The Islamia University of Bahawalpur

Department of Information Technology

SOFTWARE REQUIREMENTS SPECIFICATION

(SRS DOCUMENT)

for

< Online Garments Store>

By

Ansa Sarwar

F21BINFT1E02125

Session Fall 2022 – 2026

Supervisor

Ms. Sara Fareed

Bachelor of Science in Information Technology

Table of Contents

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) include the ones given at scope time both in doc and presentation	Action Taken

Supervised by

<Ms. Sara Fareed >

Signature

Introduction

The Garments e-commerce website is a user-centric platform designed to facilitate seamless online shopping for garments. Users can register, search products, manage a shopping cart, and securely complete orders, and there will be an admin panel, admin can handle user accounts, manage products and add or remove products. The website prioritizes an intuitive user interface, efficient order processing, and adherence to security and regulatory standards. With a focus on reliability and user satisfaction, it aims to provide a visually appealing and responsive online shopping experience for garments.

Garments Store is an ambitious endeavour to create a sophisticated and user-friendly ecommerce platform tailored specifically for the garment industry. With a focus on offering a diverse and curated collection of garments, this platform aspires to redefine online shopping experiences by providing a seamless, secure, and enjoyable journey for users seeking high-quality fashion.

In a dynamic and ever-evolving landscape where technology intertwines with fashion, Garments store emerges as a pivotal player, set to redefine the online shopping experience for garments. With the fashion industry rapidly embracing digital transformations, our platform is positioned at the forefront, weaving innovation, accessibility, and style into the fabric of every user interaction. As consumer preferences shift towards the convenience of online retail, Garments aspires to be the go-to destination, offering not just garments but an immersive digital boutique that seamlessly integrates technology and fashion aesthetics.

The people who are not willing to purchase their clothes manually, can visit our website as it provides the most efficient way of shopping. People can purchase the clothes with a single click. This document adheres to standard conventions, utilizing a consistent and clear approach to specifying requirements. Any terminology specific to the garment industry or technical domain will be defined within the document

Purpose

The purpose of the Garments e-commerce website is to provide a seamless and secure platform for users to browse, select, and purchase garments online. The website aims to enhance the overall shopping experience for customers. A garments store serves as a retail space where customers can explore and purchase a variety of clothing items, including apparel, accessories, and footwear. It acts as a marketplace where fashion brands and designers showcase their latest collections.

Scope

The scope of the project encompasses the development of a fully functional e-commerce website, including user registration, product management, shopping cart functionality, order processing, and secure payment processing. Offering a diverse and extensive range

of clothing items, including apparel, accessories, and footwear. Organizing products into categories such as men's, women's, children's, formal wear, casual wear, activewear, and more. Providing a user-friendly interface for customers to easily browse, search, and select products. Ensuring a seamless and enjoyable shopping experience through efficient navigation, visually appealing design, and personalized features. Allowing users to create accounts for personalized experiences, order history tracking, and preferences. Implementing secure authentication processes to protect user accounts

Overall description

Product perspective

Following are the product perspectives according to our website:

1. Operations
    - Add to cart: A user can add products to cart, can remove product from cart.
    - Order Processing: The system should efficiently handle the processing of customer orders, including secure payment processing, order confirmation, and tracking.
    - User Account Management: Operations related to user accounts, including registration, login, and profile management, should be supported.

2. Site Adaptation Requirements
- Device Responsiveness: The site should adapt seamlessly to various devices, including desktops, tablets, and smartphones.
- Localization: The system should support multiple languages and regional preferences to cater to a diverse user base.

3. Product Functions

Specify the functions the garments store will perform:

- Users should be able to search for garments based on various criteria and browse through a diverse product catalogue.
- Users should be able to add, modify, and remove items from their shopping cart before checkout.
- Product browsing and searching with filters.
- Shopping cart management.
- Secure order processing and checkout.
- User account management.
- Product management for administrators.

4. User Characteristics
- End Users: Individuals with varying levels of familiarity with online shopping, including both experienced online shoppers and those new to the platform.

- Administrators: Users responsible for managing product listings, inventory, and overseeing order processing.

5. General Constraints.

- Technology Stack: Constraints related to the technology stack, including compatibility with specific browsers or operating systems.
- Regulatory Compliance: Adherence to industry standards and regulations, such as data protection laws and payment card industry standards.
- Compatibility with common web browsers (Chrome, Firefox, Safari).
- Responsive design for mobile devices

Programming languages: These are the programming languages that can be used in creating a website.

- HTML5
- CSS (Bootstrap)
- JavaScript
- PHP
- SQL

Operating environment

The "Online Garments Store" will operate in the following environments:

1. Hardware Platform:

- Client-side: Any device with internet access, including desktops, laptops, tablets, and smartphones.
- Server-side: A web server with a minimum of 8 GB RAM, 4-core CPU, and 100 GB storage for hosting the website and database.

2. Operating Systems:

- Client-side: Compatible with Windows, macOS, Linux, Android, and iOS.
- Server-side: The application will run on a Linux-based operating system (e.g., Ubuntu 20.04) or Windows Server.

3. Web Browsers:

The system shall operate correctly with the following web browsers:

- Google Chrome (version 80 and above).
- Mozilla Firefox (version 75 and above).
- Apple Safari (version 11 and above).

- Microsoft Edge (version 80 and above).

4. Backend Environment:

- Server-side Language: PHP version 7.4 or higher.
- Database: MySQL version 5.7 or higher.

5. Geographical Locations:

- The system is intended to serve users globally, with initial focus on regions with a high demand for online garment shopping.
- Servers and databases can be hosted in geographically distributed data centers to reduce latency for users from different locations.

6. Hosting and Deployment:

- The application will be hosted on a cloud platform such as AWS, Google Cloud, or Microsoft Azure to ensure scalability and availability.
- Domain and website hosting will be managed by a reliable hosting provider.

7. Internet Requirements:
   - The system requires a stable internet connection with at least 2 Mbps bandwidth for users to experience smooth navigation and interactions.

Design and implementation constraints

The development of the Online Garments Store is subject to the following design and implementation constraints:

1. Programming Languages and Frameworks:

   - Front-end: The application must use HTML, CSS, and JavaScript for the front-end design. No modern JavaScript frameworks (e.g., React, Angular, or Vue.js) are permitted to maintain simplicity and reduce complexity.
   - Back-end: PHP is the required language for server-side scripting due to its widespread use, ease of integration, and compatibility with the database.
2. Database:

- MySQL is mandatory for database implementation. The system must use MySQL queries to manage data storage and retrieval due to its compatibility with PHP and cost-effectiveness.

3. Compatibility:

    - The system must ensure compatibility with older browser versions (e.g., Chrome 80+ and Firefox 75+) to accommodate users with legacy systems. Advanced modern features (e.g., Web Assembly or ES6 modules) should be avoided to maintain backward compatibility.

4. Hosting Environment:

    - The application must be deployable on shared hosting servers to reduce operational costs for small-scale businesses. This restricts the use of technologies requiring advanced server configurations (e.g., Node.js or containerized solutions like Docker).

5. User Interface (UI):

    - The design must follow responsive web design principles using only HTML and CSS. External libraries like Bootstrap can be used, but advanced design tools such as Tailwind CSS or Material Design are not allowed.

6. Security Constraints:

    - The implementation must use basic security measures, such as hashing passwords with PHP's built-in functions and validating user input. Advanced security frameworks are not to be included.

7. Performance:

    - The system must perform optimally on shared hosting environments, restricting the use of resource-intensive code libraries or computationally heavy operations.

8. Scalability:

- The initial design will not incorporate advanced scalability features, such as microservices or multi-database architecture. The focus will remain on small-scale deployments suitable for startups or small retailers.

9. Documentation and Code Standards:

- Developers must adhere to basic PHP coding standards (e.g., PSR-12) and maintain code readability and documentation, avoiding the use of overly complex or obscure methods.

10. Development Tools:

- The project will be developed using lightweight IDEs such as Visual Studio Code, and no enterprise-grade tools (e.g., IntelliJ IDEA) will be used.

Requirement identifying technique

Use case diagram

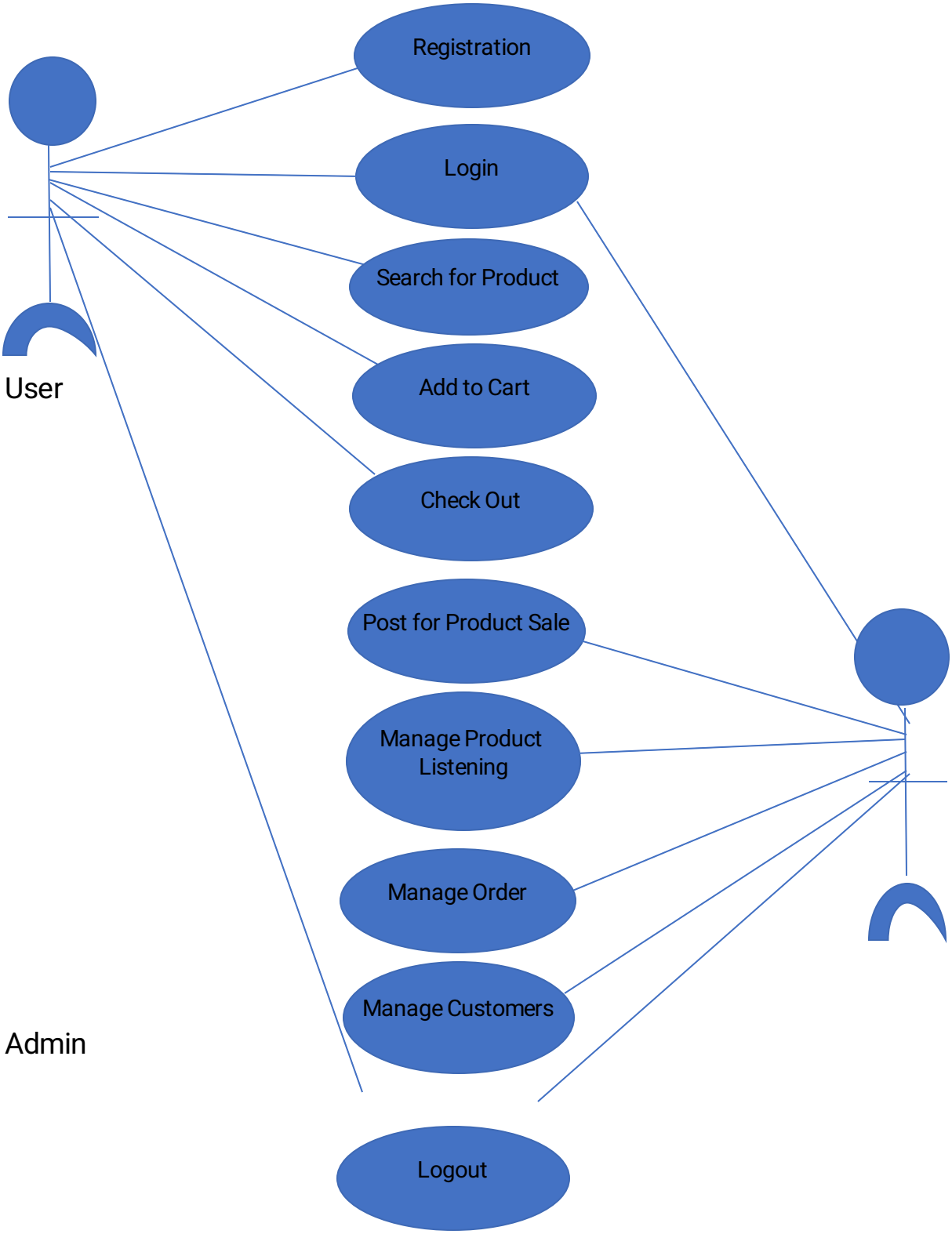

Figure-1: Use Case Diagram

Use case description

The web application enables users to browse a catalog of garments, add items to a shopping cart, and make purchases. The admin oversees the platform by managing inventory, monitoring sales, and handling user accounts.

Table Show the detail use case:

Use Case ID:	UC-1
Use Case Name:	Online Garments Store use case
Actors:	Actor1: User Actor2: Admin
Description:	1. Admin The Admin is responsible for managing and maintaining the system to ensure smooth operation. ● Role: Handles product listings, orders, and customer management. ● Actions: ■ Post products for sale. ■ Manage product listings (add, update, or delete products). ■ Process and manage customer orders. ■ Manage customer data such as user profiles and order histories. 2. User The User interacts with the system to purchase garments and manage their account. ● Role: Uses the system to search for products, place orders, and manage their account. ● Actions: ■ Register as a new user. ■ Log in to access their account. ■ Search for garments by category,

name, or filters.

- Add products to the shopping cart.
- Checkout to place an order.
- Log out from their account.

Trigger:

Admin: Triggers occur when the admin manages products, orders, or customers.

User: Triggers occur based on user actions such as account management, searching, adding products to the cart, or completing orders.

Preconditions:

Admin Preconditions

1. Post Product for Sale
    - Admin must be logged in.
    - Admin must have product posting permissions.
2. Manage Product Listing
    - Admin must be logged in.
    - Products must exist in the system.
3. Manage Orders
    - Admin must be logged in.
    - Orders must be placed by users.
4. Manage Customers
    - Admin must be logged in.
    - Customer accounts must exist.

User Preconditions

1. Registration
    - User is not already registered.
2. Login
    - Users must have a registered account.
    - User must provide valid credentials.
3. Search for Product

- Products must be available in the system.

4. Add to Cart
    - Users must be logged in (if required).
    - Product must be in stock.
5. Checkout
    - Users must have items in their cart.
    - Valid payment and shipping details must be provided.
6. Logout
    - Users must be logged in.

Postconditions:

Admin Postconditions

1. Post Product for Sale
    - Product is added to the product database and displayed on the website.
2. Manage Product Listing
    - Product information is updated or removed.
3. Manage Orders
    - Order status is updated (e.g., processed, shipped).
4. Manage Customers
    - Customer data is updated or deleted.

User Postconditions

1. Registration
    - User account is created and ready for login.
2. Login
    - Users gain access to their account.
3. Search for Product
    - Relevant products are displayed.
4. Add to Cart
    - Product is added to the shopping cart.
5. Checkout
    - Order is placed, payment is processed.

6. Logout
    - User is logged out and the session is terminated.

Normal Flow:

Admin Normal Flow

1. Post Product for Sale
    - Admin logs in → Clicks "Add Product" → Enters product details → Product is posted and listed.
2. Manage Product Listing
    - Admin logs in → Selects product → Edits or deletes product → Updates are saved.
3. Manage Orders
    - Admin logs in → Views new orders → Updates order status (e.g., shipped) → Order status is updated.
4. Manage Customers
    - Admin logs in → Views customer information → Updates or deletes customer data → Changes are saved.

User Normal Flow

1. Registration
    - User opens registration page → Fills in details → Submits form → Account is created.
2. Login
    - User opens login page → Enters credentials → Submits → User is logged in.
3. Search for Product
    - User enters search query → System displays matching products → User selects a product.
4. Add to Cart
    - User selects product → Clicks "Add to Cart" → Product is added to cart.
5. Checkout
    - User reviews cart → Provides payment details → Confirms order → Order is processed.
6. Logout

- User clicks "Logout" → User is logged out and session ends.

Alternative Flows:
[Alternative Flow 1 – Not in Network]

Admin Alternative Flows

1. Post Product for Sale (Alternative)
    - If admin fails to provide required product details → System prompts to complete missing fields → Admin completes the fields → Product is posted.
2. Manage Product Listing (Alternative)
    - If admin tries to delete a non-existent product → System displays an error → Admin selects another product to manage.
3. Manage Orders (Alternative)
    - If order status cannot be updated → System shows an error message → Admin retries or contacts support.
4. Manage Customers (Alternative)
    - If the admin tries to update a non-existent customer → System shows error → Admin searches for the correct customer.

User Alternative Flows

1. Registration (Alternative)
    - If a user enters invalid details (e.g., existing email) → System shows an error → User corrects details and resubmits.
2. Login (Alternative)
    - If the user enters wrong credentials → System shows an error → User can reset password or try again.
3. Search for Product (Alternative)
    - If no products match search → System displays "No results found" → User refines search criteria.
4. Add to Cart (Alternative)

- If the product is out of stock System shows "Out of Stock" message User is prompted to choose another product.

5. Checkout (Alternative)
    - If payment fails System shows payment error User retries with a different payment method.
6. Logout (Alternative)
    - If session expires User is automatically logged out and redirected to the login page.

Exceptions:

Admin Use Case Exceptions

1. Post Product: Missing details, duplicate products, database errors.
2. Manage Listings: Product not found, unauthorized access, dependency issues.
3. Manage Orders: Invalid order ID, status update errors, database failures.
4. Manage Customers: Customer not found, invalid updates, unauthorized access.

User Use Case Exceptions

1. Registration: Duplicate email, weak password, network issues.
2. Login: Incorrect credentials, account locked/inactive.
3. Search Product: No results, invalid filters, load errors.
4. Add to Cart: Out of stock, invalid ID, session timeout.
5. Checkout: Payment failure, address errors, cart inconsistency.
6. Logout: Session issues, unauthorized attempts.

Business Rules

Admin Business Rules

1. Only admins can post, update, or delete product listings.
2. Products must include valid details (name, price, description).
3. Admins can only manage orders and customers within their role.
4. Deleted products should not affect existing orders.

User Business Rules

1. Users must register with a unique email and strong password.
2. Login is required to add products to the cart or checkout.
3. Search results should match filters and keywords accurately.
4. Items added to the cart must reflect real-time stock availability.
5. Payments must be processed securely and verified before order confirmation.

Assumptions:

Admin Assumptions

1. Admins have valid login credentials.
2. Admins have sufficient permissions for all operations.
3. Products and orders are stored in a reliable database.
4. Admin actions follow company policies.

User Assumptions

1. Users have internet access and a compatible device.
2. Users provide accurate registration and payment details.
3. Stock quantities are updated in real time.

4. Users understand the basic system functionality (e.g., adding to cart).

Functional Requirements

1. User Management:

- Users should be able to register with unique credentials (email and password).
- The system must validate login credentials for authentication.
- Admins should manage user profiles (view, update, or delete users).

Identifier	USR-001
Title	User Registration and Login
Requirement	User Perspective: "Users should be able to register with unique credentials (email and password)." System Perspective: "The system must validate login credentials during authentication."
Source	Customer Requirements
Rationale	Enables secure access for users while preventing duplicate accounts.
Business Rule (if required)	Each email must be unique in the system.
Dependencies	None
Priority	High

2. Product Management:

- Admins can post new products with details (name, price, description, stock).
- Admins can update or delete existing product listings.

- Users should search for products by category, name, or filters

Identifier	ADM-002
Title	Product Management
Requirement	Admin Perspective: "Admins can post new products with details (name, price, description, stock)." "Admins can update or delete existing products."

Source	Admin Requirements
Rationale	Enables admins to maintain an up-to-date and relevant product catalog for users.
Business Rule (if required)	Stock cannot be set below 0.
Dependencies	None
Priority	High

3. Cart Management:

- Users can add, update, or remove items from their cart.
- The system must calculate the total cost, including applicable taxes and shipping.
- Secure payment processing must be integrated into the checkout process.

Identifier	USR-003
Title	Cart Management
Requirement	User Perspective: "Users can add, update, or remove items from their cart." System Perspective: "The cart must update in real-time with changes to items."
Source	Customer Requirements
Rationale	Provides a seamless shopping experience, enabling users to prepare for checkout easily.
Business Rule (if required)	Cart items must not exceed available stock.
Dependencies	USR-002
Priority	High

4. Session Management:

- Users and admins must log out securely to terminate sessions.
- The system should automatically log out inactive sessions after a set timeout period.

Identifier	SEC-001
Title	Session Management
Requirement	User Perspective: "Users and admins must log out securely to terminate sessions." System Perspective: "The system automatically logs out inactive sessions after a timeout period."
Source	Security Requirements
Rationale	Enhances platform security by reducing the risk of unauthorized session access.
Business Rule (if required)	Timeout period is set to 10 minutes.
Dependencies	USR-001
Priority	High

5. Reporting and Analytics:

- Admins can generate reports for product sales, customer data, and order trends.
- The system should provide dashboard insights into real-time system performance.

Identifier	ADM-004
Title	Reporting and Analytics
Requirement	Admin Perspective: "Admins can generate sales, customer data, and order trend reports." System Perspective: "The dashboard provides real-

time system performance insights."

Source	Admin Feedback
Rationale	Enables data-driven decision-making for business growth and optimization.
Business Rule (if required)	Reports should include the last 12 months.
Dependencies	ADM-003
Priority	Medium

Non-Functional Requirements

- Performance Requirements
  - Scalability
- Security Requirements
- Availability
- Usability
- Maintainability
- Compliance

Usability

Usability requirements focus on ensuring that the system is user-friendly, efficient, and accessible. They define how the system should be designed to optimize the user's experience, making it easier to interact with, reducing errors, and enhancing user satisfaction. Usability requirements often consider factors such as ease of learning, ease of use, error recovery, efficiency, and accessibility.

1. Ease of Learning:

   The system should be easy for new users to understand and operate with minimal training.

   Example: "The system shall provide an introductory tutorial upon first login."

2. Ease of Use:

   The system should have a simple, intuitive interface that minimizes user effort to complete tasks.

   Example: "The system shall provide clear, easily recognizable icons for all primary functions."

3. Error Avoidance:

The system should prevent users from making errors, or at least minimize the likelihood of them.

Example: "The system shall display a confirmation prompt when a user attempts to delete a critical item."

4. Error Recovery:

   The system should allow users to recover from errors quickly and with minimal disruption.

   Example: "If a user enters invalid data, the system shall display an error message with clear instructions on how to correct the mistake."

5. Efficiency of Interactions:

   The system should allow users to complete tasks in a timely and efficient manner.

   Example: "The system shall allow a user to filter product search results by multiple criteria within 3 clicks."

6. Accessibility:

   The system should be accessible to all users, including those with disabilities.

   Example: "The system shall support screen reader functionality for users with visual impairments."

Performance

Performance requirements define how well the system should behave under specific conditions. They are critical to ensuring the system meets user expectations and operates efficiently. These requirements are often closely tied to response time, throughput, availability, and scalability under various loads. Below are specific performance requirements for different system operations.

References

ISO/IEC 27001:2013 – Information Security Management Systems.

W3C Web Content Accessibility Guidelines (WCAG) 2.1 – World Wide Web Consortium, 2018.

"SRS for XYZ System, Version 2.0", Product Requirements Document for Feature A, Company ABC.

"Material Design Guidelines by Google", Google, 2020.

OAuth 2.0 Authentication Integration Guide, OAuth Working Group, 2019.

General Data Protection Regulation (GDPR) Compliance, European Union, 2018.