

Ajido, Andriq Klyne
Cortez, Leo Anthony

ACTIVITY

You are tasked with designing a software solution for an online shopping system. The system should allow users to **browse products**, **add items to their shopping cart**, **proceed to checkout**, and **make payments securely**. Additionally, the system should support **user registration and authentication**, **order management**, and **inventory tracking**.

Requirements

- Users should be able to **register and log in** to the system using their email address and password.
- The system should **display a catalog of products**, including details such as name, description, price, and availability.
- Users should be able to **search for products**, **filter them by category**, and **view product details**.
- Users should be able to **add items to their shopping cart**, **adjust quantities**, and **remove items if needed**.
- Users should be able to **proceed to checkout**, enter shipping and billing information, and select a payment method.
- The system should **process payments securely**, using a **third-party payment gateway**.
- Upon successful payment, users should receive an **order confirmation** with details of their purchase.
- **Administrators should be able to manage product listings, view and fulfill orders, and track inventory levels.**

Requirement Analysis

An online shopping system that lets its users first register and authenticate their users' accounts using email and password. Furthermore, letting them browse product catalogs displaying details which are name, description, price, and availability (search, filter by categories, view details), shopping cart, and check out products. On the shopping cart, users must be able to add products, adjust quantities, and remove items. Then if the user wanted to check out a product, he should enter his shipping and billing information. He should select payment methods using third-party payment such as g-cash, Maya, and ING. The system will handle the process of using a third-party payment gateway. After the user makes his payment, he will receive an invoice confirming his order with the details of her purchase. Additionally, software is included for administrators. This gives the administrators to manage product listings, view and fulfill orders, and track inventory levels.

As the system is being developed it should comply with these additional requirements.

1. **Performance:** should be responsive and capable of handling concurrent user requests without significant delay.
2. **Security:** implement robust security measures to protect user data, including encryption of sensitive information and secure authentication mechanisms.
3. **Compatibility:** compatible with various devices and browsers to provide a seamless experience across different platforms.
4. **Error Handling:** should effectively handle errors and provide meaningful error messages to users to assist in troubleshooting.
5. **Usability:** the user interface should be intuitive and user-friendly, allowing users to navigate the platform easily and complete transactions without confusion.
6. **Scalability:** the architecture should be scalable to accommodate increasing numbers of users and products without compromising performance.

During development, there are potential risks associated with the development of an online shopping system. The following are the potential risks involved in the development and possible strategies to eliminate or lessen its impacts:

1. **Resource Constraints:** the limited availability of appropriately skilled professionals can cause delays in the development.
Strategy: identify key resource requirements early in the project and secure necessary resources through recruitment, training, or outsourcing.
2. **Project requirement adjustments:** changes in project scope and requirements may occur without proper control, leading to schedule delays and increased costs.
Strategy: implement formal change management processes to evaluate and approve changes in project scope or requirements.
3. **Third-party Dependencies:** it may introduce risks related to availability, reliability, or compatibility.
Strategy: conduct a thorough evaluation and establish clear agreements regarding service levels, support mechanisms, and intellectual property rights.

Functionalities:

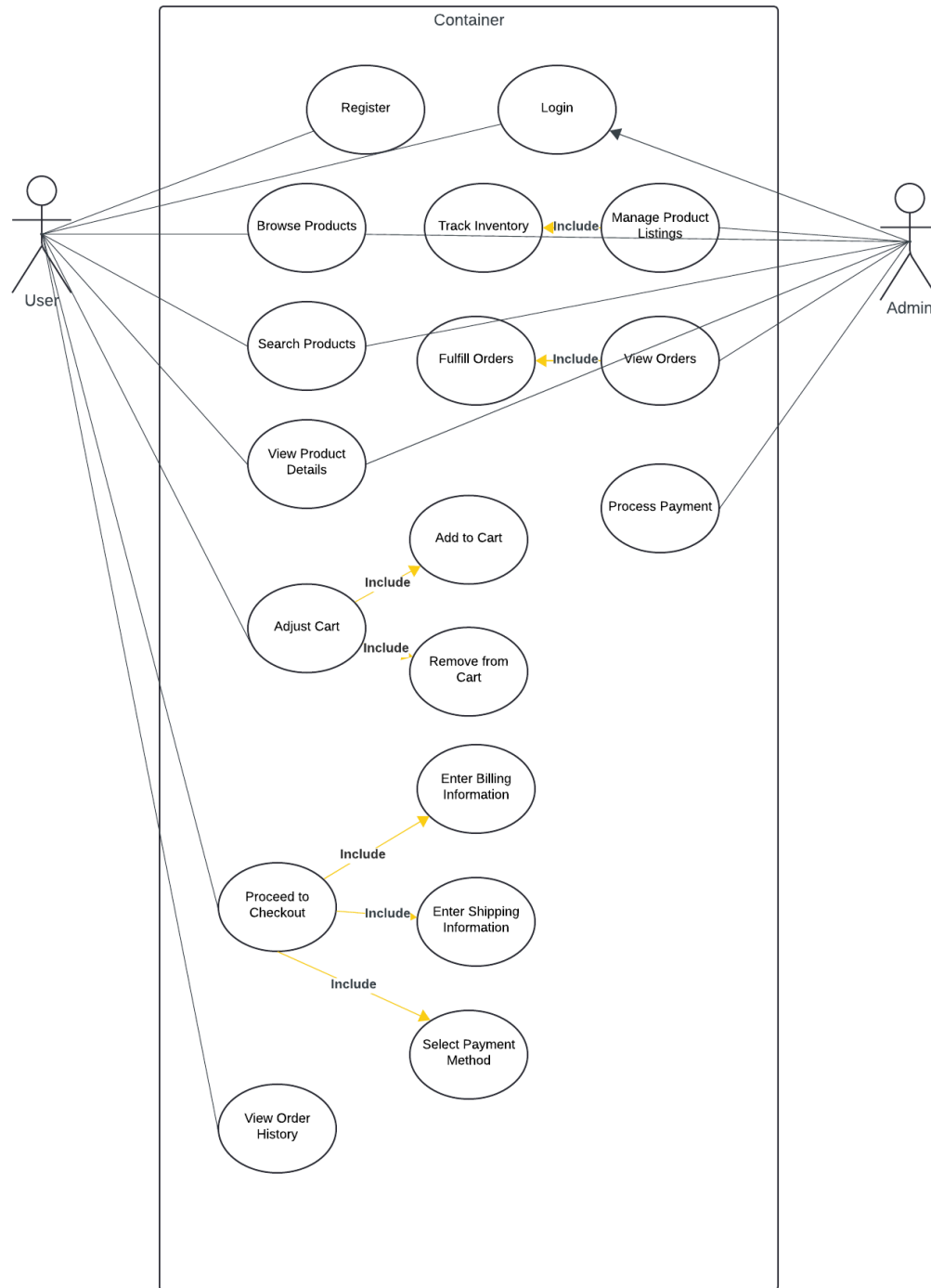


Figure 1. Use Case Diagram of the Online Shopping System

Users and administrators in the online shopping system have distinct roles and responsibilities. Users can register, and sign in, while admins can only sign in. This

ensures security for the administrators and authority over the system. The products are viewable, searchable, and navigable by both users and admins. But users have the ability to add and remove products from their cart, and potentially checkout. Administrators have the ability to manage product listings, view orders, and process payments. This distinction of tasks and roles ensures efficiency in the system, satisfying the needs of both users and administrators.

Requirement Decomposition and Functionality Decomposition:

1. Browse/Search/View products

Decomposed functionalities:

1. Display Product listing

- Make a grid or list of products with fields such as its image, name, price, stock, etc.
- Allow sorting options such as by category, price, location

2. Filter products

- Have filter options (category, price, location, popularity)
- Dynamically update product listings based on the filter

3. Search products

- Show a search bar where users can input text to find products
- Implement search functionality where it can match with either the name, description, brand, and other attributes of the product
- Display the result in a clear manner

4. View product details

- Enable users to click on a product and it will lead to the full detailed view of the product
- Show product images, and reviews
- Include recommendations

5. Add to wishlist

- Allow users to save products

2. Secure Payments

Decomposed Functionalities:

1. Select payment methods

- Present to users different payment options (PayPal, G-Cash, Cash, COD, etc)
- Allow users to choose preferred payment in checking out

2. Enter payment information

- Gather necessary payment information
- Ensure a secure payment and encrypted connection

3. Process payment

- Ensure a secure transaction
- Transmit payment details securely

4. Handle payment errors

- Implement error handling mechanism
- Offer alternative payment methods to resolve issues

3. Shopping Carts

Decomposed Functionalities:

1. Add to cart

- Allow users to add products to their cart

2. View Chart

- Allow users to view their cart any time
- Display the summary of all products added to cart

3. Remove from cart

- Enable users to remove or clear products from their carts

4. Adjust Cart

- Provide functionality to modify items in their cart

5. Cart persistent

- Implement functionality where the items inside the cart is persistent across user sessions and devices

4. Registering and Logging in

Decomposed Functionalities:

1. User registration

- Allow users to register for an account
- Collect information such as name, email address, password, etc
- Implement validation to ensure data accuracy
- Send verification email to ensure validation

2. User Login

- Implement attractive UI for the login page
- Implement authentication for registered users
- Have authentication mechanisms such as hashing of passwords
- Have a forgot password method

3. Two-factor authentication

- Allow users to have 2FA for a more secure authentication
- Implement methods such as email verification, SMS codes, etc

Prioritization:

High Priorities

1. User authentication
2. Registration
3. Product browsing
4. Product searching
5. Shopping cart
6. Checkout process
7. Secure Payment process
8. Order management and tracking

Medium Priorities

1. User account management
2. Product listing updates
3. Order history

Low Priorities

1. Two factor authentication
2. Wishlist management
3. Personal recommendations for the users

Establish Clear Interfaces

The following are the interfaces included to the software:

1. Authentication Interface

Methods:

- boolean register (string email, string password)
- boolean login (string email, string password)

2. Shopping Cart Interface

Methods:

- void addToCart (int productId, int quantity)
- void updateCart (int productId, int quantity)
- void removeFromCart (int productId)

3. Product Catalog Interface

Methods:

- list of object getProducts()
- list of object searchProduct (string query)
- list of object filterProduct (list string filters)
- object getProductDetails (int productId)

4. Checkout Interface

Methods:

- void processCheckout (string shippingInfo, string billingInfo, string paymentMethod)

5. Invoice Interface

Methods:

- void sendInvoice (int userId, Order order)

6. Payment Interface

Methods:

- void processPayment (object paymentDetails)

7. Product Management Interface

Methods:

- void addProduct (object productDetails)
- void updateProduct (int productId, object newProductDetails)
- void deleteProduct(int productId)
- object getProducts ()

8. Order Management Interface

Methods:

- list of object getOrders()
- void fulfillOrder(int orderId)
- void cancelOrder(int orderId)

9. Reports Management Interface

Methods:

- double getSales()
- double getRevenue()
- object inventoryTrend ()

Hierarchical Structure

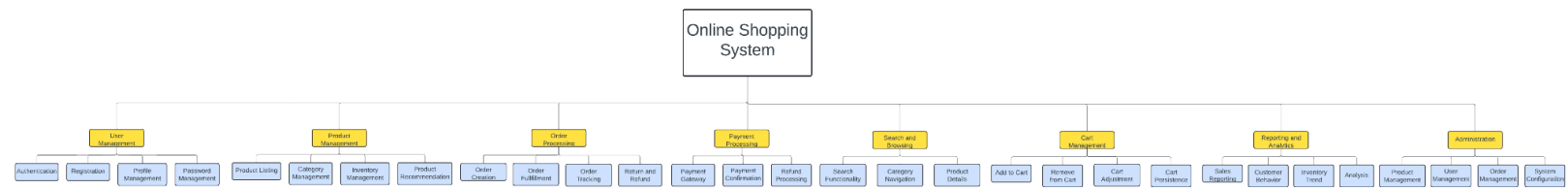


Figure 2. Full Image of the Hierarchical Structure of the Online Shopping System

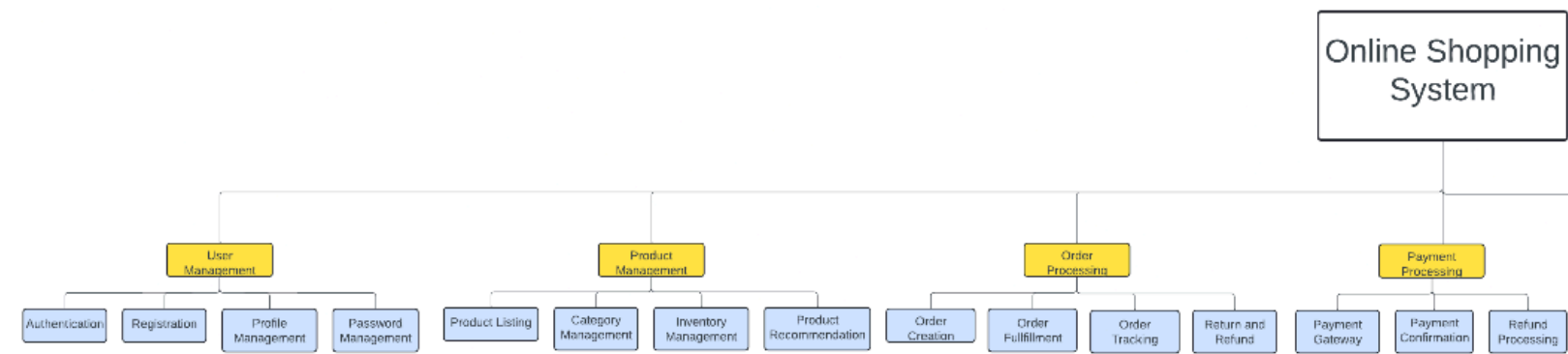


Figure 2.1 Left side of the Hierarchical Structure of the Online Shopping System

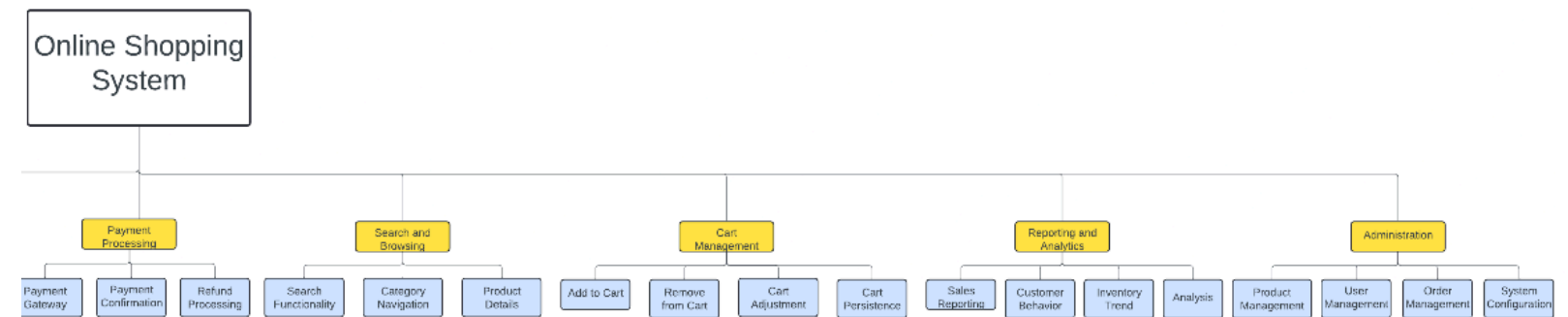


Figure 2.2 Right side of the Hierarchical Structure of the Online Shopping System

1. Online Shopping System

- Represents the entire system and it encompasses all the subsystems and functionalities

2. Subsystems

- Consists of 8 main subsystems (user management, product management, order processing, payment processing, searching and

browsing, cart management, report and analytics, and administration), each are responsible for a specific task or role in the system.

3. User management

- responsible for handling user functionalities such as registration, authentication, profile management

4. Product management

- responsible for product related task such as listing, categorization, and inventory management, and recommendations

5. Order processing

- Manages orders, and responsible for creating, fulfilling, tracking, and handling returns and orders
- Comprises of modules order creation, order fulfillment, order tracking, and return and refund

6. Payment processing

- responsible for the security of payment processes
- Includes modules such as payment gateway, payment confirmation, and refund processing

7. Search and Browsing

- responsible for product discovery through search function
- Contains modules like search functionality, category navigation, and product display

8. Cart management

- Manages carts. It allows users to add, remove, adjust items in their carts
- Consist of modules such as add to cart, remove from cart, cart **adjustment, and items adjustment**

9. Reporting and analysis

- Provides insights into sales, revenue, customer behavior, analysis, and inventory trends

10. Administration

- Responsible for managing products, users, orders, payment process, and system settings
- Contains modules for product management interface, user management, order management, and system configuration.

Document Design Decisions

In this project, the Structured Design method is used as a guiding principle for designing the software. This design emphasizes modularity and hierarchy. In addition, it breaks down a software system into smaller, more manageable components, making it easier to develop, test, and maintain the system. It involves techniques like functional

decomposition, data flow diagrams (DFDs), data dictionary, structured design, and modular programming to create a clear and organized structure for the software system.

The key principles of structured design methodology include functional decomposition, data flow, modular programming, coupling and cohesion, and structured flowcharting. These principles guide the systematic approach used in software engineering to design software systems effectively. Functional decomposition involves breaking down a system into smaller, more manageable functions, making it easier to develop and maintain. Data flow focuses on how data moves through the system, ensuring efficient processing and storage. Modular programming emphasizes creating independent modules that perform specific functions, promoting reusability and maintainability. Coupling and cohesion refer to the relationships between modules, aiming for low coupling and high cohesion to enhance system flexibility and readability. Structured flowcharting involves creating visual representations of the system's processes, aiding in understanding and communication during the design phase.

In summary, structured design methodology emphasizes on decomposition, organization, communication, and the use of data flow diagrams in the design process.