**1. Introduction**

**1.1 Purpose**

The purpose of this document is to provide a comprehensive design for the Bright Brightness Shop System, which is intended to streamline and enhance the business operations of the BestBrightness Pongola Shop. This system will address critical business needs, including inventory management, customer management, sales tracking, and e-commerce integration, through a robust web application built using ASP.NET.

**1.2 Scope**

The scope of this system design includes the following:

**Included:**

* Development of a web-based application for inventory management, customer management, and sales tracking.
* Integration of an e-commerce platform to enable online sales.
* Implementation of security measures to protect sensitive data.
* Design of a user-friendly interface for shop administrators and customers.

**Excluded:**

* Development of a mobile application.
* Integration with third-party accounting software.

**1.3 Overview**

This document outlines the system architecture, including hardware, software, and network components. It details the external interfaces, database design, user interface design, and application components. It also considers design aspects such as performance, scalability, security, usability, and dependability.

**2. System Architecture**

**2.1 High-Level Overview**

The system architecture of the Bright Brightness Shop System includes a three-tier architecture:

* **Presentation Layer:** The user interface (UI) components where users interact with the system.
* **Business Logic Layer**: Handles the core functionality, including business rules and operations.
* **Data Access Layer**: Manages interactions with the database, including CRUD operations.

**2.2 Hardware Architecture**

- User Devices: Desktop computers, laptops, tablets, and smartphones used by shop administrators and customers to access the system.

- System Devices: Servers hosting the web application, database, and e-commerce platform.

**2.3 Software Architecture**

**User Applications:**

* Web browsers on user devices for accessing the web application.

**- System Applications:**

* ASP.NET Core web application hosting the business logic and UI.
* SQL Server for managing the database.
* IIS (Internet Information Services) as the web server.

**2.4 Network Architecture**

**- Network Infrastructure:**

* The system components communicate over a secure HTTPS connection.
* The internal network uses Ethernet or Wi-Fi to connect servers and user devices.
* Internet access for remote users and online sales transactions.

**- Network Protocols:**

* HTTP/HTTPS for web communications.
* TCP/IP for internal networking.
* SSL/TLS for secure communications.

**3. External Interfaces**

**3.1 Hardware Interfaces**

* Interface with User Devices: Standard interfaces like USB, Ethernet, and Wi-Fi for connecting user devices to the network.
* Interface with System Devices: Servers connected via Ethernet to ensure fast and reliable communication.

**3.2 Software Interfaces**

- Web API: The web application may expose APIs for future integration with external systems.

- Database Interfaces: The application will interface with the SQL Server database using Entity Framework Core.

**3.3 Communication Interfaces**

1. Internal Communication: ASP.NET Core communicates with SQL Server using Entity Framework Core.
2. External Communication: The system communicates with user devices via HTTPS, ensuring data is encrypted during transmission.

**3.4 System Security and Integrity Controls**

* 1. User Authentication: ASP.NET Identity for managing user authentication and authorization.
  2. Data Encryption: SSL/TLS for encrypting data in transit and AES for data at rest.
  3. Input Validation: To prevent SQL injection and other security threats.

**4. Database Design**

**4.1 Data Requirements**

**- Data Dictionary:**

* Tables: Products, Sales, Customers, Users.
* Fields: ProductId, ProductName, Quantity, Price, etc.

**4.2 Database Schema**

**- Tables and Relationships:**

* Products table with ProductId as the primary key.
* Sales table linked to Products and Users via foreign keys.
* Customers table with a one-to-many relationship with Sales.

**4.3 Database Physical Design**

* + Data Storage: SQL Server database hosted on a dedicated server.
  + Storage Size: Adequate to accommodate the shop's growing data, including product details, sales records, and customer information.

**4.4 Data Security**

**Security Measures:**

* + Role-based access control to limit access to sensitive data.
  + Regular backups to prevent data loss.
  + Encryption of sensitive fields like passwords and payment information.

**5. User Interface Design**

* + Layouts and Mockups:
  + Home Page: Welcome message, login/register buttons, and product overview.
  + Dashboard: Features buttons for Record Sales, Manage Inventory, View Reports, with a clean and modern layout.
  + Forms: Simplified and mobile-friendly forms for login, registration, and sales recording.

**6. Application Components Design**

**6.1 Component Overview**

**Subsystems:**

* + Inventory Management: Tracks product stock levels, reordering, and adjustments.
  + Sales Management: Manages sales transactions and records.
  + Customer Management: Stores customer details and purchase history.
  + E-commerce Integration: Facilitates online sales and order processing.

**6.2 Detailed Component Design**

* + Purpose: Describe the purpose of the component.
  + Data Structures: Describe any data structures used by the component.
  + Interaction: Describe how the component interacts with other components.

**7. Design Considerations**

1. **Performance:** Describe how the design addresses performance requirements.
2. **Scalability:** Describe how the design allows for scalability.
3. **Security:** Outline security considerations and how the design addresses them.
4. **Usability:** Describe usability considerations and how they are incorporated into the design.
5. **Dependability:** Explain how the design will support dependability.

**8. Appendices**

?

**9. References**