1. Use case
2. Scenarios and User stories, acceptance criteria
   1. Scenarios

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC01 |
| Actor | Customer |
| Description | A new customer creates an account on the e-commerce platform |
| Preconditions | Customer is not logged in and does not have an existing account |
| Postconditions | Customer has a new account and is logged in |
| Normal Course | 1. Customer clicks on "Sign Up" button 2. System displays registration form 3. Customer enters required information 4. System validates the information 5. System creates the account and sends a verification email 6. Customer verifies email address 7. System logs the customer in |
| Alternative Courses | - Customer uses social media account to sign up - Customer abandons registration process |
| Exceptions | - Email address is already in use - Password does not meet complexity requirements |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC02 |
| Actor | Customer |
| Description | A customer adds a product to their shopping cart |
| Preconditions | Customer is browsing products |
| Postconditions | Selected item is added to the customer's cart |
| Normal Course | 1. Customer views product details 2. Customer selects quantity (if applicable) 3. Customer clicks "Add to Cart" button 4. System adds item to cart 5. System displays confirmation message 6. System updates cart icon to reflect new item count |
| Alternative Courses | - Customer adds item from product listing page - Customer adds item from wishlist |
| Exceptions | - Item is out of stock - Customer attempts to add more than available quantity |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC03 |
| Actor | Customer |
| Description | A customer completes the checkout process to purchase items in their cart |
| Preconditions | Customer has items in cart and is logged in |
| Postconditions | Order is placed and payment is processed |
| Normal Course | 1. Customer initiates checkout process 2. System displays shipping options 3. Customer selects shipping method 4. System displays payment options 5. Customer enters payment information 6. System validates payment information 7. System displays order summary 8. Customer confirms order 9. System processes payment 10. System generates order confirmation |
| Alternative Courses | - Customer uses saved shipping/payment information - Customer applies discount code |
| Exceptions | - Payment is declined - Item becomes unavailable during checkout |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC04 |
| Actor | Customer |
| Description | A customer uses the search functionality to find specific products |
| Preconditions | Customer is on the e-commerce site |
| Postconditions | Search results are displayed to the customer |
| Normal Course | 1. Customer enters search terms in the search bar 2. System processes the search query 3. System displays search results 4. Customer views search results 5. Customer can apply filters to refine results |
| Alternative Courses | - Customer uses voice search - Customer clicks on suggested search terms |
| Exceptions | - No results found for search query - Search system experiences an error |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC05 |
| Actor | Customer |
| Description | A customer accesses their order history to view past purchases |
| Preconditions | Customer is logged in |
| Postconditions | Customer can view details of past orders |
| Normal Course | 1. Customer navigates to "Order History" section 2. System retrieves customer's order history 3. System displays list of past orders 4. Customer selects an order to view details 5. System displays detailed information about the selected order |
| Alternative Courses | - Customer searches for a specific order - Customer filters orders by date range |
| Exceptions | - No orders found in customer's history - System unable to retrieve order details |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC06 |
| Actor | Customer |
| Description | A customer writes a review for a product they have purchased |
| Preconditions | Customer is logged in and has purchased the product |
| Postconditions | Review is submitted and pending moderation |
| Normal Course | 1. Customer navigates to product page 2. Customer selects "Write a Review" option 3. System displays review form 4. Customer writes review and selects star rating 5. Customer submits review 6. System saves review and marks it for moderation 7. System displays confirmation message |
| Alternative Courses | - Customer uploads photos with review - Customer edits a previously submitted review |
| Exceptions | - Customer attempts to review a product they haven't purchased - Review submission fails due to system error |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC07 |
| Actor | System |
| Description | The system automatically processes a new order |
| Preconditions | Customer has placed an order and payment has been confirmed |
| Postconditions | Order is processed and ready for fulfillment |
| Normal Course | 1. System receives new order notification 2. System verifies payment status 3. System checks inventory availability 4. System allocates inventory to order 5. System generates packing list 6. System updates order status to "Processing" 7. System notifies fulfillment team |
| Alternative Courses | - System splits order into multiple shipments if items are in different locations - System places backorder for out-of-stock items |
| Exceptions | - Payment verification fails - Inventory is insufficient to fulfill order |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC08 |
| Actor | Customer |
| Description | A customer tracks the status and location of their order |
| Preconditions | Customer has placed an order and is logged in |
| Postconditions | Customer views current status and location of their order |
| Normal Course | 1. Customer navigates to "Track Order" page 2. Customer enters order number or selects order from list 3. System retrieves order tracking information 4. System displays order status, estimated delivery date, and current location 5. Customer views tracking information |
| Alternative Courses | - Customer accesses tracking information via email link - Customer opts for SMS updates on order status |
| Exceptions | - Tracking information is not available - Order number entered is invalid |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC09 |
| Actor | Customer |
| Description | A customer initiates the return process for an item they purchased |
| Preconditions | Customer is logged in and has a qualifying order for return |
| Postconditions | Return request is submitted and awaiting processing |
| Normal Course | 1. Customer navigates to "Returns" section in their account 2. Customer selects the order containing the item to return 3. Customer selects specific item(s) to return and reason for return 4. System verifies return eligibility 5. System provides return instructions and generates return label 6. Customer confirms return request 7. System updates order status and notifies relevant departments |
| Alternative Courses | - Customer initiates return via customer service - System offers exchange instead of return |
| Exceptions | - Item is not eligible for return - Return period has expired |

|  |  |
| --- | --- |
| **Field** | **Description** |
| Use Case ID | UC10 |
| Actor | System |
| Description | The system automatically generates a comprehensive sales report |
| Preconditions | Sales data is available in the system |
| Postconditions | Sales report is generated and available for viewing |
| Normal Course | 1. System initiates report generation at scheduled time 2. System collects sales data for the specified period 3. System calculates key metrics (revenue, profit, top-selling items, etc.) 4. System generates graphs and charts for visual representation 5. System compiles data into a formatted report 6. System saves report and sends notification to relevant staff |
| Alternative Courses | - Staff member manually triggers report generation - System generates custom report based on specific parameters |
| Exceptions | - Insufficient data for the specified period - Error in data calculation or compilation |

User story and acceptance Criteria

|  |  |
| --- | --- |
| **User Story** | **Acceptance Criteria** |
| As a new customer, I want to create an account so that I can manage my orders and preferences | - Can enter email and password - Receive confirmation email - Can log in after account creation - Can view and edit profile information |
| As a shopper, I want to add items to my cart so that I can purchase them later | - Can add items from product page - Can add items from search results - Cart icon updates with number of items - Can view cart contents at any time |
| As a customer, I want to complete the checkout process so that I can receive my purchased items | - Can enter shipping address - Can enter payment information - Can review order before final submission - Receive order confirmation email |
| As a shopper, I want to search for products so that I can find specific items quickly | - Can enter search terms in a search bar - Can filter search results - Can sort search results - Search results display relevant product information |
| As a customer, I want to view my order history so that I can track my purchases | - Can access order history from account dashboard - Can view details of past orders - Can sort and filter order history - Can initiate returns from order history |
| As a customer, I want to write product reviews so that I can share my experience with others | - Can rate product on a scale - Can write text review - Can upload photos with review - Can edit or delete my own reviews |
| As a staff member, I want the system to process orders automatically so that I can focus on exceptions | - Orders are automatically processed after payment - Inventory is updated - Fulfillment team is notified - Order status is updated in customer's account |
| As a customer, I want to track my order so that I know when to expect delivery | - Can enter order number to view status - Can see estimated delivery date - Can opt for email/SMS updates - Can view package's current location |
| As a customer, I want to initiate a return so that I can get a refund for unwanted items | - Can select items for return from order history - Can choose reason for return - Can print return label - Can track status of return and refund |
| As a manager, I want the system to generate sales reports so that I can analyze business performance | - Can view daily, weekly, monthly reports - Can see sales by product category - Can view top-selling products - Can export reports in various formats |
| As a customer, I want to apply a coupon code so that I can get a discount on my purchase | - Can enter coupon code during checkout - See updated total after coupon application - Receive error message for invalid coupons - Can remove applied coupon |
| As a staff member, I want the system to update product inventory automatically so that we avoid overselling | - Inventory updates in real-time with sales - Low stock alerts are generated - Out-of-stock products are marked accordingly - Can manually adjust inventory if needed |
| As a customer, I want to save items to a wishlist so that I can purchase them later | - Can add items to wishlist from product pages - Can view and manage wishlist - Can move items from wishlist to cart - Can share wishlist with others |
| As a customer, I want to receive an order confirmation email so that I have a record of my purchase | - Email sent immediately after order placement - Email contains order details and total - Email includes estimated delivery date - Email has link to track order |
| As a customer, I want to modify my account information so that it stays up-to-date | - Can change email address - Can update password - Can edit shipping addresses - Can update communication preferences |
| As a staff member, I want to respond to customer queries so that I can provide good customer service | - Can view customer query details - Can categorize and prioritize queries - Can send responses to customers - Can escalate complex issues |
| As a customer, I want the system to handle failed payments gracefully so that I can try again | - Receive clear error message on payment failure - Can retry payment without losing cart contents - Offered alternative payment methods - Order not processed until payment succeeds |
| As a shopper, I want to compare products so that I can make informed decisions | - Can select multiple products to compare - See side-by-side comparison of features - Can add compared products to cart - Can print or save comparison |
| As a customer, I want to receive personalized product recommendations so that I can discover relevant items | - See recommended products based on browsing history - See recommendations based on past purchases - Recommendations update in real-time - Can give feedback on recommendations |
| As a customer, I want to share products on social media so that I can get opinions from friends | - Social sharing buttons on product pages - Can choose which platform to share on - Shared content includes product image and description - Can see number of shares for a product |
| As a customer, I want to receive a refund for returned items so that I can get my money back | - Refund initiated upon receipt of returned item - Can choose refund method (original payment or store credit) - Receive email notification of refund processing - Can track status of refund in account |
| As a shopper, I want to subscribe to a newsletter so that I can stay informed about deals and new products | - Can enter email to subscribe from multiple page locations - Receive welcome email upon subscription - Can choose frequency and types of newsletters - Can easily unsubscribe |
| As a staff member, I want the system to manage flash sales so that we can boost sales during specific periods | - Can schedule flash sales in advance - System automatically starts and ends sale at set times - Sale prices clearly displayed to customers - System prevents overselling during high demand |
| As a customer, I want to use a gift card so that I can pay for my purchase | - Can enter gift card code during checkout - See updated balance after gift card application - Can combine gift card with other payment methods - Can check gift card balance |
| As a staff member, I want to update product information so that customers have accurate details | - Can edit product descriptions - Can update pricing information - Can add or remove product images - Changes reflect immediately on site |
| As a security team member, I want the system to detect fraudulent activity so that we can prevent losses | - System flags suspicious transactions - Can review flagged transactions - Can block accounts associated with fraud - System learns from confirmed fraud cases |
| As a customer, I want to view product availability at physical stores so that I can pick up items immediately | - Can enter zip code to find nearby stores - See real-time inventory at each store - Can reserve item for in-store pickup - Receive notification when item is ready for pickup |
| As a marketing team member, I want the system to send abandoned cart emails so that we can recover potential sales | - System identifies abandoned carts after set time - Customizable email templates for abandoned carts - Can set frequency and timing of reminder emails - Can track conversion rate of reminder emails |
| As a customer, I want to participate in a loyalty program so that I can earn rewards for my purchases | - Can enroll in loyalty program - Earn points for purchases - Can view point balance and rewards - Can redeem points for discounts or products |
| As a business customer, I want to place bulk orders so that I can efficiently stock my store | - Can upload CSV file for bulk orders - Receive volume discounts automatically - Can save and reuse previous bulk orders - Receive specialized support for large orders |

|  |  |
| --- | --- |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC001 |
| Use Case Name: | Customer Applies Coupon Code |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | Customer |
| Description: | The customer applies a coupon code during the checkout process to receive a discount on their purchase. |
| Preconditions: | 1. Customer is logged into their account. 2. Customer has items in their shopping cart. 3. Customer has initiated the checkout process. 4. A valid coupon code exists in the system. |
| Postconditions: | 1. Customer is logged into their account. 2. Customer has items in their shopping cart. 3. Customer has initiated the checkout process. 4. A valid coupon code exists in the system. |
| Priority: | High |
| Frequency of Use: | Frequent (Multiple times daily) |
| Normal Course of Events: | 1. Customer enters the coupon code in the designated field during checkout. 2. Customer clicks "Apply Coupon" button. 3. System validates the coupon code. 4. System applies the discount to the order. 5. System displays updated order total with applied discount. 6. Customer proceeds with the checkout process. |
| Alternative Courses: | 1a. Customer enters an invalid coupon code:  1. System displays an error message.  2. Customer is prompted to enter a valid code or continue without a coupon.  1b. Customer enters an expired coupon code:  1. System informs the customer that the coupon has expired.  2. Customer is prompted to enter a different code or continue without a coupon.  1c. Customer enters a coupon code with unmet conditions (e.g., minimum purchase amount):  1. System informs the customer of the unmet conditions.  2. Customer is given the option to adjust their order or continue without the coupon. |
| Exceptions: | 1. System experiences an error while processing the coupon:  - System logs the error and displays a generic error message to the customer.  - Customer is prompted to try again later or contact customer support.  2. Coupon database is temporarily unavailable:  - System informs the customer that coupon processing is currently unavailable.  - Customer is given the option to complete the order without the coupon or try again later. |
| Includes: | - Coupon validation process - Order total recalculation |
| Special Requirements: | 1. Coupon code validation should be case-insensitive. 2. System should process and apply the coupon within 2 seconds. 3. Coupon usage should be tracked to prevent exceeding usage limits. |
| Assumptions: | 1. The e-commerce platform has a robust coupon management system. 2. Customers are familiar with the concept of using coupon codes for discounts. |
| Notes and Issues: | 1. Consider implementing a feature to suggest available coupon codes to customers. 2. Monitor coupon usage patterns to detect potential abuse or fraud. 3. Regularly review and update coupon policies to align with business goals. |
|  |  |
| UC002: System Updates Product Inventory |  |
|  |  |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC002 |
| Use Case Name: | System Updates Product Inventory |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | System |
| Description: | The system automatically updates product inventory levels based on various triggers such as sales, returns, and manual adjustments. |
| Preconditions: | 1. Inventory management system is operational. 2. Product database is accessible. 3. There are events that necessitate inventory updates (e.g., completed sales, confirmed returns, manual stock counts). |
| Postconditions: | 1. Product inventory levels are accurately updated. 2. Inventory change log is updated. 3. Low stock alerts are triggered if applicable. |
| Priority: | Critical |
| Frequency of Use: | Very Frequent (Multiple times per hour) |
| Normal Course of Events: | 1. System receives an inventory update trigger (e.g., completed sale, confirmed return). 2. System identifies the affected product(s). 3. System calculates the new inventory level. 4. System updates the product database with the new inventory count. 5. System logs the inventory change with details (timestamp, reason, quantity changed). 6. System checks if the new inventory level is below the set threshold. 7. If below threshold, system generates a low stock alert. |
| Alternative Courses: | 1a. Manual inventory adjustment:  1. Staff member initiates a manual inventory update.  2. System prompts for verification of the adjustment.  3. Upon verification, system updates the inventory as per the normal course.  1b. Bulk inventory update (e.g., after receiving a large shipment):  1. System receives bulk update data.  2. System processes each item in the bulk update sequentially.  3. System performs normal update process for each item. |
| Exceptions: | 1. Database connection failure:  - System logs the error and queues the update for retry.  - System alerts IT support about the database issue.  2. Inventory becomes negative:  - System flags the product for review.  - System sends an alert to inventory management staff.  3. Conflicting simultaneous updates:  - System applies updates in the order received.  - System logs a warning for manual review. |
| Includes: | - Inventory level calculation - Low stock alert generation - Inventory change logging |
| Special Requirements: | 1. Updates must be processed in real-time (within 5 seconds of trigger). 2. System must handle high-volume updates during peak sales periods. 3. All inventory changes must be traceable and reversible if needed. |
| Assumptions: | 1. The e-commerce platform has an integrated inventory management system. 2. Product SKUs or unique identifiers are consistently used across all systems. |
| Notes and Issues: | 1. Consider implementing predictive inventory management based on historical data. 2. Regularly audit inventory accuracy through cycle counts. 3. Implement safeguards against system-wide inventory errors (e.g., batch update rollback capability). |
|  |  |
| UC003: Customer Uses Wishlist |  |
|  |  |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC003 |
| Use Case Name: | Customer Uses Wishlist |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | Customer |
| Description: | The customer adds, removes, or manages items in their wishlist for future purchase consideration. |
| Preconditions: | 1. Customer is logged into their account. 2. Wishlist feature is available on the e-commerce platform. |
| Postconditions: | 1. Customer's wishlist is updated with the desired changes. 2. Wishlist data is saved and synced across customer's devices. |
| Priority: | Medium |
| Frequency of Use: | Frequent (Multiple times daily) |
| Normal Course of Events: | 1. Customer navigates to a product page. 2. Customer clicks "Add to Wishlist" button. 3. System adds the item to the customer's wishlist. 4. System confirms addition with a visual indicator. 5. Customer can view their updated wishlist. 6. Customer can remove items from the wishlist as needed. |
| Alternative Courses: | 1a. Adding to wishlist from product listing page:  1. Customer clicks wishlist icon next to product in listing.  2. System adds item to wishlist without leaving the page.  1b. Managing wishlist from dedicated wishlist page:  1. Customer navigates to their wishlist page.  2. Customer can add notes, adjust quantities, or remove items.  3. System updates wishlist in real-time.  1c. Sharing wishlist:  1. Customer selects "Share Wishlist" option.  2. System generates a shareable link or allows social media sharing. |
| Exceptions: | 1. Item is out of stock:  - System adds item to wishlist but flags it as unavailable.  - Customer is notified when item becomes available again.  2. Wishlist limit reached:  - System notifies customer that the wishlist is full.  - Customer is prompted to remove items before adding new ones. |
| Includes: | - Add to wishlist functionality - Remove from wishlist functionality - Wishlist management interface |
| Special Requirements: | 1. Wishlist should sync across devices in real-time. 2. System should provide "Add to Cart" option directly from wishlist. 3. Implement price tracking for wishlist items. |
| Assumptions: | 1. Customers understand the concept and benefits of using a wishlist. 2. The e-commerce platform can handle concurrent wishlist operations from multiple users. |
| Notes and Issues: | 1. Consider implementing AI-driven recommendations based on wishlist items. 2. Explore integration with gift registry features for special occasions. 3. Implement alerts for price drops on wishlist items to encourage purchases. |
|  |  |
| UC004: System Sends Order Confirmation Email |  |
|  |  |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC004 |
| Use Case Name: | System Sends Order Confirmation Email |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | System |
| Description: | The system automatically sends an order confirmation email to the customer after a successful purchase. |
| Preconditions: | 1. Customer has completed a purchase. 2. Order has been successfully processed and payment confirmed. 3. Customer's email address is available and valid. |
| Postconditions: | 1. Order confirmation email is sent to the customer. 2. Email sending is logged in the system. 3. Order status is updated to reflect that confirmation has been sent. |
| Priority: | High |
| Frequency of Use: | Very Frequent (After every completed order) |
| Normal Course of Events: | 1. System detects a successfully completed order. 2. System generates an order confirmation email with order details. 3. System sends the email to the customer's provided email address. 4. System logs the email sending event. 5. System updates the order status to "Confirmation Sent". |
| Alternative Courses: | 1a. Customer opts for SMS confirmation:  1. System sends an SMS instead of or in addition to the email.  1b. Order requires special handling:  1. System includes additional information or instructions in the email. |
| Exceptions: | 1. Email sending fails:  - System logs the failure and queues for retry.  - After multiple failures, system flags the order for manual follow-up.  2. Invalid email address:  - System logs the error and flags the order for manual contact.  - Customer support is notified to contact the customer via alternative means. |
| Includes: | - Email content generation - Email sending process - Order status update |
| Special Requirements: | 1. Emails should be sent within 5 minutes of order completion. 2. Email content should be personalized and include all relevant order details. 3. System should support multiple languages based on customer preference. |
| Assumptions: | 1. An email server or service is integrated with the e-commerce platform. 2. Customer has agreed to receive transactional emails. |
| Notes and Issues: | 1. Consider implementing email open tracking for analytics. 2. Regularly review and update email templates for clarity and branding consistency. 3. Ensure compliance with anti-spam laws and regulations in all operating jurisdictions. |
|  |  |
| UC005: Customer Modifies Account Information |  |
|  |  |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC005 |
| Use Case Name: | Customer Modifies Account Information |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | Customer |
| Description: | The customer updates their personal information, preferences, or account settings on the e-commerce platform. |
| Preconditions: | 1. Customer is logged into their account. 2. Customer has navigated to the account settings or profile page. |
| Postconditions: | 1. Customer's account information is updated in the system. 2. Confirmation of changes is provided to the customer. 3. Relevant systems are updated with the new information. |
| Priority: | Medium |
| Frequency of Use: | Occasional (Several times per month) |
| Normal Course of Events: | 1. Customer selects "Edit Profile" or "Account Settings". 2. System displays current account information. 3. Customer modifies desired fields (e.g., name, address, email, password). 4. Customer submits changes. 5. System validates the modifications. 6. System updates the account information in the database. 7. System confirms successful update to the customer. |
| Alternative Courses: | 1a. Changing email address:  1. System sends verification email to the new address.  2. Customer must confirm via link before change is finalized.  1b. Updating payment information:  1. System requires re-authentication for security.  2. System encrypts and securely stores new payment details. |
| Exceptions: | 1. Invalid input:  - System highlights erroneous fields.  - Customer is prompted to correct the information.  2. System unable to save changes:  - System logs the error and displays a message to the customer.  - Customer is advised to try again later or contact support. |
| Includes: | - Profile information update - Password change process - Email verification (if applicable) - Security checks for sensitive changes |
| Special Requirements: | 1. Implement strong password requirements. 2. Provide option to enable two-factor authentication. 3. Ensure GDPR compliance for EU customers. |
| Assumptions: | 1. Customer has a stable internet connection. 2. The e-commerce platform has a user-friendly interface for account management. |
| Notes and Issues: | 1. Consider implementing progressive profiling to gather additional |
| UC006: Staff Responds to Customer Query |  |
|  |  |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC006 |
| Use Case Name: | Staff Responds to Customer Query |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | Customer Support Staff |
| Description: | A customer support staff member responds to a customer's query through the e-commerce platform's support system. |
| Preconditions: | 1. Customer has submitted a query through the platform's support channel. 2. Staff member is logged into the customer support system. 3. Query is assigned to or picked up by the staff member. |
| Postconditions: | 1. Staff response is recorded in the system. 2. Customer is notified of the response. 3. Query status is updated (e.g., to "Responded" or "Resolved"). |
| Priority: | High |
| Frequency of Use: | Very Frequent (Multiple times per hour) |
| Normal Course of Events: | 1. Staff member reviews the customer query. 2. Staff member researches the issue if necessary. 3. Staff member composes a response addressing the customer's concerns. 4. Staff member submits the response through the support system. 5. System sends the response to the customer. 6. System updates the query status. 7. Staff member closes the ticket if the issue is resolved. |
| Alternative Courses: | 1a. Query requires escalation:  1. Staff member flags the query for escalation.  2. System assigns the query to a higher-tier support team.  1b. Staff member initiates follow-up:  1. Staff sets a reminder for follow-up.  2. System prompts staff at the designated time. |
| Exceptions: | 1. System outage during response:  - Staff's response is auto-saved as a draft.  - Staff is notified to retry once system is back online.  2. Customer has multiple open queries:  - System alerts staff to related open queries.  - Staff can choose to consolidate responses if appropriate. |
| Includes: | - Query review process - Response composition - Query status update - Customer notification |
| Special Requirements: | 1. Responses should be sent within the company's SLA timeframe. 2. Staff should have access to customer purchase history and previous interactions. 3. System should support canned responses for common queries. |
| Assumptions: | 1. Staff are trained in customer service protocols. 2. The support system has necessary tools for efficient query handling. |
| Notes and Issues: | 1. Consider implementing AI-assisted response suggestions. 2. Regularly update FAQ and knowledge base to reduce repetitive queries. 3. Implement customer satisfaction surveys after query resolution. |
|  |  |
| UC007: System Handles Failed Payment |  |
|  |  |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC007 |
| Use Case Name: | System Handles Failed Payment |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | System |
| Description: | The system automatically handles a failed payment attempt during the checkout process. |
| Preconditions: | 1. Customer has attempted to complete a purchase. 2. Payment processing has failed for any reason (e.g., insufficient funds, expired card). |
| Postconditions: | 1. Customer is notified of the payment failure. 2. Order is placed on hold or cancelled based on system settings. 3. Payment failure is logged for analysis and reporting. |
| Priority: | Critical |
| Frequency of Use: | Frequent (Multiple times daily) |
| Normal Course of Events: | 1. System detects a failed payment. 2. System identifies the reason for failure from payment processor response. 3. System updates the order status to "Payment Failed". 4. System generates a user-friendly error message. 5. System displays the error message to the customer. 6. System sends an email notification about the failed payment. 7. System provides options to retry payment or use alternative payment method. |
| Alternative Courses: | 1a. Temporary hold on order:  1. System places the order on hold for a predefined period.  2. System sends reminders to complete payment.  3. If payment isn't completed within the timeframe, order is cancelled.  1b. Automatic retry:  1. For certain error types, system attempts to retry the payment automatically.  2. Customer is notified of the retry attempt and result. |
| Exceptions: | 1. Payment processor is unavailable:  - System logs the error and notifies IT support.  - Customer is informed of technical difficulties and asked to try later.  2. Suspected fraudulent activity:  - System flags the transaction for review.  - Order is put on hold pending manual verification. |
| Includes: | - Payment failure detection - Order status update - Customer notification - Payment retry options |
| Special Requirements: | 1. System should handle failures gracefully without exposing technical details to customers. 2. Provide clear instructions for customers to resolve payment issues. 3. Implement secure logging of payment failure reasons for PCI compliance. |
| Assumptions: | 1. The e-commerce platform is integrated with a reliable payment processing system. 2. Customers have alternative payment methods available. |
| Notes and Issues: | 1. Regularly analyze payment failure patterns to identify and address recurring issues. 2. Consider implementing smart retry logic based on failure reason. 3. Ensure customer support is promptly notified of repeated payment failures. |
|  |  |
| UC008: Customer Compares Products |  |
|  |  |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC008 |
| Use Case Name: | Customer Compares Products |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | Customer |
| Description: | The customer uses the product comparison feature to evaluate and compare multiple products side-by-side. |
| Preconditions: | 1. Customer is browsing products on the e-commerce platform. 2. At least two comparable products are available. 3. Product comparison feature is enabled on the platform. |
| Postconditions: | 1. Customer views a side-by-side comparison of selected products. 2. Comparison data is temporarily saved for the session. |
| Priority: | Medium |
| Frequency of Use: | Frequent (Multiple times daily) |
| Normal Course of Events: | 1. Customer selects "Add to Compare" for desired products. 2. System adds products to comparison list. 3. Customer navigates to the comparison page. 4. System retrieves detailed information for each product. 5. System displays products in a side-by-side comparison view. 6. Customer reviews the comparison data. 7. Customer can add or remove products from comparison. |
| Alternative Courses: | 1a. Comparing products from different categories:  1. System groups comparable attributes.  2. System clearly indicates non-applicable features.  1b. Saving comparison for later:  1. Customer chooses to save the comparison.  2. System stores the comparison in the customer's account. |
| Exceptions: | 1. Too many products added to compare:  - System limits the number of products that can be compared simultaneously.  - Customer is prompted to remove items before adding new ones.  2. Product becomes unavailable during comparison:  - System flags the product as unavailable in the comparison view.  - Customer is given the option to remove it from comparison. |
| Includes: | - Add to compare functionality - Comparison view generation - Dynamic attribute comparison - Add/remove products from comparison |
| Special Requirements: | 1. Comparison should be responsive and work well on mobile devices. 2. Implement a print-friendly version of the comparison. 3. Allow sharing of product comparisons via link or social media. |
| Assumptions: | 1. Products have standardized attributes for fair comparison. 2. The e-commerce platform can handle concurrent comparison requests from multiple users. |
| Notes and Issues: | 1. Consider implementing AI-driven recommendations based on compared products. 2. Regularly review and update product attributes to ensure meaningful comparisons. 3. Analyze comparison data to identify popular feature comparisons and improve product listings. |
|  |  |
| UC009: System Generates Product Recommendations |  |
|  |  |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC009 |
| Use Case Name: | System Generates Product Recommendations |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | System |
| Description: | The system automatically generates personalized product recommendations for customers based on various factors such as browsing history, purchase history, and similar user behavior. |
| Preconditions: | 1. Customer data (browsing history, purchase history) is available. 2. Recommendation algorithm is in place and operational. 3. Sufficient product data exists in the system. |
| Postconditions: | 1. Personalized product recommendations are generated. 2. Recommendations are displayed to the customer at appropriate touchpoints. |
| Priority: | High |
| Frequency of Use: | Very Frequent (Multiple times per session) |
| Normal Course of Events: | 1. System identifies the customer (logged in or via cookie). 2. System retrieves customer's historical data. 3. System analyzes data using recommendation algorithm. 4. System generates a list of recommended products. 5. System displays recommendations at relevant locations (e.g., product pages, homepage, email). |
| Alternative Courses: | 1a. New customer with no history:  1. System uses popular or trending items for recommendations.  2. System quickly adapts as customer interacts with the site.  1b. Seasonal or promotional recommendations:  1. System incorporates current promotions or seasonal trends.  2. Recommendations are adjusted to highlight relevant offers. |
| Exceptions: | 1. Insufficient data for personalization:  - System falls back to general popularity-based recommendations.  - System logs the event for algorithm improvement.  2. Recommended product becomes unavailable:  - System dynamically replaces the recommendation.  - Unavailable products are excluded from future recommendations until restocked. |
| Includes: | - Customer data analysis - Recommendation algorithm processing - Dynamic recommendation display - Recommendation performance tracking |
| Special Requirements: | 1. Recommendations should be generated in real-time (< 500ms). 2. System should handle high volumes during peak traffic periods. 3. Implement A/B testing capability for recommendation strategies. |
| Assumptions: | 1. Customer consents to data use for personalization. 2. Product catalog is regularly updated with new items. |
| Notes and Issues: | 1. Regularly review and refine the recommendation algorithm. 2. Consider implementing multi-armed bandit algorithms for continuous optimization. 3. Ensure recommendations comply with privacy regulations (e.g., GDPR, CCPA). |
|  |  |
| UC010: Customer Shares Product on Social Media |  |
|  |  |
| Field | Value |
| ------- | ------- |
| Use Case ID: | UC010 |
| Use Case Name: | Customer Shares Product on Social Media |
| Created By: | System Analyst |
| Date Created: | 2024-10-21 |
| Last Updated By: | System Analyst |
| Date Last Updated: | 2024-10-21 |
|  |  |
| Field | Description |
| ------- | ------------- |
| Actor: | Customer |
| Description: | The customer shares a product from the e-commerce site on their social media platform of choice. |
| Preconditions: | 1. Customer is viewing a product page on the e-commerce site. 2. Social media sharing functionality is enabled on the platform. 3. Customer has an active account on the chosen social media platform. |
| Postconditions: | 1. Product information is shared on the customer's social media account. 2. Share event is logged in the e-commerce system for analytics. |
| Priority: | Medium |
| Frequency of Use: | Occasional (Several times per week) |
| Normal Course of Events: | 1. Customer clicks on the "Share" button on a product page. 2. System displays a list of available social media platforms. 3. Customer selects their preferred platform. 4. System generates a sharing link with product details and image. 5. System opens a sharing dialog for the selected platform. 6. Customer customizes the share message (optional). 7. Customer confirms the share. 8. System logs the sharing event. |
| Alternative Courses: | 1a. Customer uses platform-specific share button:  1. System generates sharing content specific to that platform.  2. Customer is directed to the platform to complete the share.  1b. Customer copies share link manually:  1. System provides a copyable link with product details.  2. Customer copies and shares the link manually. |
| Exceptions: | 1. Social media platform API is unavailable:  - System displays an error message.  - Customer is offered alternative sharing methods.  2. Customer is not logged into the selected social media platform:  - System prompts customer to log in to the platform.  - Share process continues after successful login. |
| Includes: | - Social media integration - Share link generation - Analytics tracking |
| Special Requirements: | 1. Shared content should be optimized for each social media platform. 2. Implement Open Graph tags for rich media sharing. 3. Ensure quick loading of shared links on social media platforms. |
| Assumptions: | 1. Customers are willing to share product information on their social networks. 2. The e-commerce platform has necessary permissions to integrate with social media APIs. |
| Notes and Issues: | 1. Monitor and analyze which products are shared most frequently. 2. Consider implementing incentives for customers who share products. 3. Regularly update social media integration to accommodate platform changes. |

CRC

|  |  |  |
| --- | --- | --- |
| **Class** | **Responsibilities** | **Collaborators** |
| Staff | - Manage employee records - Assign tasks and responsibilities - Evaluate performance - Provide training and development | - Manager - Department - Employee |
| Manager | - Oversee department operations - Make strategic decisions - Allocate resources - Provide leadership and guidance | - Staff - Department - Employee |
| Employee | - Perform assigned tasks and duties - Communicate with manager and colleagues - Contribute to department goals - Maintain productivity and efficiency | - Staff - Manager - Department |
| Department | - Coordinate team activities - Manage departmental resources - Ensure alignment with organizational objectives - Collaborate with other departments | - Staff - Manager - Employee |
| Address | - Store location information - Validate address data - Provide address details for various purposes | - Person - Customer - Shipping |
| Person | - Manage personal information - Maintain contact details - Provide identification data | - Employee - Customer - Address |
| FullName | - Store first and last name - Provide formatted name display - Validate name integrity | - Person |
| Account | - Store user login credentials - Manage account settings - Handle authentication and authorization | - Customer - Security |
| Cart | - Add and remove products - Calculate total price - Apply discounts - Save cart for later use | - Customer - Product - Order |
| CartItem | - Represent a product in the cart - Track quantity and price - Provide item details | - Cart - Product |
| Product | - Store product information - Provide product details - Manage product categories - Handle product searches | - ProductCategory - Supplier - Image - Review |
| ProductCategory | - Organize products into categories - Manage category hierarchy - Provide category-specific details | - Product - ProductCatalog |
| Supplier | - Provide product inventory - Handle product procurement - Manage supplier-related information | - Product |
| Image | - Store product images - Resize and optimize images - Manage image galleries - Handle image uploads | - Product - User Interface |
| Video | - Store product videos - Manage video quality and resolution - Provide video playback capabilities | - Product - User Interface |
| Resolution | - Represent video or image resolution - Provide resolution details - Ensure consistent quality | - Image - Video |
| Review | - Store customer reviews - Calculate product ratings - Display review information - Moderate review content | - Customer - Product |
| Customer | - Manage account information - Place orders - Provide shipping and payment info - Leave feedback and ratings | - User Interface - Cart - Order - Payment - Review |
| Notification | - Send order confirmations - Provide updates to customers - Alert customers about promotions - Manage notification preferences | - Customer - Order Management - Marketing |
| Email | - Send email notifications - Manage email templates - Track email deliverability | - Notification - Customer |
| InApp | - Deliver in-app notifications - Manage push notification preferences - Integrate with mobile platforms | - Notification - Customer |
| Phone | - Store customer phone numbers - Provide phone-based communication - Handle SMS and voice interactions | - Customer - Notification |
| Payment | - Process customer payments - Handle payment methods - Verify payment information - Issue refunds | - Customer - Order - PaymentType - Transaction |
| PaymentType | - Represent different payment options - Manage payment processing details - Provide payment-specific functionality | - Payment |
| Transaction | - Record payment transactions - Manage transaction statuses - Facilitate payment processing | - Payment - Order |
| Order | - Combine cart items - Calculate final price - Track order status - Store order history | - Cart - Customer - Order Management - Payment - Shipping |
| VoucherVault | - Store active vouchers - Manage voucher redemption - Provide voucher-related data | - Voucher - Order |
| Voucher | - Represent discount or promotional codes - Apply vouchers to orders - Track voucher usage and validity | - VoucherVault - Order |
| Warehouse | - Store product inventory - Manage warehouse locations - Coordinate shipping and returns | - Inventory - Shipping |
| Shipping | - Calculate shipping costs - Manage available shipping methods - Coordinate with shipping carriers - Track shipment status | - Order - Order Management - Warehouse - ShippingDevice |
| Shipper | - Represent a shipping carrier - Provide shipping rates and services - Handle shipment tracking and updates | - Shipping |
| ShippingDevice | - Manage a fleet of delivery devices - Plan efficient delivery routes - Monitor device status and maintenance | - Shipping - GPS |
| ShippingMethod | - Encapsulate shipping service details - Provide shipping cost calculations - Facilitate shipment processing | - Shipping |
| GPS | - Provide location data for shipments - Calculate optimal delivery routes - Track shipment progress and status | - ShippingDevice - Shipping |
|  | - Track order status | - Order Management System |
|  | - Store order history | - Payment System |
|  |  | - Shipping Department |
| Order Management System | - Process new orders | - Order |
|  | - Update order status | - Inventory |
|  | - Coordinate shipping and inventory | - Shipping Department |
|  | - Handle order cancellations and returns | - Billing System |
| Payment System | - Process payments | - Customer |
|  | - Verify payment information | - Order |
|  | - Handle payment methods | - Payment Gateway |
|  | - Issue refunds | - Billing System |
| Shipping Department | - Calculate shipping costs | - Order |
|  | - Manage shipping methods | - Order Management System |
|  | - Coordinate couriers | - Warehouse Management System |
|  | - Track shipments | - Drone System |
| Warehouse Management System | - Manage inventory locations | - Inventory |
|  | - Optimize picking and packing | - Shipping Department |
|  | - Coordinate shipping | - Order Management System |
|  | - Handle returns |  |
| Drone System | - Manage drone fleet | - Shipping Department |
|  | - Plan delivery routes | - GPS System |
|  | - Monitor drone status |  |
|  | - Handle drone maintenance |  |
| Customer Service | - Handle customer inquiries | - Customer |
|  | - Process returns and exchanges | - Order Management System |
|  | - Resolve customer disputes | - Inventory |
|  | - Provide product support | - Shipping Department |
| Analytics Module | - Collect and analyze data | - System |
|  | - Generate performance reports | - Marketing Module |
|  | - Provide marketing insights |  |
|  | - Monitor system performance |  |
| Marketing Module | - Manage promotions | - Customer |
|  | - Handle email marketing | - Analytics Module |
|  | - Personalize customer experiences |  |
|  | - Manage loyalty programs |  |
| Search Engine | - Index product catalog | - Product Catalog |
|  | - Process search queries | - User Interface |
|  | - Rank search results | - Analytics Module |
|  | - Provide autocomplete suggestions |  |
| Recommendation Engine | - Analyze user behavior | - Product Catalog |
|  | - Generate product recommendations | - Customer |
|  | - Personalize product listings | - Analytics Module |
|  | - Perform A/B testing |  |
| Notification System | - Send order confirmations | - Customer |
|  | - Provide updates to customers | - Order Management System |
|  | - Alert customers about promotions | - Marketing Module |
|  | - Manage notification preferences |  |

5.