cookies
 - Clears user session data
6. System redirects to homepage
7. System displays message "Logged out successfully"
8. Navigation menu updated to show Login/Register

Alternative Flows:

AF1: Skip Confirmation

- System configured to logout immediately
- Steps 2-3 skipped
- Flow continues from step 4

AF2: Save Cart for Guest

- If customer had items in cart:
 - System may prompt to save cart
 - If logged out as guest, cart saved to localStorage
 - Cart available if customer logs back in

Exception Flows:

EF1: Logout Failed

- System fails to invalidate session
- System displays error
- Customer can retry
- For security, force logout client-side anyway

Business Rules:

- BR063: Logout invalidates JWT token
 - BR064: Cart preserved for logged-in users (saved to DB)
 - BR065: Guest cart lost on logout/browser close
 - BR066: No confirmation required for logout (optional)
-

Use Case UC17: Admin Login

| Element | Description |
|------------------|--------------------------------------------------------------------------------------------------|
| Use Case ID | UC17 |
| Use Case Name | Admin Login |
| Actor(s) | Administrator (Primary) |
| Description | Allows administrator to authenticate and access admin dashboard |
| Preconditions | - Administrator has valid admin credentials - Administrator is not currently logged in |
| Postconditions | - Administrator authenticated - Admin JWT token issued - Access to admin dashboard granted |
| Priority | Critical |
| Frequency of Use | High (Daily for admins) |

Main Flow:

1. Administrator navigates to admin login page (/admin/login)
2. System displays admin login form
3. Administrator enters credentials:
 - Email address
 - Password
4. Administrator clicks "Login" button
5. System validates credentials:
 - Checks email exists
 - Verifies password matches
 - Verifies user has admin role
6. System generates JWT token with admin privileges
7. System stores token
8. System redirects to admin dashboard (UC18)
9. System logs admin login activity

Alternative Flows:

AF1: Invalid Credentials or Not Admin

- At step 5, if:
 - Email not found, OR
 - Password incorrect, OR
 - User is not admin (role = "user")
- System displays "Invalid admin credentials"
- Login attempt logged for security
- Flow returns to step 3

AF2: Account Locked

- At step 5, if account locked:
 - System displays "Account locked" message
 - Admin must contact super admin
 - Cannot proceed

Exception Flows:

EF1: Authentication Service Error

- System cannot connect to auth service
- System displays error message
- Administrator can retry

Business Rules:

- BR067: Only users with role="admin" can access admin panel
- BR068: Admin logins logged with timestamp and IP
- BR069: Admin account locks after 5 failed attempts
- BR070: Admin sessions timeout after 8 hours of inactivity

Use Case UC18: View Dashboard

| Element | Description |
|------------------|------------------------------------------------------------------------|
| Use Case ID | UC18 |
| Use Case Name | View Dashboard |
| Actor(s) | Administrator (Primary) |
| Description | Allows administrator to view overview of system status and key metrics |
| Preconditions | - Administrator is logged in |
| Postconditions | - Dashboard with KPIs displayed |
| Priority | High |
| Frequency of Use | Very High |

Main Flow:

1. Administrator accesses admin dashboard (after login)

2. System retrieves dashboard data:

- Total orders (today, this week, this month)
- Total revenue (today, this week, this month)
- Total products
- Low stock alerts
- Total customers
- Recent orders
- Pending orders count

3. System displays dashboard with widgets:

- **KPI Cards:**

- Total Orders (with percentage change)
- Total Revenue (with percentage change)
- Total Products
- Total Customers

- **Charts:**

- Revenue chart (line/bar chart by day/month)
- Orders chart (line/bar chart by day/month)
- Top selling products (bar chart)

- **Tables:**

- Recent orders (last 10)
- Low stock products
- Pending orders requiring attention

4. Each widget provides quick actions:

- Click on order to view details
- Click on product to edit
- Click on chart to view detailed analytics

Alternative Flows:

AF1: No Data Available

- If system is new with no orders:
 - System displays "No orders yet"
 - System shows welcome message
 - System provides quick start guide

AF2: Filter Dashboard by Date Range

- Administrator selects date range filter
- System updates all metrics for selected range
- Charts and numbers refresh accordingly

Exception Flows:

EF1: Data Loading Error

- System fails to retrieve dashboard data
- System displays error for affected widgets
- System shows cached data if available
- Administrator can refresh

Business Rules:

- BR071: Dashboard data updates in real-time
- BR072: Low stock threshold: 5 units
- BR073: Recent orders: last 10 orders
- BR074: Default time range: current month

Use Case UC19: Create Product

| Element | Description |
|-------------------------|----------------------------------------------------------------------------------|
| Use Case ID | UC19 |
| Use Case Name | Create Product |
| Actor(s) | Administrator (Primary) |
| Description | Allows administrator to add a new product to the catalog |
| Preconditions | - Administrator is logged in - Administrator has access to product management |
| Postconditions | - New product added to database - Product visible to customers |
| Priority | High |
| Frequency of Use | Medium |

Main Flow:

1. Administrator navigates to "Add Product" page
2. System displays product creation form with fields:
 - **Basic Information:**
 - Product name (required)
 - Category (dropdown: Phone / Laptop) (required)
 - Brand (required)
 - Price in VND (required)
 - Stock quantity (required, default: 0)
 - **Detailed Information:**
 - Description (rich text, optional)
 - Specifications (JSON/structured fields):
 - Screen size
 - Processor
 - RAM
 - Storage
 - Additional specs based on category
 - **Images:**
 - Product image URL (required)
 - Additional images (optional, future feature)
3. Administrator fills in all required fields
4. Administrator clicks "Create Product" button
5. System validates input:
 - All required fields filled
 - Price > 0
 - Stock ≥ 0
 - Image URL valid format
 - Specifications valid
6. System creates product in database
7. System generates unique product ID
8. System displays success message
9. System redirects to product list or created product details
10. Product immediately visible to customers (if stock > 0)

Alternative Flows:

AF1: Validation Errors

- At step 5, if validation fails:
 - System highlights error fields with messages
 - Administrator corrects errors
 - Flow returns to step 3

AF2: Save as Draft

- Administrator clicks "Save as Draft"
 - System saves product with status="draft"
 - Product not visible to customers
 - Administrator can edit and publish later

AF3: Duplicate Product

- Administrator chooses to duplicate existing product
- System pre-fills form with existing product data
- Administrator modifies as needed
- Flow continues from step 3

Exception Flows:

EF1: Creation Failed

- System fails to create product
- System displays error message
- Data not saved
- Administrator can retry

Business Rules:

- BR075: Product name must be unique
- BR076: Price must be positive number
- BR077: Stock cannot be negative
- BR078: Category must be "Phone" or "Laptop"
- BR079: Product ID auto-generated (PROD-XXXXX format)
- BR080: Products with stock=0 shown as "Out of Stock"

Use Case UC20: Edit Product

| Element | Description |
|------------------|--------------------------------------------------------------------------------|
| Use Case ID | UC20 |
| Use Case Name | Edit Product |
| Actor(s) | Administrator (Primary) |
| Description | Allows administrator to modify existing product information |
| Preconditions | - Administrator is logged in - Product exists in database |
| Postconditions | - Product information updated - Changes reflected immediately for customers |
| Priority | High |
| Frequency of Use | High |

Main Flow:

1. Administrator selects product to edit from product list
2. System retrieves product details
3. System displays edit product form pre-filled with current data
4. Administrator modifies desired fields:
 - Name, category, brand
 - Price
 - Stock quantity
 - Description
 - Specifications
 - Image URL
5. Administrator clicks "Save Changes" button
6. System validates modifications
7. System updates product in database
8. System logs change history (what changed, by whom, when)
9. System displays success message
10. System redirects to product details or product list
11. Changes immediately visible to customers

Alternative Flows:

AF1: No Changes Made

- If administrator submits without changes:
 - System displays "No changes to save"
 - Product remains unchanged

AF2: Price Change

- If price changed:
 - System may flag for review
 - System logs old and new price
 - Active cart items get new price on next cart view

AF3: Stock Update

- If stock changed:
 - System validates stock adjustments
 - If stock becomes 0, product shown as "Out of Stock"
 - If stock increases, product available again
 - System may notify customers who favorited (future)

AF4: Category Change (Restricted)

- System may restrict category changes
- If allowed:
 - System warns about specification mismatch
 - Administrator must confirm change

Exception Flows:

EF1: Update Failed

- System fails to update product
- System displays error message
- Changes not saved
- Administrator can retry

EF2: Concurrent Edit Conflict

- Another admin edited same product simultaneously
- System displays conflict warning
- System shows both versions
- Administrator must resolve conflict

Business Rules:

- BR081: All edits logged with admin ID and timestamp
 - BR082: Price changes effective immediately
 - BR083: Stock changes effective immediately
 - BR084: Cannot edit to negative stock
 - BR085: Category changes restricted or warned
-

Use Case UC21: Delete Product

| Element | Description |
|-------------------------|-----------------------------------------------------------------------------|
| Use Case ID | UC21 |
| Use Case Name | Delete Product |
| Actor(s) | Administrator (Primary) |
| Description | Allows administrator to remove a product from the catalog |
| Preconditions | - Administrator is logged in - Product exists in database |
| Postconditions | - Product deleted from database - Product no longer visible to customers |
| Priority | Medium |
| Frequency of Use | Low |

Main Flow:

1. Administrator selects product to delete from product list
2. Administrator clicks "Delete" button
3. System displays confirmation dialog:
 - "Are you sure you want to delete [Product Name]?"
 - Warning: "This action cannot be undone"
 - Product details summary
 - "Cancel" and "Delete" buttons
4. Administrator clicks "Delete" to confirm
5. System validates deletion:
 - Checks if product in any pending orders
 - Checks if product in any carts
6. System soft-deletes product:
 - Marks as deleted (status="deleted")
 - Product hidden from customer view
 - Product data preserved for order history
7. System logs deletion (admin ID, timestamp, product info)
8. System displays success message "Product deleted"
9. System refreshes product list (deleted product removed)

Alternative Flows:

AF1: Cancel Deletion

- At step 4, administrator clicks "Cancel"
- System closes dialog
- Product remains unchanged

AF2: Product in Active Orders

- At step 5, if product in pending/processing orders:
 - System displays warning
 - System lists affected orders
 - Administrator must:
 - Cancel orders first, OR
 - Wait for orders to complete, OR
 - Proceed anyway (soft delete, keep for history)

AF3: Product in Carts

- At step 5, if product in user carts:
 - System displays warning "X users have this in cart"
 - System offers to:
 - Proceed with deletion (removes from all carts)
 - Cancel deletion
 - If proceeding, system removes from all carts

AF4: Soft Delete vs Hard Delete

- Default: Soft delete (set deleted flag)
- Option for hard delete (permanently remove):
 - Only if no order history
 - Requires super admin permission (future)

Exception Flows:

EF1: Deletion Failed

- System fails to delete product
- System displays error message
- Product remains in database
- Administrator can retry

Business Rules:

- BR086: Products with order history cannot be hard-deleted
- BR087: Soft-deleted products remain in database for history
- BR088: Deletion removes product from all user carts
- BR089: Deletion logged with admin details
- BR090: Deleted products not visible in customer searches

Use Case UC22: Manage Inventory

| Element | Description |
|------------------|------------------------------------------------------------------|
| Use Case ID | UC22 |
| Use Case Name | Manage Inventory |
| Actor(s) | Administrator (Primary) |
| Description | Allows administrator to view and manage product inventory levels |
| Preconditions | - Administrator is logged in |
| Postconditions | - Inventory levels updated - Low stock alerts addressed |
| Priority | High |
| Frequency of Use | High (Daily) |

Main Flow:

1. Administrator navigates to "Inventory Management" page

2. System displays inventory overview:

- **Summary Cards:**

- Total products
- Out of stock count
- Low stock count (< 5 units)
- Total inventory value

- **Product List with:**

- Product name
- Category
- Current stock
- Stock status (In Stock / Low Stock / Out of Stock)
- Last updated date
- Quick actions (Edit, Restock)

3. Administrator can:

- Sort by stock level, name, category
- Filter by stock status
- Search products

4. Administrator identifies low-stock products

5. Administrator clicks "Restock" for specific product

6. System displays restock dialog:

- Current stock: X
- Add quantity: [input field]
- New total: X + input

7. Administrator enters quantity to add

8. Administrator clicks "Update Stock"

9. System updates stock level

10. System logs inventory change

11. System displays updated inventory list

Alternative Flows:

AF1: Bulk Stock Update

- Administrator selects multiple products
- Administrator clicks "Bulk Update Stock"
- System displays bulk update form
- Administrator enters new quantities
- System updates all selected products
- System logs all changes

AF2: Set Stock Alert Threshold

- Administrator clicks "Stock Alert Settings"
- System displays threshold setting
- Administrator sets custom low-stock threshold
- System updates alert logic
- Future alerts based on new threshold

AF3: Export Inventory Report

- Administrator clicks "Export Inventory"
- System generates Excel/CSV file with:
 - All products
 - Stock levels
 - Values
 - Last update dates
- System provides download link

AF4: View Stock History

- Administrator clicks "View History" for product
- System displays stock change log:
 - Date/time
 - Old quantity
 - New quantity
 - Change reason (sale, restock, adjustment)
 - Admin who made change

Exception Flows:

EF1: Update Failed

- System fails to update stock
- System displays error message
- Stock unchanged
- Administrator can retry

Business Rules:

- BR091: Low stock threshold: 5 units (configurable)
 - BR092: Stock cannot be negative
 - BR093: All stock changes logged with admin ID
 - BR094: Inventory value = sum(stock × price) for all products
 - BR095: Stock updates trigger availability status updates
-

Use Case UC23: View All Orders

| Element | Description |
|------------------|-------------------------------------------------------------|
| Use Case ID | UC23 |
| Use Case Name | View All Orders |
| Actor(s) | Administrator (Primary) |
| Description | Allows administrator to view and manage all customer orders |
| Preconditions | - Administrator is logged in |
| Postconditions | - Complete order list displayed with filters applied |
| Priority | Critical |
| Frequency of Use | Very High (Multiple times daily) |

Main Flow:

1. Administrator navigates to "Orders" page
2. System retrieves all orders from database
3. System displays orders list with:
 - **For each order:**
 - Order ID
 - Order date/time
 - Customer name
 - Total amount
 - Number of items
 - Current status (badge with color coding)
 - Actions (View, Update Status)
 - **Filter options:**
 - Status (All, Pending, Processing, Shipped, Delivered, Cancelled)
 - Date range
 - Customer search
 - Amount range
 - **Sort options:**
 - Date (newest/oldest first)
 - Amount (high/low)
 - Status
4. System defaults to showing:
 - All statuses
 - Sorted by date (newest first)
 - 20 orders per page
5. Administrator can:
 - View order details (UC24)
 - Update order status (UC24)
 - Search/filter/sort orders
 - Export order list

Alternative Flows:

AF1: Filter by Status

- Administrator selects status filter (e.g., "Pending")
- System displays only orders with selected status
- Order count updated
- Administrator can clear filter

AF2: Search by Customer or Order ID

- Administrator enters search term
- System searches in:
 - Order ID
 - Customer name
 - Customer email
- System displays matching orders

AF3: Date Range Filter

- Administrator selects date range
- System displays orders within range
- Summary statistics updated

AF4: Export Orders

- Administrator clicks "Export"
- System generates Excel/CSV with:
 - All filtered orders
 - Order details
 - Customer information
- System provides download link

AF5: Quick Status Update

- Administrator can update status directly from list
- System displays status dropdown for each order
- On change, system updates immediately
- No need to open detailed view

Exception Flows:

EF1: Loading Error

- System fails to retrieve orders
- System displays error message
- Administrator can retry

Business Rules:

- BR096: Default sort: newest orders first
 - BR097: Default view: all statuses
 - BR098: Pagination: 20 orders per page
 - BR099: All admins see all orders (no customer privacy from admins)
 - BR100: Export limited to 1000 orders at a time
-

Use Case UC24: Update Order Status

| Element | Description |
|-------------------------|-------------------------------------------------------------------------------------|
| Use Case ID | UC24 |
| Use Case Name | Update Order Status |
| Actor(s) | Administrator (Primary), System (Secondary) |
| Description | Allows administrator to change the status of customer orders |
| Preconditions | - Administrator is logged in - Order exists in system |
| Postconditions | - Order status updated - Status history logged - Customer notified (optional) |
| Priority | Critical |
| Frequency of Use | Very High |

Main Flow:

1. Administrator views order details (from UC23)
2. System displays current order status and timeline
3. Administrator clicks "Update Status" button
4. System displays status update form:
 - Current status: [Current Status]
 - New status dropdown:
 - Pending
 - Processing
 - Shipped
 - Delivered
 - Cancelled
 - Notes/comments (optional)
 - Notification checkbox (Send email to customer)
5. Administrator selects new status
6. Administrator optionally adds notes
7. Administrator clicks "Update" button
8. System validates status transition:
 - Checks if transition is logical
 - (e.g., cannot go from Delivered to Pending)
9. System updates order status
10. System logs status change:
 - Old status → New status
 - Timestamp
 - Admin ID
 - Notes
11. System sends notification to customer (if enabled)
12. System displays success message
13. System refreshes order details page

Alternative Flows:

AF1: Cancel Order

- Administrator selects "Cancelled" status
- System prompts for cancellation reason
- System asks about stock restoration:
 - "Restore stock for cancelled items?"
- If confirmed:
 - System restores stock for all order items
 - Stock increases by order quantities
- Order marked as cancelled
- Customer notified

AF2: Invalid Status Transition

- At step 8, if transition not allowed:
 - System displays error message
 - System explains valid transitions
 - Administrator must select valid status

AF3: Bulk Status Update

- Administrator selects multiple orders (from UC23)
- Administrator clicks "Bulk Update Status"
- System displays bulk update dialog
- Administrator selects new status for all
- System updates all selected orders
- System logs all changes

AF4: Add Tracking Information (Future Feature)

- If status changed to "Shipped":
 - System prompts for tracking number
 - System prompts for carrier
 - System stores tracking information
 - Customer receives tracking link

Exception Flows:

EF1: Update Failed

- System fails to update status
- System displays error message
- Status unchanged
- Administrator can retry

EF2: Notification Failed

- Status updated successfully
- Email notification fails
- System logs notification error
- System displays warning
- Administrator can manually notify customer

Business Rules:

- BR101: Valid status transitions:
 - Pending → Processing, Cancelled
 - Processing → Shipped, Cancelled
 - Shipped → Delivered
 - Delivered → (final state, no transitions)
 - Cancelled → (final state, no transitions)
- BR102: Cancelling order restores stock
- BR103: All status changes logged
- BR104: Customer notified on major status changes
- BR105: Status history immutable (cannot delete)

Use Case UC25: View Analytics

| Element | Description |
|------------------|--------------------------------------------------------------|
| Use Case ID | UC25 |
| Use Case Name | View Analytics |
| Actor(s) | Administrator (Primary), Business Owner (Secondary) |
| Description | Allows administrator to view business analytics and reports |
| Preconditions | - Administrator is logged in - System has historical data |
| Postconditions | - Analytics and insights displayed |
| Priority | Medium |
| Frequency of Use | Medium (Weekly/Monthly) |

Main Flow:

1. Administrator navigates to "Analytics" or "Reports" page
2. System displays analytics dashboard with:
 - **Time Period Selector:**
 - Today, Last 7 days, Last 30 days, This Month, Last Month, This Year, Custom Range
 - **Key Metrics:**
 - Total Revenue (with trend)
 - Total Orders (with trend)
 - Average Order Value
 - Conversion Rate
 - New Customers
 - **Charts:**
 - Revenue over time (line chart)
 - Orders over time (line chart)
 - Revenue by category (pie chart)
 - Top selling products (bar chart)
 - Order status distribution (pie chart)
 - **Tables:**
 - Top 10 products by revenue
 - Top 10 products by quantity sold
 - Customer order frequency
3. Administrator selects time period
4. System recalculates metrics for selected period
5. System updates all charts and tables
6. Administrator can:
 - Hover over charts for detailed data
 - Click on data points for drill-down
 - Export reports

Alternative Flows:

AF1: Custom Date Range

- Administrator selects "Custom Range"
- System displays date picker
- Administrator selects start and end dates
- System validates date range
- System updates analytics

AF2: Compare Periods

- Administrator enables "Compare to previous period"
- System displays:
 - Current period metrics
 - Previous period metrics
 - Percentage change
- Charts show both periods overlaid

AF3: Filter by Category

- Administrator selects category filter
- System shows analytics for selected category only
- All metrics and charts filtered accordingly

AF4: Export Analytics Report

- Administrator clicks "Export Report"
- System generates PDF or Excel report with:
 - All current metrics
 - All charts (as images)
 - Selected time period
 - Generation timestamp
- System provides download link

Exception Flows:

EF1: Insufficient Data

- If selected period has no data:
 - System displays "No data available for this period"
 - System suggests different time range

EF2: Calculation Error

- System fails to calculate metrics
- System displays error for affected sections
- Administrator can refresh

Business Rules:

- BR106: Analytics calculated on demand (not pre-computed)
 - BR107: Conversion rate = Orders / Unique Visitors (if tracking available)
 - BR108: Trend indicators show change from previous equivalent period
 - BR109: All currency values in VND
 - BR110: Export reports include generation timestamp
-

Use Case UC26: Generate Reports

| Element | Description |
|-------------------------|------------------------------------------------------------------------|
| Use Case ID | UC26 |
| Use Case Name | Generate Reports |
| Actor(s) | Administrator (Primary), Business Owner (Secondary) |
| Description | Allows administrator to generate and download various business reports |
| Preconditions | - Administrator is logged in - System has data to report on |
| Postconditions | - Report generated and available for download |
| Priority | Low |
| Frequency of Use | Low (Monthly/Quarterly) |

Main Flow:

1. Administrator navigates to "Reports" page
2. System displays report generation form with:
 - **Report Type:**
 - Sales Report
 - Inventory Report
 - Customer Report
 - Order Report
 - Revenue Report
 - **Time Period:**
 - This Month, Last Month, This Quarter, Last Quarter, This Year, Custom Range
 - **Format:**
 - PDF
 - Excel/CSV
 - **Options** (varies by report type):
 - Include charts
 - Include summary
 - Group by category
3. Administrator selects report parameters
4. Administrator clicks "Generate Report" button
5. System validates parameters
6. System collects data based on selections
7. System generates report file:
 - Formats data appropriately
 - Includes selected visualizations
 - Adds header/footer with timestamp
8. System displays download link
9. Administrator downloads report

Alternative Flows:

AF1: Schedule Recurring Report (Future Feature)

- Administrator enables "Schedule Report"
- System displays scheduling options:
 - Frequency (Daily, Weekly, Monthly)
 - Recipients (email list)
 - Time to send
- System saves schedule
- System generates and emails report automatically

AF2: Sales Report Details

- If "Sales Report" selected:
 - System includes:
 - Total revenue
 - Revenue by category
 - Revenue by product
 - Payment method breakdown
 - Discounts applied (if any)
 - Top selling products

AF3: Inventory Report Details

- If "Inventory Report" selected:
 - System includes:
 - Current stock levels
 - Stock value
 - Low stock items
 - Out of stock items
 - Stock movement (in/out)
 - Restock recommendations

AF4: Customer Report Details

- If "Customer Report" selected:
 - System includes:
 - Total customers
 - New customers in period
 - Customer order frequency
 - Top customers by spending
 - Customer retention metrics

Exception Flows:

EF1: Generation Failed

- System fails to generate report
- System displays error message
- Administrator can retry

EF2: Large Report Timeout

- Report generation takes too long
- System switches to background processing
- System displays "Report will be ready soon"
- System sends email when ready

Business Rules:

- BR111: Reports include generation timestamp
 - BR112: All financial values in VND
 - BR113: Maximum report period: 1 year
 - BR114: Reports stored temporarily (7 days)
 - BR115: Only administrators can generate reports
-

Use Case UC27: Manage Users

| Element | Description |
|------------------|-------------------------------------------------------|
| Use Case ID | UC27 |
| Use Case Name | Manage Users |
| Actor(s) | Administrator (Primary) |
| Description | Allows administrator to view and manage user accounts |
| Preconditions | - Administrator is logged in |
| Postconditions | - User list displayed with management options |
| Priority | Low |
| Frequency of Use | Low |

Main Flow:

1. Administrator navigates to "Users" page
2. System retrieves all user accounts
3. System displays users list with:
 - User ID
 - Name
 - Email
 - Registration date
 - Role (User / Admin)
 - Account status (Active / Suspended)
 - Last login date
 - Total orders
 - Total spent
 - Actions (View, Edit, Suspend/Activate)
4. Administrator can:
 - Search users by name/email
 - Filter by role or status
 - Sort by various fields
 - View user details
 - Edit user information
 - Suspend/activate accounts

Alternative Flows:

AF1: View User Details

- Administrator clicks on user
- System displays user profile with:
 - Account information
 - Order history
 - Total spent
 - Account activity log

AF2: Suspend User Account

- Administrator clicks "Suspend" for user
- System displays confirmation dialog
- If confirmed:
 - System sets account status to "Suspended"
 - User cannot login
 - System displays success message

AF3: Promote to Admin

- Administrator clicks "Promote to Admin"
- System displays confirmation dialog
- If confirmed:
 - System changes user role to "admin"
 - User gains admin access
 - System logs role change

AF4: Reset User Password (Future Feature)

- Administrator clicks "Reset Password"
- System generates temporary password
- System sends password reset email to user
- System displays confirmation

Exception Flows:

EF1: Cannot Suspend Self

- If administrator tries to suspend own account:
 - System displays error
 - "Cannot suspend your own account"
 - Action cancelled

Business Rules:

- BR116: Only admins can manage users
 - BR117: Suspended users cannot login
 - BR118: Admin role changes logged
 - BR119: Cannot delete users with order history
 - BR120: Super admin role required for admin management (future)
-

Use Case UC28: View System Logs

| Element | Description |
|-------------------------|--------------------------------------------------------------|
| Use Case ID | UC28 |
| Use Case Name | View System Logs |
| Actor(s) | Administrator (Primary) |
| Description | Allows administrator to view system activity and audit logs |
| Preconditions | - Administrator is logged in - System has logging enabled |
| Postconditions | - System logs displayed |
| Priority | Low |
| Frequency of Use | Low (As needed for debugging/auditing) |

Main Flow:

1. Administrator navigates to "System Logs" page

2. System retrieves recent log entries

3. System displays logs with:

- **For each log entry:**

- Timestamp
- Log level (Info, Warning, Error)
- Category (Auth, Product, Order, System)
- User/Admin ID (if applicable)
- Action performed
- Details/Message

- **Filter options:**

- Log level
- Category
- Date range
- User/Admin
- Search by keyword

4. System defaults to showing:

- Last 24 hours
- All log levels
- All categories
- 100 entries per page

5. Administrator can:

- Filter and search logs
- View detailed log entry
- Export logs

Alternative Flows:

AF1: Filter by Log Level

- Administrator selects log level (e.g., "Error")
- System displays only logs of that level
- Helpful for troubleshooting

AF2: View Detailed Log Entry

- Administrator clicks on log entry
- System displays expanded view with:
 - Full message
 - Stack trace (if error)
 - Request details
 - User context
 - Related logs

AF3: Export Logs

- Administrator clicks "Export Logs"
- System generates text or CSV file
- System applies current filters to export
- Administrator downloads file

Exception Flows:

EF1: Logs Loading Error

- System fails to retrieve logs
- System displays error message
- Administrator can retry

Business Rules:

- BR121: Logs retained for 90 days
- BR122: Sensitive data (passwords) never logged
- BR123: All admin actions logged
- BR124: Logs immutable (cannot edit/delete)
- BR125: Super admin required to export logs

3.2.3. Use Case Prioritization Matrix

| Priority | Use Cases | Rationale |
|--------------------------------|------------------------------------------------------|----------------------------------------------------------------------------------|
| Critical (Must Have) | UC02, UC03, UC06, UC07, UC11, UC17, UC18, UC23, UC24 | Core functionality required for MVP. System cannot function without these. |
| High (Should Have) | UC01, UC04, UC08, UC09, UC12, UC13, UC19, UC20, UC22 | Important features that significantly enhance usability and functionality. |
| Medium (Could Have) | UC05, UC10, UC14, UC15, UC21, UC25 | Nice-to-have features that improve user experience but not essential for launch. |
| Low (Won't Have in MVP) | UC16, UC26, UC27, UC28 | Features that can be added post-launch or are primarily for convenience. |

3.3. Functional Requirements

This section lists all functional requirements for the Microshop E-Commerce Platform, organized by module.

3.3.1. User Management Module (FR001-FR012)

| Requirement ID | Requirement Description | Priority | Related Use Case |
|----------------|-------------------------------------------------------------------------------------------------|----------|------------------|
| FR001 | The system shall allow new users to register with name, email, and password | Critical | UC01 |
| FR002 | The system shall validate email format and uniqueness during registration | Critical | UC01 |
| FR003 | The system shall enforce minimum password length of 6 characters | Critical | UC01 |
| FR004 | The system shall hash passwords using bcrypt before storage | Critical | UC01, UC02 |
| FR005 | The system shall allow registered users to login with email and password | Critical | UC02 |
| FR006 | The system shall generate JWT tokens upon successful authentication | Critical | UC02 |
| FR007 | The system shall lock user accounts after 5 consecutive failed login attempts | High | UC02 |
| FR008 | The system shall allow users to update their profile information | Medium | UC15 |
| FR009 | The system shall allow users to change their password with current password verification | Medium | UC15 |
| FR010 | The system shall allow users to logout and invalidate their session token | Medium | UC16 |
| FR011 | The system shall differentiate between regular users and administrators using role-based access | Critical | UC02, UC17 |
| FR012 | The system shall maintain user session for 24 hours (or 30 days with "Remember Me") | Medium | UC02 |

3.3.2. Product Management Module (FR013-FR030)

| Requirement ID | Requirement Description | Priority | Related Use Case |
|----------------|-------------------------------------------------------------------------------------------------------|----------|------------------|
| FR013 | The system shall allow customers to browse products with pagination (12 per page) | Critical | UC03 |
| FR014 | The system shall display product name, price, category, image, and stock status | Critical | UC03, UC06 |
| FR015 | The system shall allow customers to search products by name, description, and specifications | High | UC04 |
| FR016 | The system shall provide autocomplete suggestions during product search | Medium | UC04 |
| FR017 | The system shall allow customers to filter products by category, brand, price range, and stock status | High | UC05 |
| FR018 | The system shall allow customers to sort products by price, name, and date added | High | UC03 |
| FR019 | The system shall display detailed product information including specifications and images | Critical | UC06 |
| FR020 | The system shall track product views for analytics purposes | Medium | UC06 |
| FR021 | The system shall allow administrators to create new products with all required information | Critical | UC19 |
| FR022 | The system shall validate product data (price > 0, stock >= 0, required fields) | Critical | UC19 |
| FR023 | The system shall generate unique product IDs automatically | Critical | UC19 |
| FR024 | The system shall allow administrators to edit existing product information | Critical | UC20 |
| FR025 | The system shall log all product changes with timestamp and admin ID | High | UC20 |
| FR026 | The system shall allow administrators to delete products (soft delete) | Medium | UC21 |
| FR027 | The system shall preserve deleted products for order history | Medium | UC21 |
| FR028 | The system shall support two product categories: Phones and Laptops | Critical | All Product UCs |
| FR029 | The system shall mark products as "Out of Stock" when stock quantity is 0 | High | UC03, UC06 |
| FR030 | The system shall mark products as "Low Stock" when quantity is below threshold (5 units) | Medium | UC22 |

3.3.3. Shopping Cart Module (FR031-FR042)

| Requirement ID | Requirement Description | Priority | Related Use Case |
|----------------|----------------------------------------------------------------------------------|----------|------------------|
| FR031 | The system shall allow customers to add products to cart with specified quantity | Critical | UC07 |
| FR032 | The system shall validate stock availability before adding to cart | Critical | UC07 |
| FR033 | The system shall enforce maximum quantity limit of 10 units per product | High | UC07 |
| FR034 | The system shall increase quantity if product already exists in cart | High | UC07 |
| FR035 | The system shall display cart icon with item count in header | High | UC07, UC08 |
| FR036 | The system shall allow customers to view all items in their cart | Critical | UC08 |
| FR037 | The system shall calculate and display subtotals and total price in real-time | Critical | UC08 |
| FR038 | The system shall validate cart items (stock, prices) when viewing cart | High | UC08 |
| FR039 | The system shall allow customers to update item quantities in cart | High | UC09 |
| FR040 | The system shall allow customers to remove items from cart | High | UC10 |
| FR041 | The system shall store cart in database for logged-in users | High | UC07, UC08 |
| FR042 | The system shall store cart in browser localStorage for guest users | High | UC07, UC08 |

3.3.4. Order Management Module (FR043-FR054)

| Requirement ID | Requirement Description | Priority | Related Use Case |
|----------------|--------------------------------------------------------------------------------|----------|------------------|
| FR043 | The system shall require user login before checkout | Critical | UC11 |
| FR044 | The system shall allow customers to enter shipping information during checkout | Critical | UC11 |
| FR045 | The system shall validate all required shipping fields (name, phone, address) | Critical | UC11 |
| FR046 | The system shall generate unique order IDs in format ORD-YYYYMMDD-XXXXX | Critical | UC11 |
| FR047 | The system shall create orders with status "Pending" | Critical | UC11 |
| FR048 | The system shall decrement product stock upon order placement | Critical | UC11 |
| FR049 | The system shall clear customer cart after successful order placement | Critical | UC11 |
| FR050 | The system shall allow customers to view their order history | High | UC12 |
| FR051 | The system shall allow customers to view detailed information for each order | High | UC13 |
| FR052 | The system shall allow customers to track order status with timeline | Medium | UC14 |
| FR053 | The system shall allow customers to cancel orders with status "Pending" | Medium | UC13 |
| FR054 | The system shall restore stock when order is cancelled | Medium | UC13, UC24 |

3.3.5. Admin Dashboard Module (FR055-FR060)

| Requirement ID | Requirement Description | Priority | Related Use Case |
|----------------|--------------------------------------------------------------------------------------|----------|------------------|
| FR055 | The system shall provide admin dashboard with key performance indicators (KPIs) | Critical | UC18 |
| FR056 | The system shall display total orders, revenue, products, and customers on dashboard | Critical | UC18 |
| FR057 | The system shall provide charts for revenue and orders over time | High | UC18, UC25 |
| FR058 | The system shall allow administrators to view all orders with filters and search | Critical | UC23 |
| FR059 | The system shall allow administrators to update order status | Critical | UC24 |
| FR060 | The system shall log all status changes with timestamp and admin ID | High | UC24 |

3.4. Non-Functional Requirements

This section specifies the non-functional requirements that define system qualities and constraints.

3.4.1. Performance Requirements (NFR001-NFR004)

| Requirement ID | Requirement Description | Measurement Criteria | Priority |
|-----------------------|----------------------------------------------------------------------------------------|---------------------------------------------|-----------------|
| NFR001 | The system shall load product listing page within 2 seconds under normal conditions | Page load time \leq 2s | High |
| NFR002 | The system shall process checkout and create order within 3 seconds | Checkout completion time \leq 3s | High |
| NFR003 | The system shall support at least 100 concurrent users without performance degradation | Response time increase < 10% with 100 users | Medium |
| NFR004 | The system shall display search results within 1 second for queries | Search response time \leq 1s | High |

3.4.2. Security Requirements (NFR005-NFR010)

| Requirement ID | Requirement Description | Implementation Method | Priority |
|-----------------------|-------------------------------------------------------------------------------------|------------------------------------|-----------------|
| NFR005 | The system shall store passwords using bcrypt hashing with salt | Bcrypt library with cost factor 10 | Critical |
| NFR006 | The system shall use JWT tokens for authentication with HS256 algorithm | jsonwebtoken library | Critical |
| NFR007 | The system shall enforce HTTPS for all client-server communication | HTTPS/TLS configuration | Critical |
| NFR008 | The system shall validate and sanitize all user inputs to prevent injection attacks | Input validation middleware | Critical |
| NFR009 | The system shall implement CORS policy to restrict API access | CORS middleware configuration | High |
| NFR010 | The system shall log all admin actions for audit purposes | Logging middleware | High |

3.4.3. Usability Requirements (NFR011-NFR014)

| Requirement ID | Requirement Description | Success Criteria | Priority |
|----------------|------------------------------------------------------------------------------------------------|---------------------------------------------|----------|
| NFR011 | The system shall provide a responsive design that works on desktop, tablet, and mobile devices | Works on screens from 320px to 1920px width | Critical |
| NFR012 | The system shall display error messages in Vietnamese language | All error messages in Vietnamese | High |
| NFR013 | The system shall provide visual feedback for all user actions within 0.5 seconds | Loading indicators, success/error messages | High |
| NFR014 | The system shall follow consistent UI/UX patterns throughout the application | Consistent colors, fonts, layouts | Medium |

3.4.4. Reliability Requirements (NFR015-NFR018)

| Requirement ID | Requirement Description | Target Metric | Priority |
|----------------|-------------------------------------------------------------------------------------|-------------------------------------|----------|
| NFR015 | The system shall have an uptime of at least 99% during business hours | Uptime \geq 99% | High |
| NFR016 | The system shall handle database connection failures gracefully with error messages | No application crashes on DB errors | High |
| NFR017 | The system shall implement data validation to prevent corrupt data entry | Validation on all inputs | Critical |
| NFR018 | The system shall backup database daily to prevent data loss | Automated daily backups | High |

3.4.5. Scalability Requirements (NFR019-NFR020)

| Requirement ID | Requirement Description | Implementation Approach | Priority |
|----------------|--------------------------------------------------------------------------------------|--------------------------|----------|
| NFR019 | The system shall use microservices architecture to allow independent service scaling | 3 services + API Gateway | High |
| NFR020 | The system shall use Docker containers for easy deployment and scaling | Docker & Docker Compose | High |

APPENDIX A: REQUIREMENTS TRACEABILITY MATRIX

| Use Case ID | Functional Requirements | Non-Functional Requirements | Priority |
|-------------|-------------------------------------------------|-----------------------------|----------|
| UC01 | FR001, FR002, FR003, FR004 | NFR005, NFR008, NFR011 | Critical |
| UC02 | FR005, FR006, FR007, FR012 | NFR005, NFR006, NFR008 | Critical |
| UC03 | FR013, FR014, FR018 | NFR001, NFR011 | Critical |
| UC04 | FR015, FR016 | NFR004, NFR011 | High |
| UC05 | FR017 | NFR011 | High |
| UC06 | FR019, FR020 | NFR001, NFR011 | Critical |
| UC07 | FR031, FR032, FR033, FR034, FR035 | NFR008, NFR013 | Critical |
| UC08 | FR036, FR037, FR038 | NFR001, NFR011 | Critical |
| UC09 | FR039 | NFR013 | High |
| UC10 | FR040 | NFR013 | High |
| UC11 | FR043, FR044, FR045, FR046, FR047, FR048, FR049 | NFR002, NFR008, NFR017 | Critical |
| UC12 | FR050 | NFR001, NFR011 | High |
| UC13 | FR051, FR053 | NFR011 | High |
| UC14 | FR052 | NFR011 | Medium |
| UC15 | FR008, FR009 | NFR005, NFR008 | Medium |
| UC16 | FR010 | - | Medium |
| UC17 | FR011 | NFR005, NFR006, NFR010 | Critical |
| UC18 | FR055, FR056, FR057 | NFR001 | High |
| UC19 | FR021, FR022, FR023 | NFR008, NFR010, NFR017 | Critical |
| UC20 | FR024, FR025 | NFR008, NFR010 | Critical |
| UC21 | FR026, FR027 | NFR010 | Medium |
| UC22 | FR030 | NFR001, NFR010 | High |
| UC23 | FR058 | NFR001 | Critical |
| UC24 | FR059, FR060, FR054 | NFR010, NFR013 | Critical |
| UC25 | FR057 | NFR004 | Medium |

APPENDIX B: REQUIREMENTS SUMMARY

Summary Statistics

| Category | Count |
|-----------------------------|-----------------------------------|
| Stakeholders | 5 groups |
| Use Cases | 28 total (16 customer, 12 admin) |
| Functional Requirements | 60 requirements |
| Non-Functional Requirements | 20 requirements |
| Business Rules | 125 rules (embedded in use cases) |

Requirements Coverage by Priority

| Priority | Use Cases | Functional Req | Non-Functional Req |
|----------|-----------|----------------|--------------------|
| Critical | 11 | 28 | 8 |
| High | 10 | 22 | 9 |
| Medium | 6 | 8 | 2 |
| Low | 1 | 2 | 1 |

Module Coverage

| Module | Use Cases | Functional Req | % of Total FR |
|--------------------|-----------|----------------|---------------|
| User Management | 4 | 12 | 20% |
| Product Management | 7 | 18 | 30% |
| Shopping Cart | 4 | 12 | 20% |
| Order Management | 6 | 12 | 20% |
| Admin Dashboard | 7 | 6 | 10% |

DOCUMENT VERSION CONTROL

Version: 1.0 **Last Updated:** [TODO: Add date when finalized] **Status:** Complete - Ready for review **Next Steps:**

- Review and validate with team
- Get stakeholder approval
- Proceed to Chapter 4 (Architecture)

End of Chapter 3: Requirement Specifications