

E-Commerce Application

ARCHITECTURALLY SIGNIFICANT REQUIREMENT DOCUMENT



July 3, 2023

Arch Neo 2023, Batch 2, Group -2

Ramesh Selvam,

Shubhanjan Banerjee,

Chiranjit Show,

Rama Krishnan Kumar Poraiyan,

Naveen Narendula,

Bhargav Iyer

Table of Contents

[Introduction 3](#_Toc139142050)

[Documentation Purpose and Scope 3](#_Toc139142051)

[Product Perspective: 3](#_Toc139142052)

[Product Scope: 3](#_Toc139142053)

[Platform Development 3](#_Toc139142054)

[User Registration and Authentication 3](#_Toc139142055)

[Product Catalogue Management 3](#_Toc139142056)

[Shopping Cart Functionality 3](#_Toc139142057)

[Checkout and Payment 3](#_Toc139142058)

[Order Tracking 4](#_Toc139142059)

[Account Management 4](#_Toc139142060)

[Notification System 4](#_Toc139142061)

[Integration with Supply Chain 4](#_Toc139142062)

[Performance and Scalability 4](#_Toc139142063)

[Security and Privacy 4](#_Toc139142064)

[Localization 4](#_Toc139142065)

[Intended Audience 5](#_Toc139142066)

[Business Owners and Management 5](#_Toc139142067)

[Product Managers 5](#_Toc139142068)

[Software Architects 5](#_Toc139142069)

[Developers and Engineers 5](#_Toc139142070)

[Quality Assurance/Testers 5](#_Toc139142071)

[User Experience (UX) Designers 5](#_Toc139142072)

[~~Marketing and Sales Teams~~ 5](#_Toc139142073)

[~~End Users~~ 5](#_Toc139142074)

[Business Context 6](#_Toc139142075)

[Stakeholders 7](#_Toc139142076)

[Technical Constraints 8](#_Toc139142077)

[Business Constraints 9](#_Toc139142078)

[Quality attribute requirements 10](#_Toc139142079)

[Use Case 12](#_Toc139142080)

[User searches for products 12](#_Toc139142081)

[User adds products to the cart 13](#_Toc139142082)

[User checks out and places an order 14](#_Toc139142083)

[User tracks an order 15](#_Toc139142084)

[User views order history 16](#_Toc139142085)

[User views invoice 17](#_Toc139142086)

[Administrator manages products 18](#_Toc139142087)

[Administrator manages orders 19](#_Toc139142088)

[Supplier manages inventory 21](#_Toc139142089)

[Supplier manages product catalogue 23](#_Toc139142090)

[System Landscape View: 25](#_Toc139142091)

[Context View 26](#_Toc139142092)

[Container View 27](#_Toc139142093)

[Component View 28](#_Toc139142094)

[Deployment View (Development) 29](#_Toc139142095)

[Deployment View (Production) 30](#_Toc139142096)

[DSL: 31](#_Toc139142097)

# Introduction

## Documentation Purpose and Scope

The ecommerce system provides a platform for conducting sales of a wide variety of goods and provides a way of bringing inventory management and customers on an online platform to conduct transactions in a secure manner across the globe. It is implemented as an online enterprise. This system provides an avenue for customers to shop from a wide variety of products online. It also provides a platform where they can update their inventory to the system for customers to view and purchase. The biggest advantage of the service is the comfort it brings with remote usage. The ability to compare various price ranges, brands and even customer reviews and experiences provides for a more honest/depth understand.

The vision of the ecommerce system is to be able to provide a smooth and user-friendly platform for customers to select from a wide range of products conveniently and purchased and to cater to the needs of both customers and inventory manager.

## Product Perspective:

The system includes the user subsystem as well the inventory subsystem. The ecommerce platform provides an outstanding way of bringing and customers on an online platform to make purchases in an efficient and secure manner irrespective of the distance between the two. It is a platform for customers to shop items online without having to visit a store physically, and a platform for inventory manager to sell their items online without having to meet the customers physically or have a physical store set up for his products. This system is a one stop for customers to shop from millions of products online. The inventory manager uploads his listing to the system and the customers browse from these items and purchase them.

## Product Scope:

### Platform Development

Develop a web-based e-commerce platform that allows customers to browse and purchase furniture and appliances online. The ecommerce system provides a platform for conducting sales of a wide variety of goods and provides a way of bringing inventory management and customers on an online platform to conduct transactions in a secure manner across the globe. It is implemented as an online enterprise.

### User Registration and Authentication

Implement a user registration and authentication system to allow users to create accounts, log in, and securely access the platform.

### Product Catalogue Management

Create a searchable product catalogue with categories, allowing users to browse and search for specific furniture and appliances. The ability to compare various price ranges, brands and even customer reviews and experiences provides for a more honest/depth understand. The vision of the ecommerce system is to be able to provide a smooth and user-friendly platform for customers to select from a wide range of products conveniently and purchased and to cater to the needs of both customers and inventory manager.

### Shopping Cart Functionality

Develop a shopping cart feature that enables users to add products, modify quantities, and proceed to checkout. This system provides an avenue for customers to shop from a wide variety of products online.

### Checkout and Payment

Implement a secure and user-friendly checkout process, allowing customers to provide delivery addresses and make payments online or through cash on delivery (COD).

### Order Tracking

Develop a real-time order tracking system that provides updates to customers regarding the status and delivery date of their orders.

### Account Management

Include functionality for users to manage their profiles, view order history, and access invoices.

### Notification System

Implement a notification system to send timely updates to customers, such as order confirmations, payment notifications, and delivery status updates.

### Integration with Supply Chain

Integrate the platform with the existing delivery and supply chain management systems to facilitate order processing and fulfilment. It also provides a platform where they can update their inventory to the system for customers to view and purchase. The biggest advantage of the service is the comfort it brings with remote usage.

### Performance and Scalability

Design and optimize the architecture to ensure the platform can handle concurrent user traffic, a growing product catalogue, and deliver responsive user experiences.

### Security and Privacy

Implement security measures, including secure payment processing, user data protection, and adherence to data privacy regulations.

### Localization

Support multiple languages and regional preferences to cater to users from different regions and enhance their shopping experience.

## Intended Audience

### Business Owners and Management

The project stakeholders who have initiated the project and hold the responsibility for its success. They would be interested in understanding how the platform will align with the business goals and contribute to its growth.

### Product Managers

Individuals responsible for overseeing the development and delivery of the e-commerce platform. They need to be familiar with the project scope, requirements, and progress to ensure that the platform meets the desired functionality and objectives.

### Software Architects

Architects who are responsible for designing the system's architecture and ensuring its scalability, security, and performance. They would be interested in the architectural aspects and decisions related to the platform development.

### Developers and Engineers

The development team involved in building the e-commerce platform. They need to have a clear understanding of the project scope to guide their coding, implementation, and testing activities.

### Quality Assurance/Testers

Individuals responsible for testing the platform to ensure its functionality, usability, and performance meet the defined requirements. They need to be aware of the project scope to design effective test cases and validate the platform against the specified criteria.

### User Experience (UX) Designers

Designers responsible for creating an intuitive and user-friendly interface. They should be aware of the project scope to align the user experience with the intended goals and requirements.

## Business Context

The business context for the e-commerce platform development project can be described as follows:

### Business Overview

The company is a new start-up operating in the Indian region, primarily engaged in selling furniture and appliances through offline channels. They have an established delivery and supply chain infrastructure in place.

### Expansion into Online Space

The company aims to expand its operations in the online space by developing an e-commerce platform. The platform will allow customers to purchase a variety of products, including furniture and appliances, online.

### Objective

The primary objective is to establish a user-friendly e-commerce platform that enables customers to browse, select, and purchase products online, leveraging the company's existing delivery and supply chain capabilities.

### Reducing Dependence on Offline Channels

The company seeks to decrease reliance on traditional offline channels and reduce the costs associated with commission fees charged by existing marketplaces.

### Simplicity and Timely Deliveries

The company wants to avoid overloading the platform with excessive features such as wallets and coupons to ensure a streamlined user experience and prevent any delays in order processing and delivery.

### Target Customer Base

The platform will cater to customers residing in Tier 2 and Tier 3 cities in India. The availability of delivery services in specific pin codes will be communicated to users as necessary.

Overall, the business context highlights the company's desire to establish an independent e-commerce platform that enables customers to conveniently purchase furniture and appliances online. The emphasis is on simplicity, timely deliveries, and leveraging the existing delivery and supply chain infrastructure to expand the company's reach in the online market.

### Stakeholders

The stakeholders for the e-commerce platform development project can include:

#### Business Owners and Management

The individuals who have a vested interest in the success of the project, as they are responsible for the overall business strategy, investment decisions, and project governance.

#### Product Managers

Individuals who are accountable for overseeing the development of the e-commerce platform, aligning it with business goals, and ensuring it meets the needs of the target market.

#### Software Architects

Architects responsible for designing the system's architecture, making technology decisions, and ensuring scalability, performance, and security of the platform.

#### Developers and Engineers

The development team tasked with building the e-commerce platform, including front-end and back-end developers, who will be responsible for coding, testing, and implementing the required functionality.

#### Quality Assurance/Testers

Individuals responsible for testing the e-commerce platform to ensure it meets quality standards, functions correctly, and provides a seamless user experience.

#### User Experience (UX) Designers

Designers who are involved in creating an intuitive and user-friendly interface for the e-commerce platform, focusing on usability, accessibility, and visual design.

#### Supply Chain and Logistics Managers

Individuals responsible for managing the company's supply chain and logistics operations. They will be involved in integrating the e-commerce platform with existing systems to ensure smooth order processing and timely deliveries.

#### Customer Support Representatives

Representatives who will handle customer inquiries, order issues, and provide support through various channels such as chat, email, or phone.

#### End Users

The ultimate users of the e-commerce platform—customers who will visit the website, browse products, make purchases, and interact with the platform. Their feedback and satisfaction are essential for the success of the project.

It is important to identify and engage these stakeholders throughout the project to gather requirements, ensure alignment with business objectives, obtain feedback, and address any concerns or issues that may arise during the development process.

## Technical Constraints

The technical constraints for the e-commerce platform development project can include:

#### Technology Stack

The project may have constraints related to the selection of technology stack or platforms to be used for development. These constraints could be influenced by existing infrastructure, organizational preferences, compatibility requirements, or third-party integrations.

#### Scalability

The platform should be designed to handle increasing user traffic and a growing product catalogue. Scalability constraints may include considerations for database performance, server load balancing, caching mechanisms, and horizontal or vertical scaling options.

#### Performance

The platform should be designed to deliver optimal performance to ensure fast response times, quick page load times, and smooth user interactions. Constraints related to performance may involve optimizing code, database queries, network latency, and minimizing resource usage.

#### Security

The e-commerce platform must adhere to industry-standard security practices to protect user data, prevent unauthorized access, and safeguard against potential threats such as data breaches or injection attacks. Constraints may include using encryption, secure authentication mechanisms, and implementing security protocols.

#### Integration with Third-Party Systems

The platform may need to integrate with external systems such as payment gateways, inventory management systems, or customer relationship management (CRM) software. Constraints can arise from compatibility requirements, API availability, data synchronization, and maintaining data integrity during integration.

#### Mobile Responsiveness

The e-commerce platform may require mobile responsiveness to ensure a seamless user experience across various devices and screen sizes. Constraints may include designing responsive layouts, optimizing images and media, and ensuring compatibility with different mobile platforms and browsers.

#### Data Management

Constraints related to data management may include data storage, retrieval, and backup mechanisms. Compliance with data privacy regulations, data encryption, and proper handling of personally identifiable information (PII) may also impose constraints.

#### Cross-Browser Compatibility

The e-commerce platform should be compatible with popular web browsers to ensure consistent functionality and appearance across different browser versions and platforms. Constraints may involve testing and resolving any compatibility issues that arise.

#### Localization

If the platform is intended to serve customers from different regions or countries, constraints related to localization may include support for multiple languages, currencies, and regional preferences.

#### Development Time and Resource Constraints

The project may have limitations on development time, available resources, or budget. These constraints can influence the scope, prioritization of features, and overall development timeline.

## Business Constraints

The business constraints for the e-commerce platform development project can include:

#### Budget

The availability of financial resources allocated to the project can impose constraints on the scope, timeline, and resources available for development.

#### Timeframe

The project may have specific time constraints, such as a desired launch date or market opportunity window, which can impact the development timeline and prioritization of features.

#### Resource Availability

Constraints related to the availability of skilled personnel, development teams, or external vendors can impact the project's execution and may require adjustments to the scope or timeline.

#### Legal and Regulatory Compliance

The e-commerce platform must adhere to applicable laws, regulations, and industry standards, such as consumer protection, data privacy, and taxation requirements. Compliance constraints may influence the design, functionality, and operational aspects of the platform.

#### Competitive Landscape

The business may face constraints imposed by the competitive market landscape. These constraints can include the need to differentiate from competitors, offer unique features or pricing strategies, and respond to market trends or customer demands.

#### Existing Infrastructure

Constraints can arise from the need to integrate the e-commerce platform with existing systems, databases, or infrastructure. Compatibility and interoperability considerations may need to be addressed to ensure a seamless transition and minimize disruptions to the business operations.

#### Customer Expectations

The platform should meet the expectations of the target customers in terms of usability, convenience, and functionality. Understanding customer needs and aligning the platform with their expectations may impose constraints on the design and development process.

#### Brand Identity

The e-commerce platform should align with the brand identity, values, and positioning of the business. Constraints related to branding guidelines, visual design, and user experience may need to be considered during the development process.

#### Operational Considerations

Constraints related to operational aspects, such as inventory management, order fulfilment, customer support, and supply chain logistics, may need to be taken into account to ensure the platform can effectively support the business operations.

#### Vendor or Partner Constraints

If the business relies on third-party vendors or partners for specific services, constraints may arise from their capabilities, availability, pricing models, or contractual agreements.

Understanding and addressing these business constraints is crucial for successfully delivering an e-commerce platform that aligns with the business goals, meets customer expectations, and operates within the defined limitations.

## Quality attribute requirements

Quality attribute requirements, also known as non-functional requirements, describe the desired qualities and characteristics of the e-commerce platform beyond its core functionality. These requirements are essential for ensuring the platform's performance, usability, security, and other key aspects meet user expectations and business needs. Here are some examples of quality attribute requirements for the e-commerce platform:

#### Performance

The platform should have fast response times and quick page load speeds to provide a seamless user experience. For example, pages should load within a certain time limit (e.g., 2 seconds) across various devices and network conditions.

#### Scalability

The platform should be designed to handle increasing user traffic and a growing number of products without compromising performance. It should support horizontal and vertical scaling to accommodate future growth.

#### Reliability

The platform should be highly reliable, with minimal downtime and system failures. It should have a robust error-handling mechanism and graceful degradation in case of unexpected issues.

#### Security

The platform should adhere to industry-standard security practices to protect user data, prevent unauthorized access, and safeguard against potential threats such as data breaches and cyberattacks.

#### Usability

The platform should be intuitive and easy to use, ensuring that users can navigate through the website, search for products, and make purchases without confusion or frustration.

#### Accessibility

The platform should be accessible to users with disabilities, conforming to accessibility standards such as WCAG (Web Content Accessibility Guidelines).

#### Compatibility

The platform should be compatible with different web browsers, operating systems, and devices to provide a consistent experience to all users.

#### Maintainability

The platform should be designed and implemented in a way that facilitates easy maintenance, updates, and enhancements to accommodate future changes or new features.

#### Data Integrity and Privacy

The platform should ensure the integrity of user data, prevent data corruption, and protect user privacy by adhering to data privacy regulations and best practices.

#### Performance under Load

The platform should be tested for its ability to handle peak user loads without performance degradation. Load testing should be performed to assess its performance and scalability.

#### Response Time for Payment Processing

The payment processing system should have a defined response time for completing transactions, ensuring quick and efficient payment processing for customers.

#### Order Fulfilment Time

The time between order placement and delivery should be specified to manage customer expectations and ensure timely deliveries.

#### Customer Support Response Time

The expected response time for customer support inquiries and issue resolution should be defined to provide excellent customer service.

Each of these quality attribute requirements plays a critical role in shaping the overall user experience and performance of the e-commerce platform. By defining and adhering to these requirements, the platform can meet user expectations, gain customer trust, and contribute to the success of the business.

## Use Case

User Registration

*Scenario of Registration Process*

* User can enter FirstName, Middle Name, Last Name, Email Id, Phone Number, Password.
* User can click on registration Button.
* Validation can be done at front end and backend code.
* If validation successfully done, then user can see registration success and login page link/button.
* If validation failed, then user can see validation error message on registration page.

A picture containing text, screenshot, font, diagram

Description automatically generated

User Login

*Scenario of Login Process*

* User can enter Username and password, username can be

email Id or Phone Number.

* If user enter valid details, then he can see home page of ecommerce application.
* If validation failed, user could see login failed error on Login page.

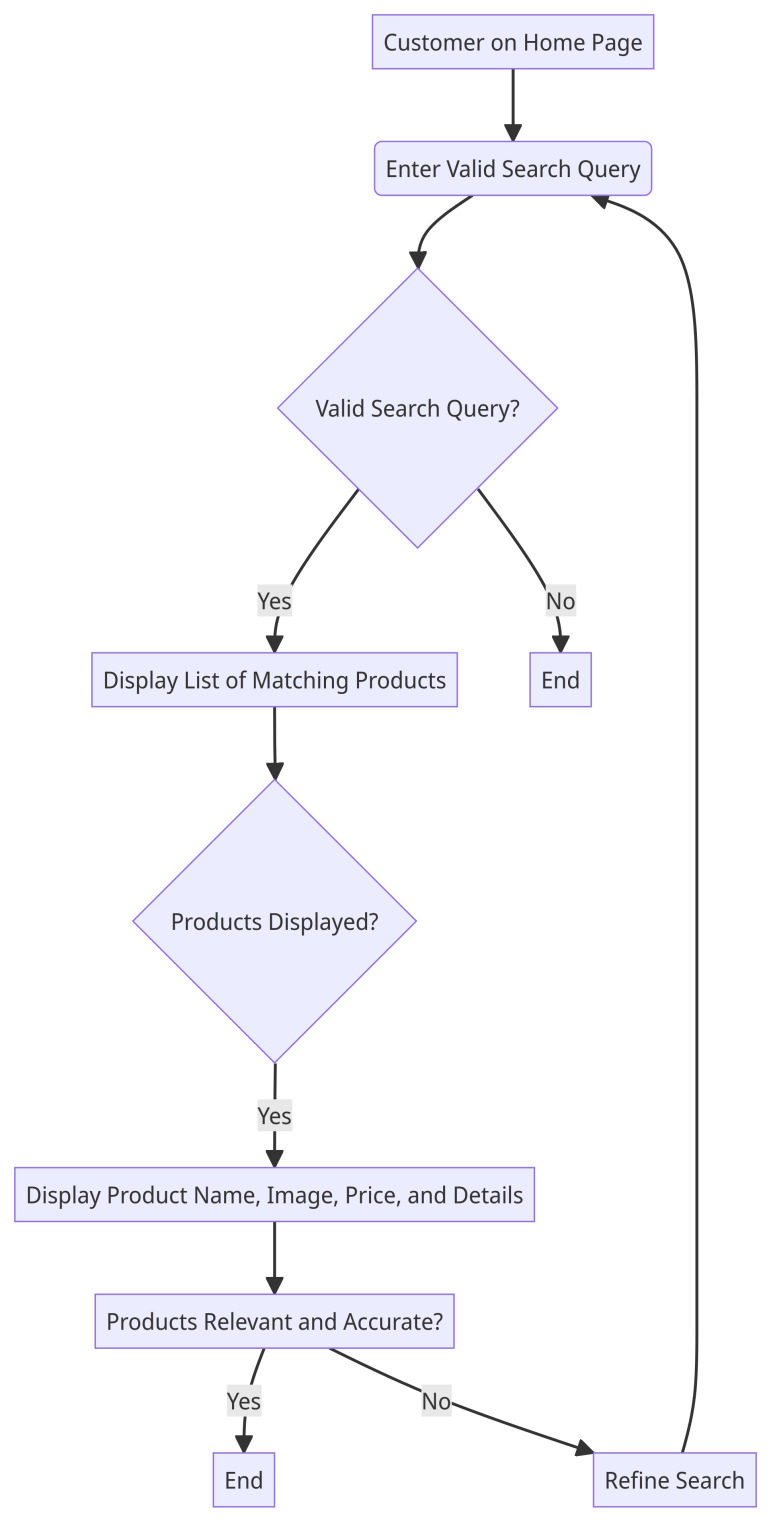
A screenshot of a computer

Description automatically generated with medium confidence

### User searches for products

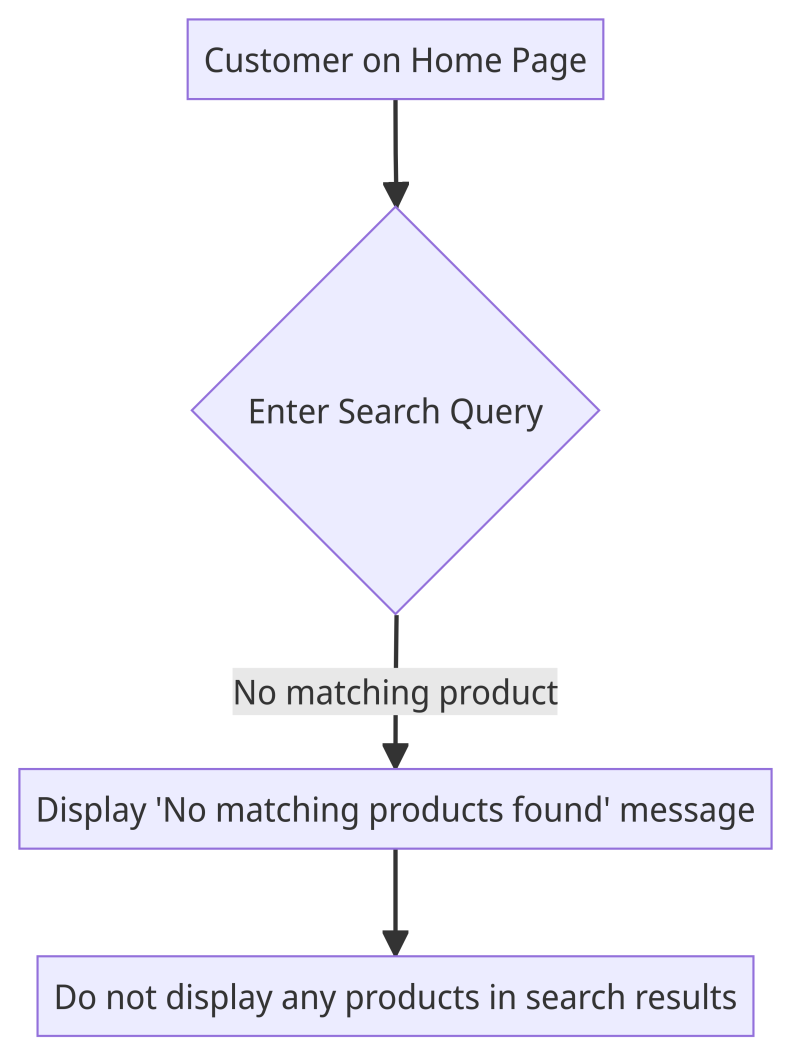
#### Scenario: Successful product search

* Given that the customer is on the home page
* When the customer enters a valid search query for a product
* Then the system should display a list of products that match the search query
* And the displayed products should include the product name, image, price, and relevant details
* And the displayed products should be relevant and accurately match the search query



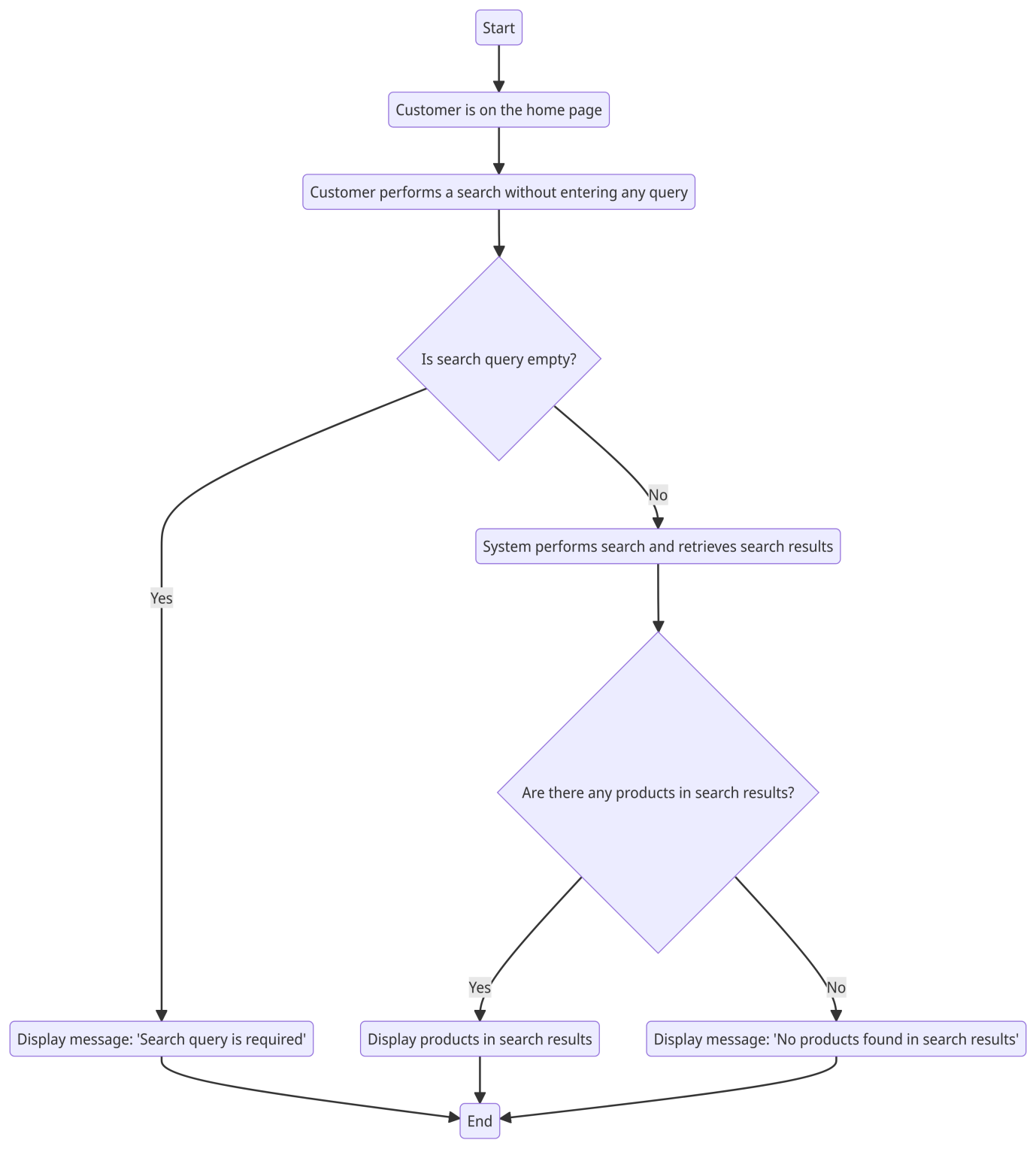
#### Scenario: No matching results for the search query

* Given that the customer is on the home page
* When the customer enters a search query for a product that does not exist in the system
* Then the system should display a message indicating that no matching products were found
* And the system should not display any products in the search results



#### Scenario: Handling empty search query

* Given that the customer is on the home page
* When the customer performs a search without entering any search query
* Then the system should display a message indicating that a search query is required
* And the system should not perform a search or display any products in the search results



#### Scenario: Search results pagination

* Given that the customer is on the home page
* When the customer performs a search that returns a large number of results
* Then the system should display the search results in a paginated manner, showing a limited number of products per page
* And the system should provide navigation options for the customer to view additional pages of search results

#### Scenario: Search performance and response time

* Given that the customer is on the home page
* When the customer performs a search
* Then the system should provide search results within an acceptable response time, typically within a few seconds
* And the system should be able to handle concurrent searches and maintain performance under expected load

### User adds products to the cart

#### Scenario: Successful addition of a product to the cart

* Given that the customer is viewing a product
* When the customer selects the option to add the product to the cart
* Then the system should update the cart to include the selected product
* And the system should display a confirmation message indicating that the product has been successfully added to the cart

#### Scenario: Incrementing the quantity of an existing product in the cart

* Given that the customer has already added a product to the cart
* When the customer adds the same product again
* Then the system should increment the quantity of the product in the cart
* And the system should update the total price in the cart to reflect the increased quantity

#### Scenario: Displaying the updated cart summary

* Given that the customer has added one or more products to the cart
* When the customer views the cart summary
* Then the system should display the list of products in the cart
* And the system should show the individual prices, quantities, and total price for each product in the cart
* And the system should provide options for the customer to modify or remove items from the cart

#### Scenario: Handling out-of-stock products

* Given that the customer tries to add a product that is currently out of stock
* When the customer attempts to add the out-of-stock product to the cart
* Then the system should display a message indicating that the product is currently unavailable
* And the system should not add the out-of-stock product to the cart

#### Scenario: Persisting the cart across sessions

* Given that the customer has added products to the cart
* When the customer logs out and logs back in
* Then the system should retain the products in the cart and restore the cart contents for the logged-in customer

### User checks out and places an order

#### 1. Scenario: Successful checkout and order placement

Given that the customer has products in the cart

When the customer proceeds to the checkout process

Then the system should prompt the customer to provide the required delivery address

And the system should provide options for the customer to select a preferred payment method

And the system should validate and process the payment successfully

And the system should generate a unique order ID for the placed order

And the system should display a confirmation message indicating that the order has been successfully placed

#### 2. Scenario: Handling empty cart during checkout

Given that the customer has not added any products to the cart

When the customer proceeds to the checkout process

Then the system should display a message indicating that the cart is empty

And the system should not allow the customer to proceed with the checkout process

#### 3. Scenario: Required delivery address validation

Given that the customer is on the checkout page

When the customer provides the required delivery address information

Then the system should validate the provided address for completeness and correctness

And the system should display error messages or warnings if the address is incomplete or invalid

And the system should not allow the customer to proceed with the checkout process until a valid address is provided

#### 4. Scenario: Payment processing and confirmation

Given that the customer has provided the required delivery address

When the customer selects a payment method and initiates the payment process

Then the system should securely process the payment transaction

And the system should display a payment confirmation message or receipt to the customer

And the system should send a notification to the customer regarding the payment status

#### 5. Scenario: Order confirmation and tracking

Given that the customer has successfully placed an order

When the order is confirmed by the system

Then the system should display the order details, including the order ID, items purchased, total amount, and delivery information

And the system should send a confirmation email or notification to the customer with the order details and estimated delivery date

And the system should provide a tracking mechanism for the customer to track the status of the placed order

### User tracks an order

#### 1. Scenario: Successful order tracking

* Given that the customer has placed an order and received an order ID or tracking number
* When the customer navigates to the order tracking page
* Then the system should provide a field for the customer to enter the order ID or tracking number
* And the system should display the current status and location of the order
* And the system should provide estimated delivery dates or timeframes for each stage of the delivery process
* And the system should update the tracking information in real-time or with a reasonable delay

#### 2. Scenario: Invalid or nonexistent order tracking

Given that the customer enters an invalid or nonexistent order ID or tracking number

When the customer submits the tracking request

Then the system should display a message indicating that the entered order ID or tracking number is not valid or does not exist

And the system should not display any tracking information for the invalid order

#### 3. Scenario: Tracking information availability

Given that the customer has placed an order recently

When the customer tries to track the order immediately after placing it

Then the system should display a message indicating that the tracking information is not available yet

And the system should provide an estimated timeframe or expected time for the tracking information to become available

#### 4. Scenario: Detailed order status and history

Given that the customer is tracking an order

When the customer views the order tracking details

Then the system should display a comprehensive view of the order status, including the current location, transit history, and expected delivery date

And the system should provide any relevant updates or notifications regarding the order, such as delays or changes in the delivery schedule

#### 5. Scenario: Multiple order tracking

Given that the customer has placed multiple orders

When the customer wants to track multiple orders simultaneously

Then the system should provide a mechanism for the customer to enter multiple order IDs or tracking numbers at once

And the system should display the tracking information for all the entered orders in a consolidated view

### User views order history

#### 1. Scenario: Successful access to order history

Given that the customer is logged into their account

When the customer navigates to the order history page

Then the system should display a list of previous orders placed by the customer

And the system should show the order details, including the order ID, items purchased, total amount, and delivery information

And the system should display the orders in reverse chronological order, with the most recent order appearing at the top

#### 2. Scenario: Filtering or sorting order history

Given that the customer is viewing the order history page

When the customer wants to filter or sort the order history

Then the system should provide options for the customer to filter or sort the orders based on criteria such as date, status, or product category

And the system should update the order history view accordingly, displaying only the orders that match the selected filter or sorting criteria

#### 3. Scenario: Order details and receipts

Given that the customer is viewing the order history

When the customer selects a specific order from the order history list

Then the system should display the detailed order information, including the order ID, items purchased, individual prices, quantities, and total amount

And the system should provide the option to view or download an order receipt or invoice in a printable format

#### 4. Scenario: Order status and tracking from history

Given that the customer is viewing the order history

When the customer selects a specific order from the order history list

Then the system should display the current status of the order, such as "Processing," "Shipped," or "Delivered"

And the system should provide a link or button for the customer to track the order or access the order tracking details directly from the order history page

#### 5. Scenario: Pagination and navigation of order history

Given that the customer has a large number of orders in their order history

When the customer views the order history page

Then the system should display the orders in a paginated manner, showing a limited number of orders per page

And the system should provide navigation options for the customer to move between pages of the order history

### User views invoice

#### 1. Scenario: Successful access to invoice

Given that the customer is logged into their account

When the customer navigates to the invoice page or selects a specific order

Then the system should display the invoice for the selected order

And the system should show the invoice details, including the order ID, items purchased, individual prices, quantities, and total amount

And the system should include relevant information such as customer details, billing address, payment method, and any applicable taxes or discounts

#### 2. Scenario: Printable and downloadable invoice

Given that the customer is viewing the invoice

When the customer wants to print or download the invoice

Then the system should provide a printable or downloadable version of the invoice in a suitable format (such as PDF or HTML)

And the system should ensure that the printable or downloadable invoice accurately represents the information displayed on the screen

#### 3. Scenario: Invoice layout and formatting

Given that the customer is viewing the invoice

When the customer reviews the invoice layout and formatting

Then the system should present the invoice in a clear and organized manner

And the system should use appropriate formatting, such as headings, subtotals, and a total amount, to facilitate easy comprehension of the invoice details

#### 4. Scenario: Handling invoice errors or discrepancies

Given that the customer notices errors or discrepancies in the displayed invoice

When the customer raises a concern or contacts customer support

Then the system should promptly investigate and resolve the issue

And the system should provide clear communication to the customer regarding the resolution and any necessary adjustments to the invoice

#### 5. Scenario: Accessing historical invoices

Given that the customer wants to view previous invoices

When the customer navigates to the invoice history or order history page

Then the system should provide a list of previous orders with links to access the respective invoices

And the system should ensure that the invoices remain accessible even after a certain period of time or order completion

### Administrator manages products

#### 1. Scenario: Adding a new product

Given that the Administrator is logged into their account

When the Administrator navigates to the product management section

Then the system should provide a form or interface to add a new product

And the system should validate and require essential information such as product name, description, price, and category

And the system should save the product details and assign a unique identifier or SKU

#### 2. Scenario: Updating an existing product

Given that the Administrator is logged into their account

When the Administrator selects a specific product from the product list

Then the system should display the product details for editing

And the system should allow the Administrator to modify the product information, such as name, description, price, and category

And the system should save the updated product details and reflect the changes in the product list

#### 3. Scenario: Removing a product

Given that the Administrator is logged into their account

When the Administrator selects a specific product from the product list

Then the system should provide an option or button to remove the product

And the system should prompt the Administrator for confirmation before permanently deleting the product

And the system should remove the product from the product list and any associated data or references

#### 4. Scenario: Managing product categories

Given that the Administrator is logged into their account

When the Administrator navigates to the product category management section

Then the system should display the list of existing categories

And the system should allow the Administrator to add new categories or modify existing ones

And the system should ensure that categories are properly organized and associated with the relevant products

#### 5. Scenario: Handling product images

Given that the Administrator is managing a product

When the Administrator uploads or modifies product images

Then the system should accept image files of specified formats and sizes

And the system should store and associate the images with the respective product

And the system should display the product images appropriately in the product listing and details pages

### Administrator manages orders

#### 1. Scenario: Viewing order list

Given that the Administrator is logged into their account

When the Administrator navigates to the order management section

Then the system should display a list of orders, including relevant details such as order ID, customer name, order status, and total amount

And the system should provide pagination or scrolling functionality if the order list is extensive

#### 2. Scenario: Filtering and sorting orders

Given that the Administrator is viewing the order list

When the Administrator wants to filter or sort the orders

Then the system should provide options to filter or sort the orders based on criteria such as order status, customer name, order date, or total amount

And the system should update the order list according to the selected filter or sorting criteria

#### 3. Scenario: Updating order status

Given that the Administrator is viewing an order in the order list

When the Administrator wants to update the order status

Then the system should provide options to change the order status, such as marking it as "Processing," "Shipped," or "Delivered"

And the system should save and reflect the updated order status in the order list and order details

#### 4. Scenario: Viewing order details

Given that the Administrator is viewing an order in the order list

When the Administrator selects a specific order

Then the system should display the order details, including customer information, product details, quantities, prices, and any applied discounts or taxes

And the system should provide a summary of the order, including the total amount and payment status

#### 5. Scenario: Managing order fulfillment

Given that the Administrator is viewing an order in the order list

When the Administrator wants to manage order fulfillment, such as assigning it to a delivery person or updating delivery details

Then the system should provide options or fields to input and save the relevant information

And the system should ensure that the updated fulfillment details are reflected in the order details and any relevant notifications or updates are sent to the customer

#### 6. Scenario: Generating order reports

Given that the Administrator wants to generate order reports

When the Administrator navigates to the reporting section or selects a specific reporting option

Then the system should generate and display relevant order reports, such as sales reports, order volume reports, or revenue reports

And the system should provide options to filter the reports based on criteria such as date range, product category, or customer segment

### Supplier manages inventory

#### 1. Scenario: Viewing inventory

Given that the Supplier is logged into their account

When the Supplier navigates to the inventory management section

Then the system should display a list of available inventory items

And the system should show relevant details for each item, such as product name, SKU, quantity in stock, and location

#### 2. Scenario: Updating inventory quantities

Given that the Supplier is viewing the inventory

When the Supplier wants to update the quantity of a specific item

Then the system should provide options to adjust the quantity, such as adding or subtracting units

And the system should validate and update the inventory quantity accordingly

And the system should reflect the updated quantity in the inventory list and any associated reports or notifications

#### 3. Scenario: Adding new inventory items

Given that the Supplier is logged into their account

When the Supplier wants to add a new item to the inventory

Then the system should provide a form or interface to enter the details of the new item, such as product name, SKU, quantity, location, and other relevant attributes

And the system should validate the entered information and save the new item in the inventory

#### 4. Scenario: Removing inventory items

Given that the Supplier is viewing the inventory

When the Supplier wants to remove a specific item from the inventory

Then the system should provide options to delete the item

And the system should prompt the Supplier for confirmation before permanently removing the item from the inventory

And the system should update the inventory list and any associated records or reports to reflect the removal of the item

#### 5. Scenario: Tracking inventory movement

Given that the Supplier wants to track the movement of inventory items

When the Supplier selects a specific item from the inventory list

Then the system should display a history of inventory transactions for that item, including details such as date, quantity, and location changes

And the system should provide options to filter and search the inventory movement history based on criteria such as date range or location

#### 6. Scenario: Managing low stock or out-of-stock items

Given that the Supplier is viewing the inventory

When the Supplier wants to identify low stock or out-of-stock items

Then the system should highlight or provide filters to easily identify items with low quantities or no stock

And the system should allow the Supplier to set alerts or notifications for low stock levels to proactively manage inventory replenishment

### 

### Supplier manages product catalogue

#### 1. Scenario: Viewing the product catalogue

Given that the Supplier is logged into their account

When the Supplier navigates to the product catalogue management section

Then the system should display a list of products in the catalogue

And the system should show relevant details for each product, such as product name, SKU, description, and pricing information

#### 2. Scenario: Adding a new product to the catalogue

Given that the Supplier is logged into their account

When the Supplier wants to add a new product to the catalogue

Then the system should provide a form or interface to enter the details of the new product, such as product name, SKU, description, category, pricing, and any other relevant attributes

And the system should validate the entered information and save the new product in the catalogue

#### 3. Scenario: Updating an existing product in the catalogue

Given that the Supplier is viewing the product catalogue

When the Supplier selects a specific product from the catalogue

Then the system should display the product details for editing

And the system should allow the Supplier to modify the product information, such as name, description, category, pricing, or any other relevant attributes

And the system should save the updated product details and reflect the changes in the product catalogue

#### 4. Scenario: Removing a product from the catalogue

Given that the Supplier is viewing the product catalogue

When the Supplier wants to remove a specific product from the catalogue

Then the system should provide options to delete the product

And the system should prompt the Supplier for confirmation before permanently removing the product from the catalogue

And the system should update the product catalogue and any associated records or reports to reflect the removal of the product

#### 5. Scenario: Managing product images in the catalogue

Given that the Supplier is managing a product in the catalogue

When the Supplier wants to upload or modify product images

Then the system should accept image files of specified formats and sizes

And the system should store and associate the images with the respective product in the catalogue

And the system should display the product images appropriately in the catalogue listing and details pages

#### 6. Scenario: Managing product categories

Given that the Supplier is managing a product in the catalogue

When the Supplier wants to assign or update the product category

Then the system should provide options to select the appropriate category from a predefined list

And the system should ensure that the product is properly categorized in the catalogue for easy navigation and search

# Repository branching strategy:

We will maintain our repository in combination of the following strategies:

### Feature Branching:

* Each feature or user story is developed in a dedicated branch.
* Developers create a new branch for each feature, work on it independently, and merge it back to the main branch once complete.
* Isolation of features allows for parallel development and testing.
* Pull requests or code reviews are typically used to ensure quality before merging.
* All feature branching will follow the below naming convention:
  + feature/<issue-number>-<issue-title>
  + e.g. feature/01-basic-skeleton-preperation

### Gitflow:

* Gitflow is a branching model that utilizes two main branches: "main" and "develop."
* The "develop" branch serves as the main development branch, where ongoing work and new features are integrated.
* The "main" branch represents the stable production-ready codebase.
* Feature branches are created from the "develop" branch and merged back into it.
* When a release is ready, a release branch is created from the "develop" branch and merged into both "develop" and "main" after testing.

### Release Branching:

* A separate branch is created for each release version.
* Bug fixes and maintenance work for a specific release are performed in the respective release branch.
* The main development continues on a separate branch.
* This strategy allows for isolated bug fixing in released versions while ongoing development progresses independently.
  + release-<year>-<month>-<date>.<major version>.<minor version>.<patches version>
  + e.g. release-2023-07-03.0.0.0

# Non Functional Requirements

### Performance:

* The system should provide fast and responsive search functionality, returning search results within a few seconds. This means optimizing the search algorithm, database queries, and indexing mechanisms to ensure quick response times.
* The system should be able to handle concurrent searches and maintain performance under expected load. This involves load testing the system to determine its maximum capacity and ensuring that it can handle multiple simultaneous searches without significant degradation in performance.
* The system should have efficient pagination and navigation options for large search result sets and order history. This includes implementing pagination techniques to divide search results into manageable chunks, as well as providing user-friendly navigation options to allow users to easily browse through large sets of data.

### Scalability:

* The system should be scalable to handle increasing numbers of products, users, and concurrent transactions. This involves designing the system in a way that allows for horizontal scalability, such as using distributed computing or containerization techniques, so that additional resources can be added to handle increased demand.
* The architecture should support horizontal scaling, allowing for the addition of more servers or resources to handle increased demand. This means designing the system with components that can be easily replicated or distributed across multiple servers, and implementing load balancing mechanisms to distribute the workload evenly.

### Reliability:

* The system should have a high level of availability and minimize downtime to ensure uninterrupted service for users. This requires implementing fault-tolerant mechanisms, such as redundant servers, automatic failover, and backup systems, to ensure that the system remains operational even in the event of hardware or software failures.
* The system should handle errors and exceptions gracefully, providing informative error messages and recovering from failures without data loss. This involves implementing error handling and logging mechanisms to capture and report errors, as well as implementing appropriate backup and recovery strategies to minimize data loss in case of failures.

### Security:

* The system should ensure the privacy and confidentiality of user data, including personal information and payment details. This includes implementing secure storage and transmission mechanisms, such as encryption, to protect sensitive data from unauthorized access.
* The system should implement proper authentication and authorization mechanisms to prevent unauthorized access to sensitive data or administrative functions. This involves implementing secure user authentication processes, such as strong password policies, and restricting access to sensitive operations or data based on user roles and permissions.
* The system should use encryption and secure communication protocols for data transmission, such as HTTPS, to protect data from interception or tampering during transit.
* The system should protect against common security threats, such as cross-site scripting (XSS) and SQL injection attacks, by implementing input validation and sanitization techniques, as well as using parameterized queries or prepared statements to prevent SQL injection vulnerabilities.

### Usability:

* The system should have an intuitive and user-friendly interface, making it easy for customers to search for products, add items to the cart, and complete the checkout process. This involves conducting user research and usability testing to ensure that the interface is designed in a way that meets user expectations and minimizes friction in the shopping process.
* The system should provide clear and informative error messages to assist users in resolving any issues that may arise. This includes providing specific error messages that clearly explain the problem and suggest possible solutions, helping users understand and address errors they encounter.
* The system should be responsive and accessible across different devices and screen sizes. This involves employing responsive design techniques and ensuring compatibility with various web browsers and mobile devices, so that users can access and use the system seamlessly from different platforms.

### Maintainability:

* The system should have modular and well-structured code, making it easier to maintain and enhance in the future. This includes following software development
* best practices, such as modular design principles and code documentation, to ensure that the codebase is clean, organized, and easily understandable.
* The system should adhere to coding standards and best practices, promoting readability and maintainability. This involves following established coding conventions and guidelines, such as using meaningful variable names, properly documenting code, and applying appropriate design patterns and architectural principles.
* The system should have proper documentation, including system architecture, APIs, and user guides, to assist with future development and troubleshooting. This includes providing comprehensive documentation that describes the system's structure, components, interfaces, and functionality, as well as user guides that explain how to use the system effectively.

### Integration:

* The system should have the ability to integrate with external systems or services, such as payment gateways, shipping providers, and inventory management systems. This involves designing the system with extensibility in mind, using standard integration protocols and formats, and providing APIs or web hooks for seamless data exchange with other systems.
* The system should support standard integration protocols and formats, facilitating seamless data exchange with other systems. This includes supporting commonly used protocols and formats, such as RESTful APIs, JSON, XML, or SOAP, to enable interoperability with external systems.

### Compliance:

* The system should comply with relevant legal and regulatory requirements, such as data protection laws and consumer protection regulations. This involves understanding and adhering to applicable laws and regulations, such as GDPR (General Data Protection Regulation), PCI DSS (Payment Card Industry Data Security Standard), and local consumer protection laws, to ensure the system's compliance.
* The system should adhere to industry standards and best practices for e-commerce systems, ensuring compliance with security and privacy standards. This includes following established guidelines and frameworks, such as OWASP (Open Web Application Security Project) recommendations, for securing web applications and protecting against common vulnerabilities and attacks.

# Required Questionnaire for non-functional requirements:

### Performance:

* What are the acceptable response times for critical operations? Provide specific time thresholds if available.
* How should the system handle peak loads or high traffic situations?
* Are there any specific performance metrics or benchmarks that need to be achieved (e.g., maximum latency, minimum throughput)?
* Are there any specific performance testing requirements or scenarios that should be considered?

### Scalability:

* What are the expected growth projections for the system in terms of data volume, user base, or transactions?
* How should the system handle increases in data volume, users, or transactions? Should it scale vertically (adding more resources to the existing system) or horizontally (distributing the load across multiple instances)?
* Are there any specific scalability targets or constraints that need to be met?
* Are there any expectations or requirements for automatic scaling based on demand?

### Reliability:

* What is the required uptime or availability of the system? Are there any specific service level agreements (SLAs) that need to be met?
* How should the system handle failures, errors, or exceptions? Should it have mechanisms for fault tolerance, redundancy, or failover?
* Are there any specific backup and recovery requirements, such as data backups, disaster recovery plans, or replication strategies?
* Is there a need for real-time monitoring or alerting mechanisms to ensure system reliability?

### Security:

* What are the specific security requirements for the system, such as authentication, authorization, and data encryption?
* Are there any regulatory compliance standards that the system must adhere to (e.g., GDPR, HIPAA, PCI-DSS)?
* Are there any specific access control policies or restrictions for different user roles or sensitive data?
* Are there any requirements for auditing, logging, or tracking user actions within the system?

### Usability:

* What are the usability expectations for the system's user interface? Is there a need for a user-friendly and intuitive design?
* Are there any specific accessibility requirements to accommodate users with disabilities?
* Are there any compatibility requirements, such as support for different browsers, devices, or screen sizes?
* Should the system provide helpful and informative error messages to guide users in troubleshooting?

### Maintainability:

* Are there any specific coding standards, guidelines, or best practices that developers should follow?
* Is there a need for well-documented code, including inline comments, code samples, or external documentation?
* Are there any version control or code review requirements?
* Should the system be designed for easy troubleshooting, debugging, and maintenance?

### Integration:

* Are there any external systems, databases, or services that the system needs to integrate with? Specify the nature and purpose of these integrations.
* Are there any specific protocols, APIs, or data exchange formats that the system should support?
* Are there any authentication or authorization requirements for accessing external systems or services?
* Are there any limitations or constraints regarding data privacy or security when integrating with external systems?

### Compliance:

* Are there any specific legal, regulatory, or industry standards that the system must comply with (e.g., data protection, financial regulations, healthcare regulations)?
* Are there any data retention or archival requirements that need to be considered?
* Are there any specific privacy or consent requirements for handling user data?
* Are there any limitations or restrictions on the geographic location or hosting of the system?

Enlisted below are all the major functions supported by the online shopping system along with the user classes.

● Register: for customers

● Login: for customers

● Logout: for customers

● View Account Details: for customers

● Edit Account Details: for customers

● Search item: for customers

● View item: for customers

● Add item to cart: for customers

● View shopping cart: for customers

● Change items in cart: for customers

● Proceed to buy : for customers

● Delivery & payment: for customers

● Place order: for customers

● Track order: for customers

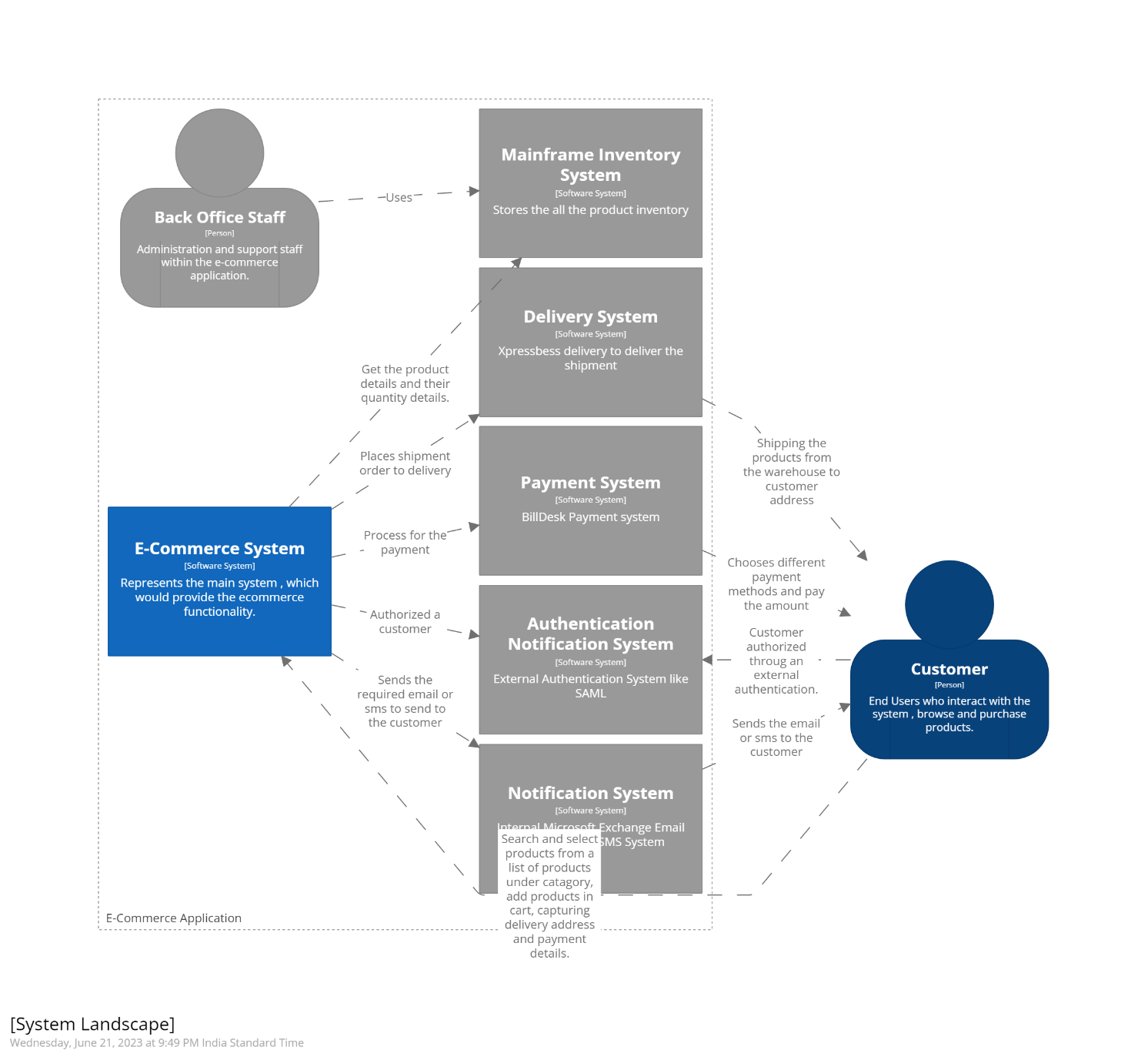
● Cancel order: for customers

● Return item: for customers

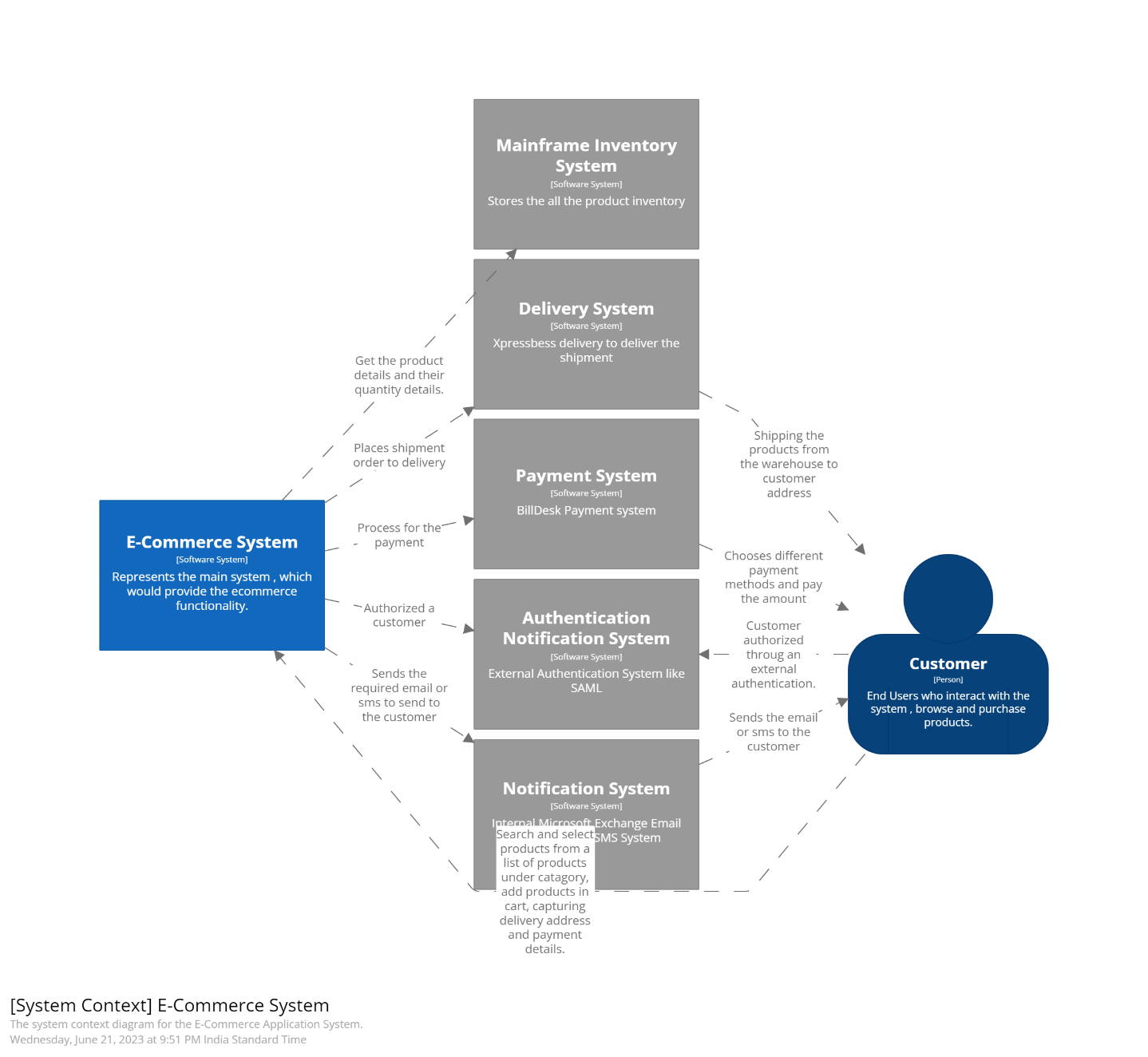
● View orders and returns: for customers

Architecturally Significant Requirements (ASRs) play a crucial role in shaping the architecture of a software system. While there isn't a standardized format for an ASR document, you can follow a structure similar to the one outlined below:

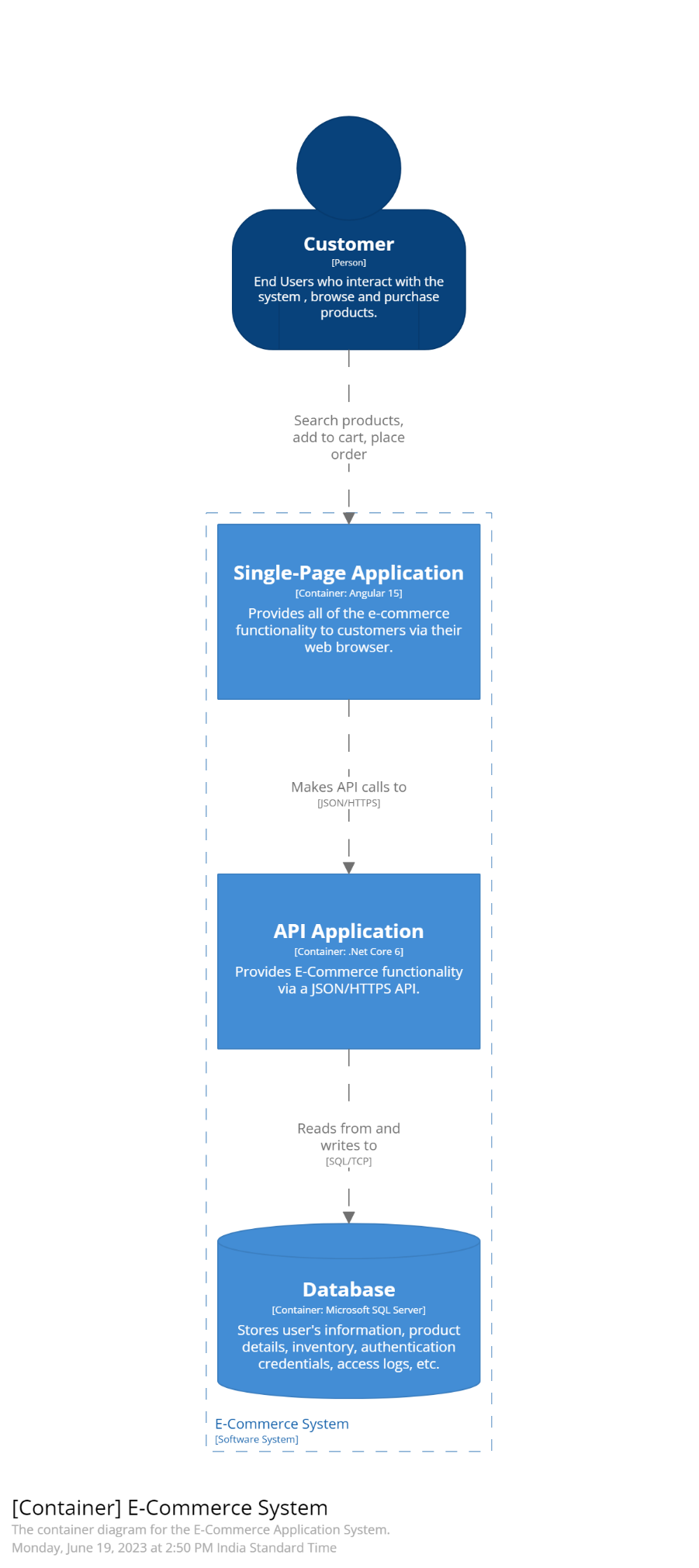
### System Landscape View:



### Context View



### Container View





### Component View

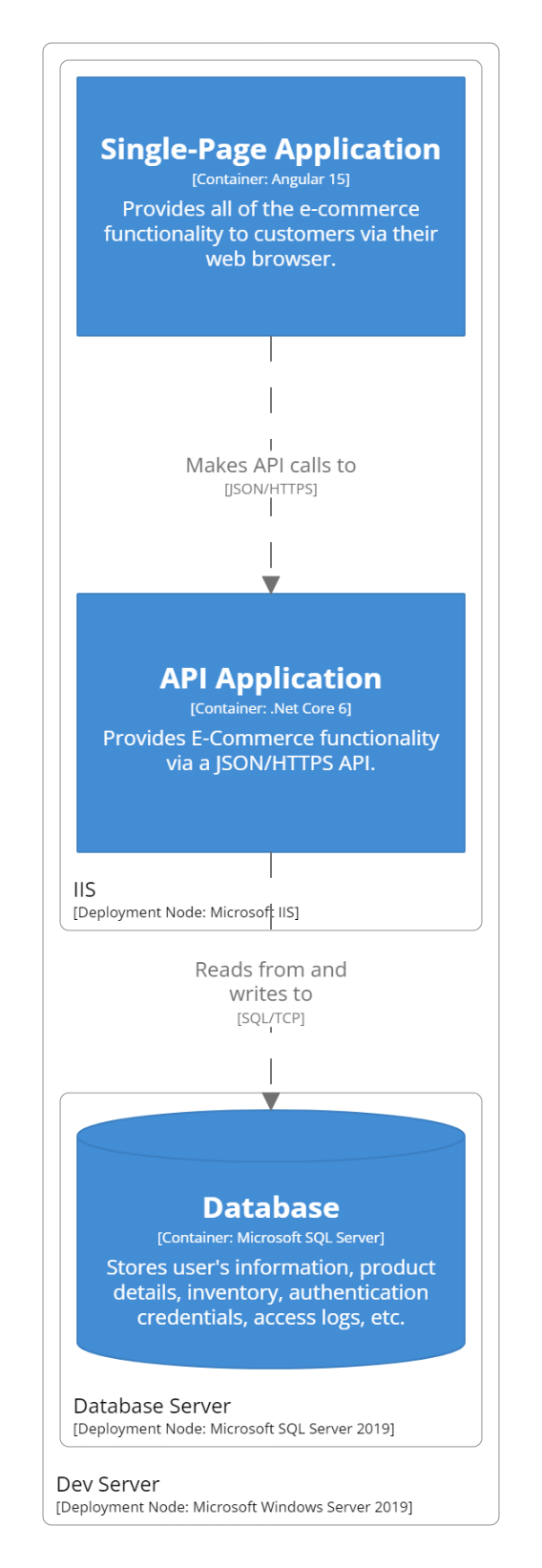
A picture containing text, screenshot, font, number

Description automatically generated

A blue rectangle with white text

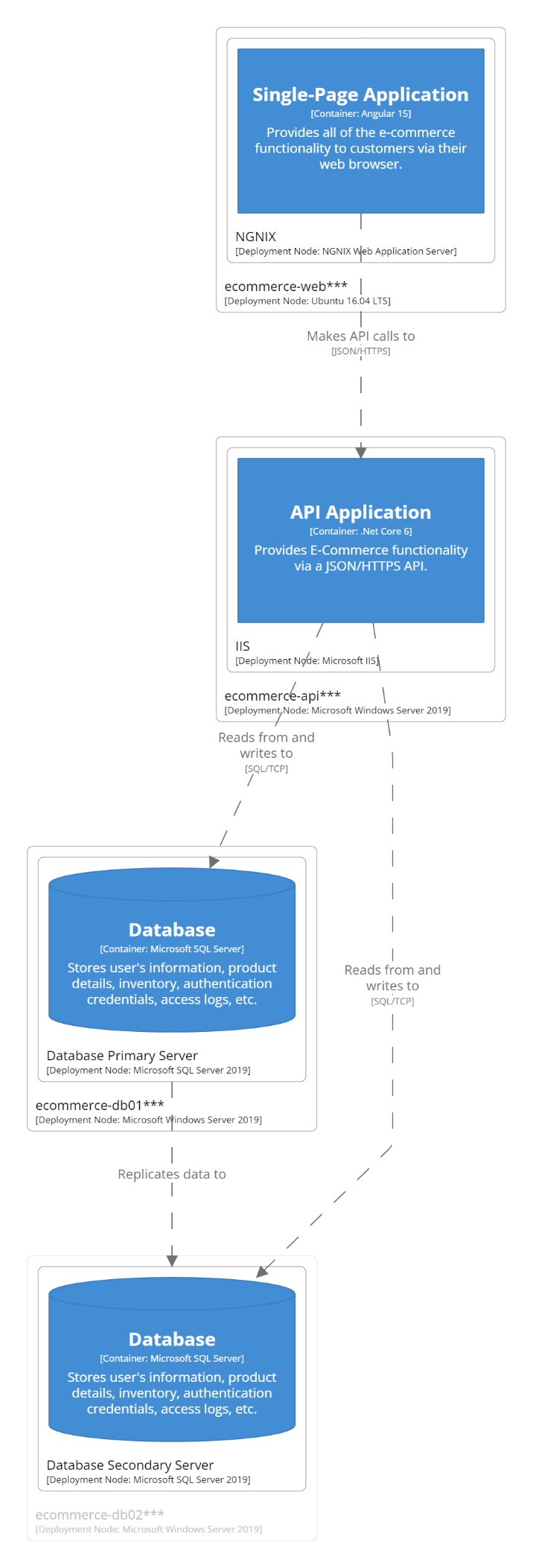
Description automatically generated

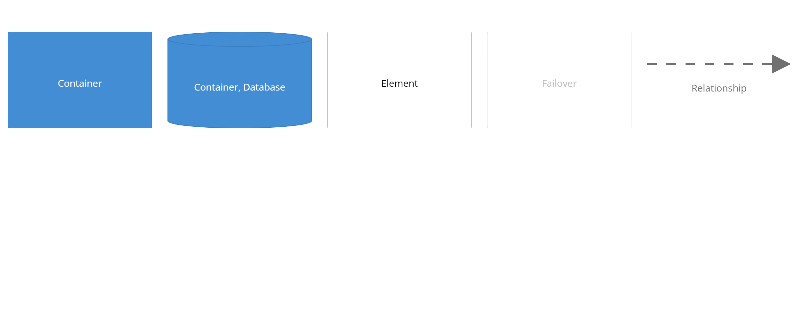
### Deployment View (Development)

A picture containing text, screenshot, font, diagram

Description automatically generated

### Deployment View (Production)

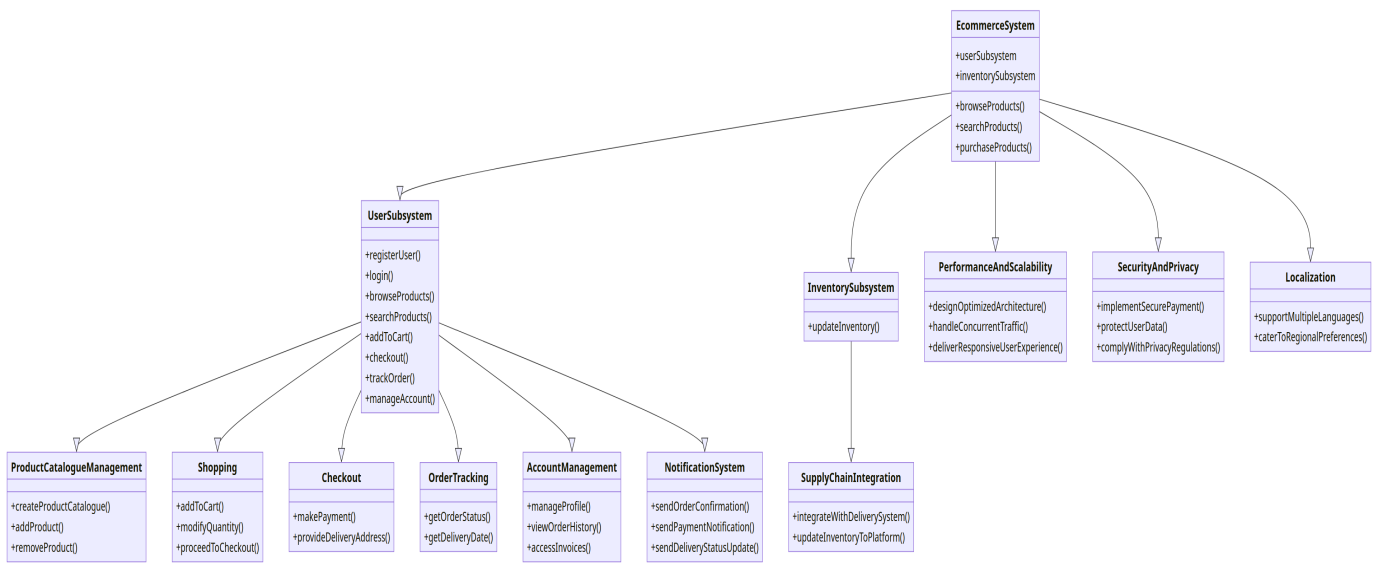




### ER Diagram:



### Class Diagram



### Sequence Diagram

