



*Best
Brightness*

SYSTEM REQUIREMENTS

Presented by: Innovation Nexus



OVERVIEW

01

System Vision

02

System Functional Requirements

03

System Object Classes

04

Evidence of
Information Gathering



SYSTEM VISION

Client: Mr. Msane's
cleaning products
business

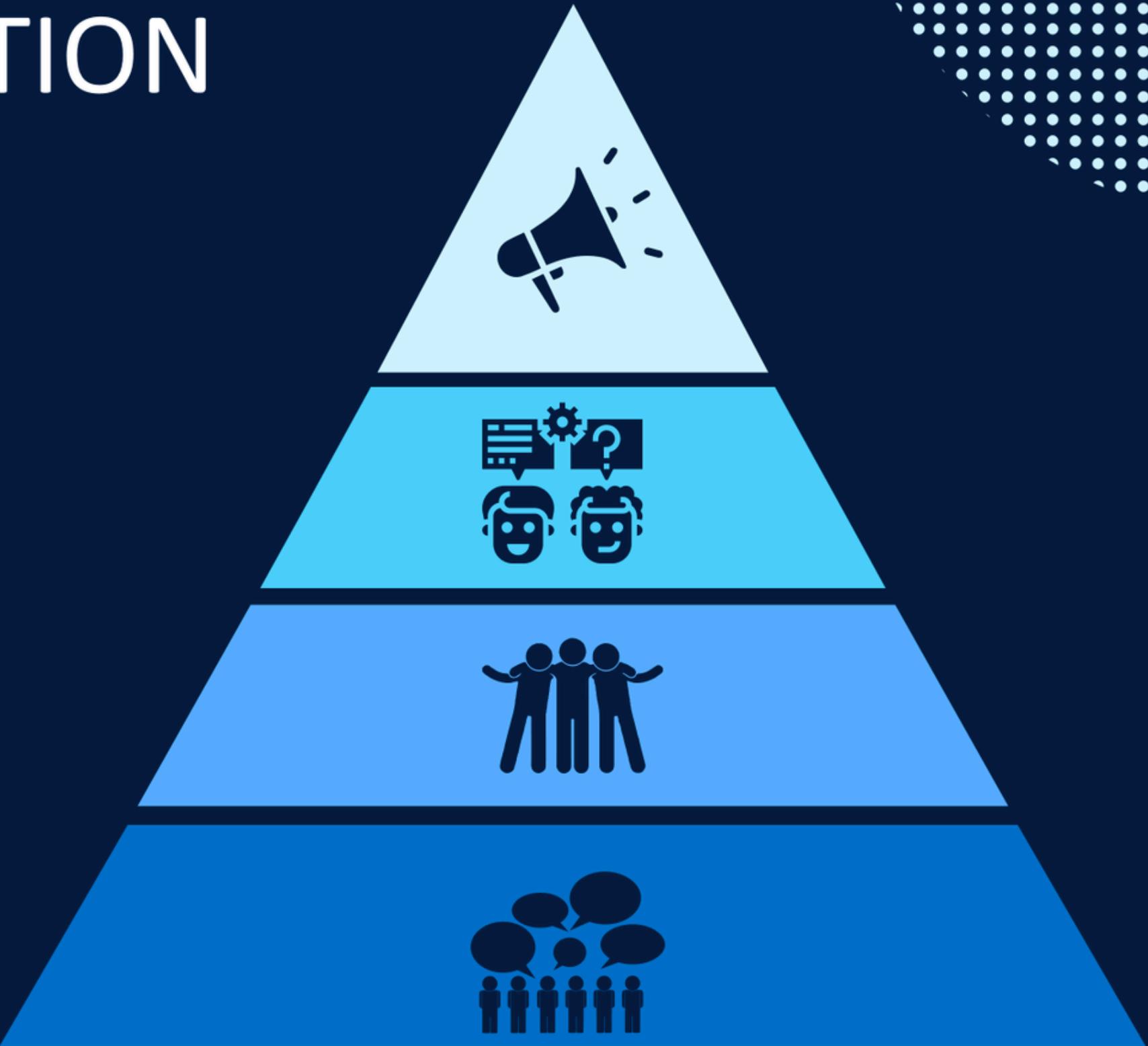
Medium-sized,
multiple branches

Goals: Improve
inventory, customer
service, compliance



: PROBLEM DESCRIPTION

- 01** MANUAL INVENTORY → ERRORS
- 02** NO CUSTOMER TRACKING, LOYALTY PROGRAMS
- 03** NO E-COMMERCE, LOCAL SALES ONLY
- 04** LACK OF INTEGRATION → INEFFICIENCIES
- 05** LIMITED REPORTING → POOR DECISIONS





GOALS AND OBJECTIVES

Growth

Customer Satisfaction



SYSTEM CAPABILITIES



Inventory: Real-time tracking, reorder alerts, barcode scanning



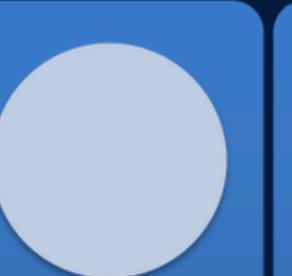
Customer: Profiles, loyalty programs, personalized services



E-commerce: Online store, real-time stock, multiple payments



Integration: Centralized database, real-time synchronization



Reporting: Dashboards, customizable reports



BUSINESS BENEFITS

Tangible: Fewer inventory errors,
increased sales, cost savings

Intangible: Better customer
satisfaction, compliance, decisions



```
graph TD; A[User management: Register, login, logout] --> B[Inventory: Upload, update, view products]; B --> C[Purchase: Cart, checkout]; C --> D[Reporting: Sales and inventory reports];
```

User management: Register,
login, logout

Inventory: Upload,
update, view products

Purchase: Cart, checkout

Reporting: Sales and
inventory reports

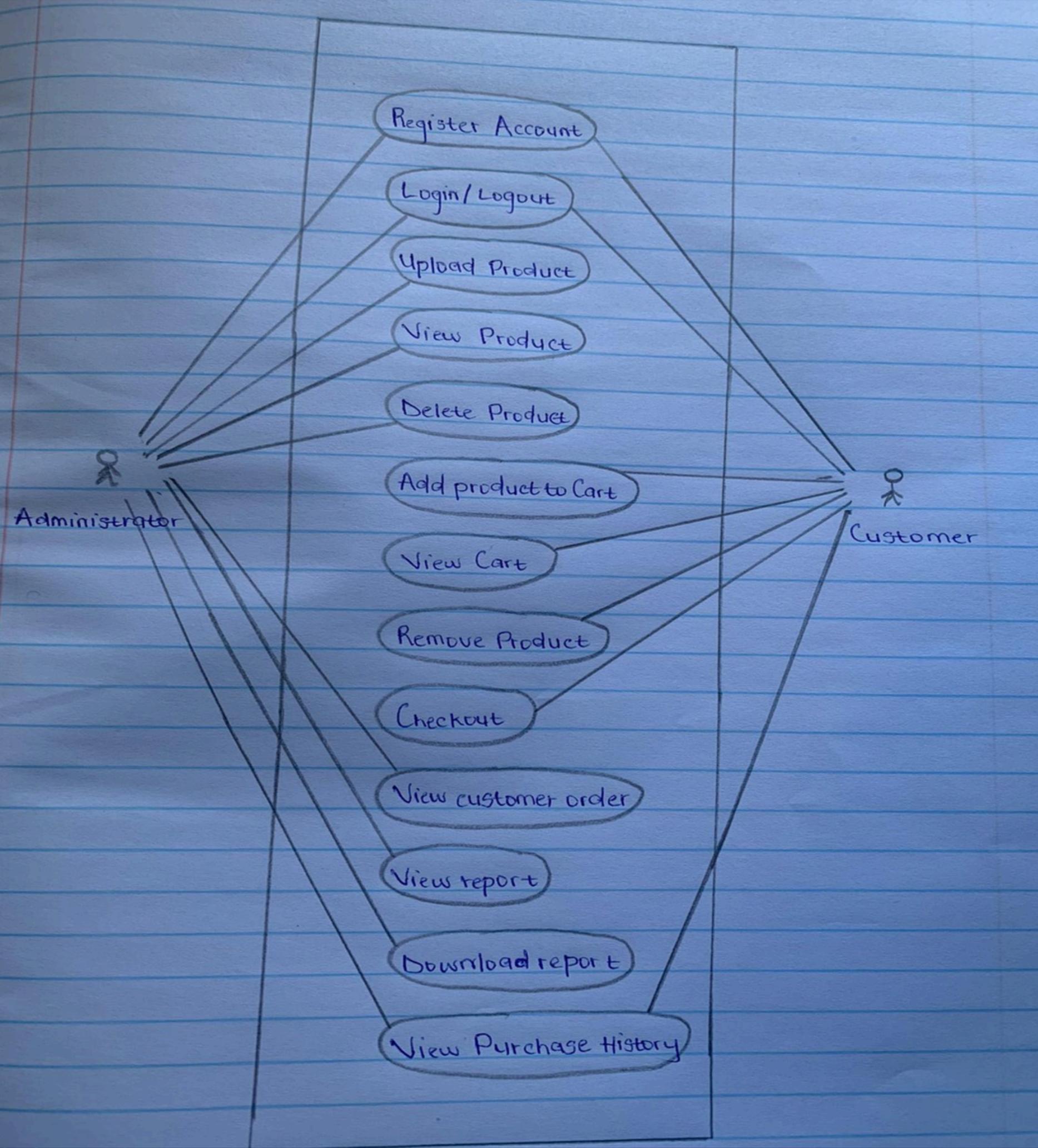


PROJECT STAKEHOLDERS

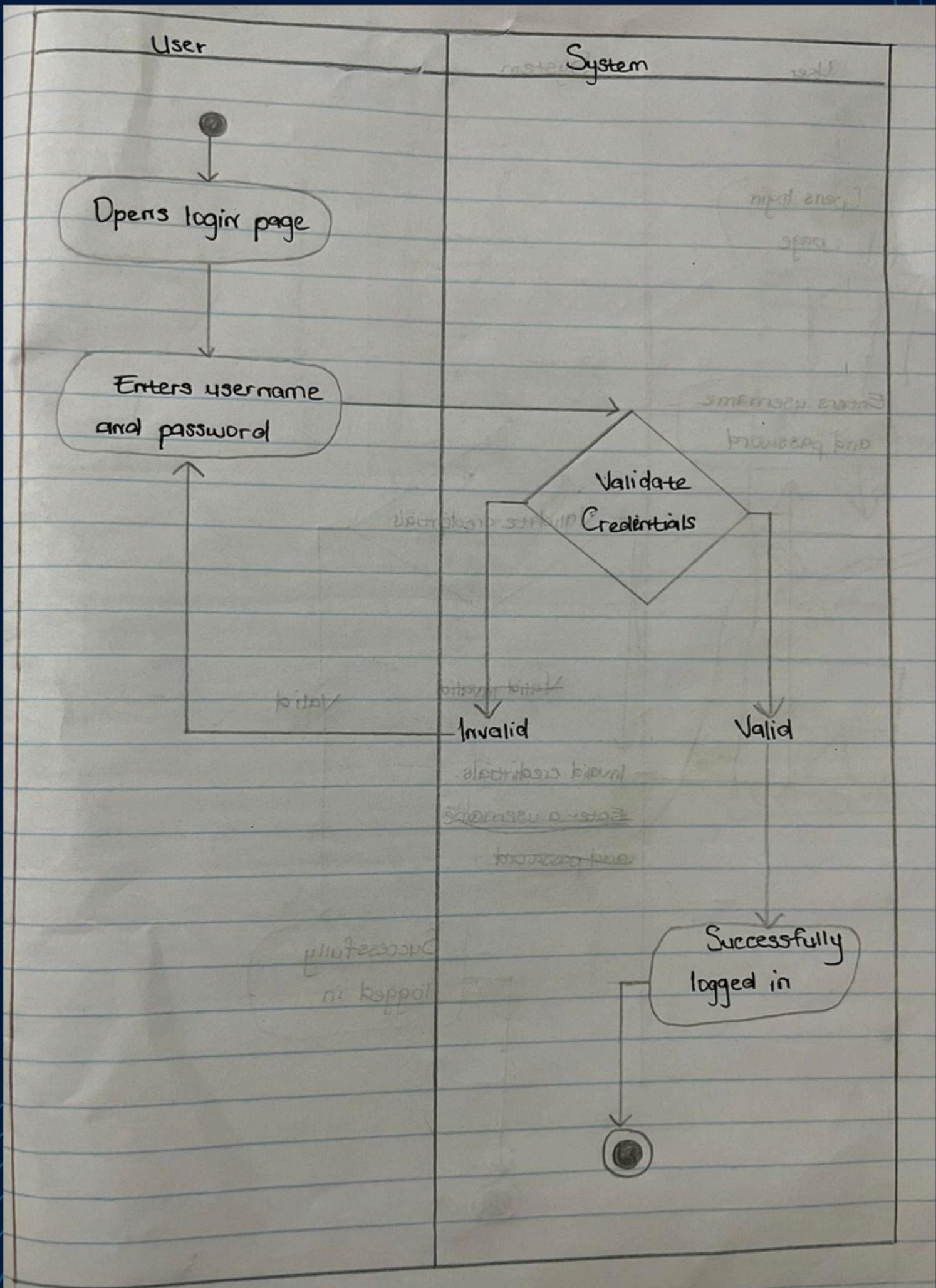


System Functional Requirements

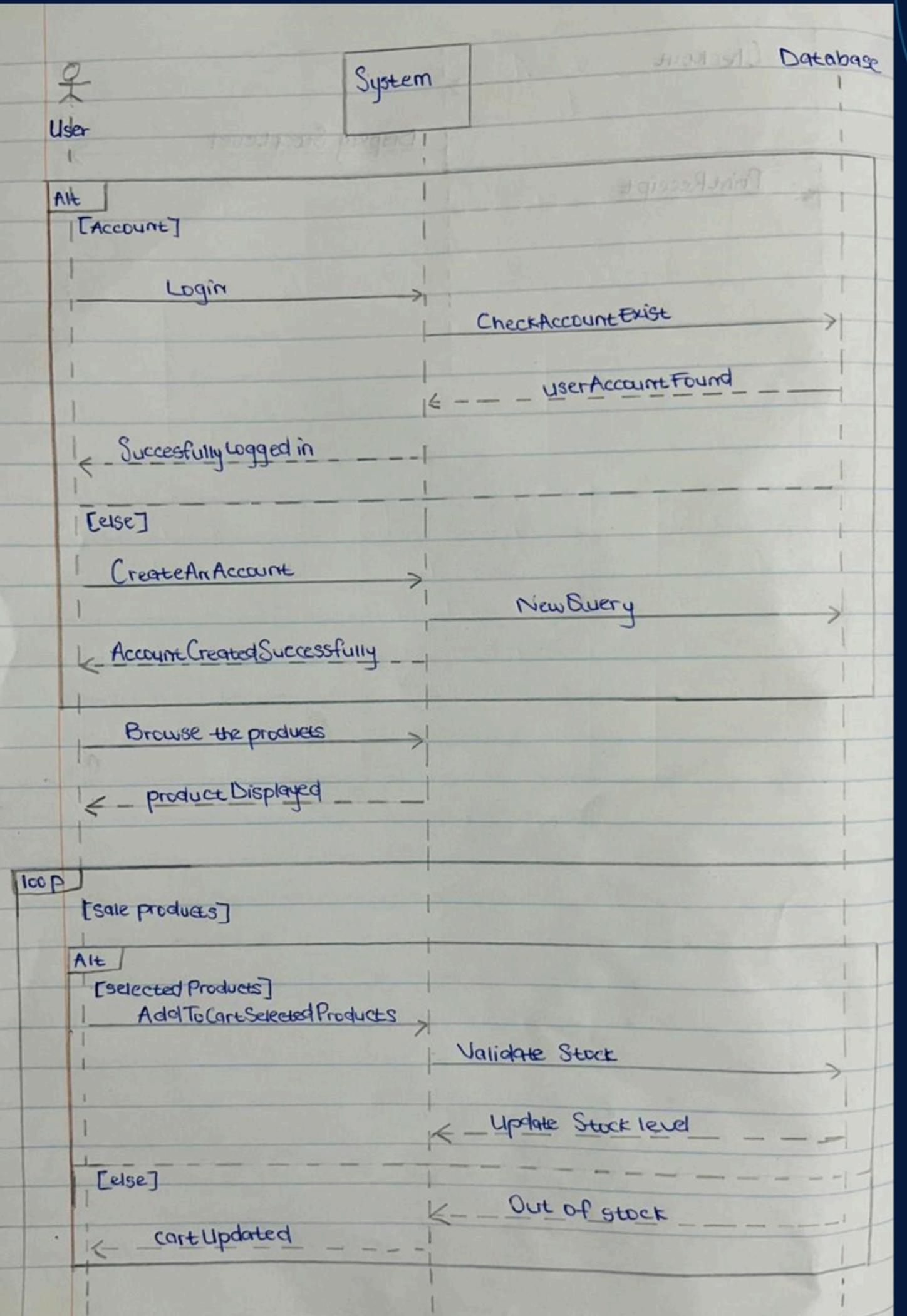
- Use Cases: Login, browse, purchase, upload, reports
- Diagrams: Use case diagrams: Actors & interactions
- Activity diagrams: Workflow steps
- Sequence diagrams: System-user communication



USE CASE DIAGRAM

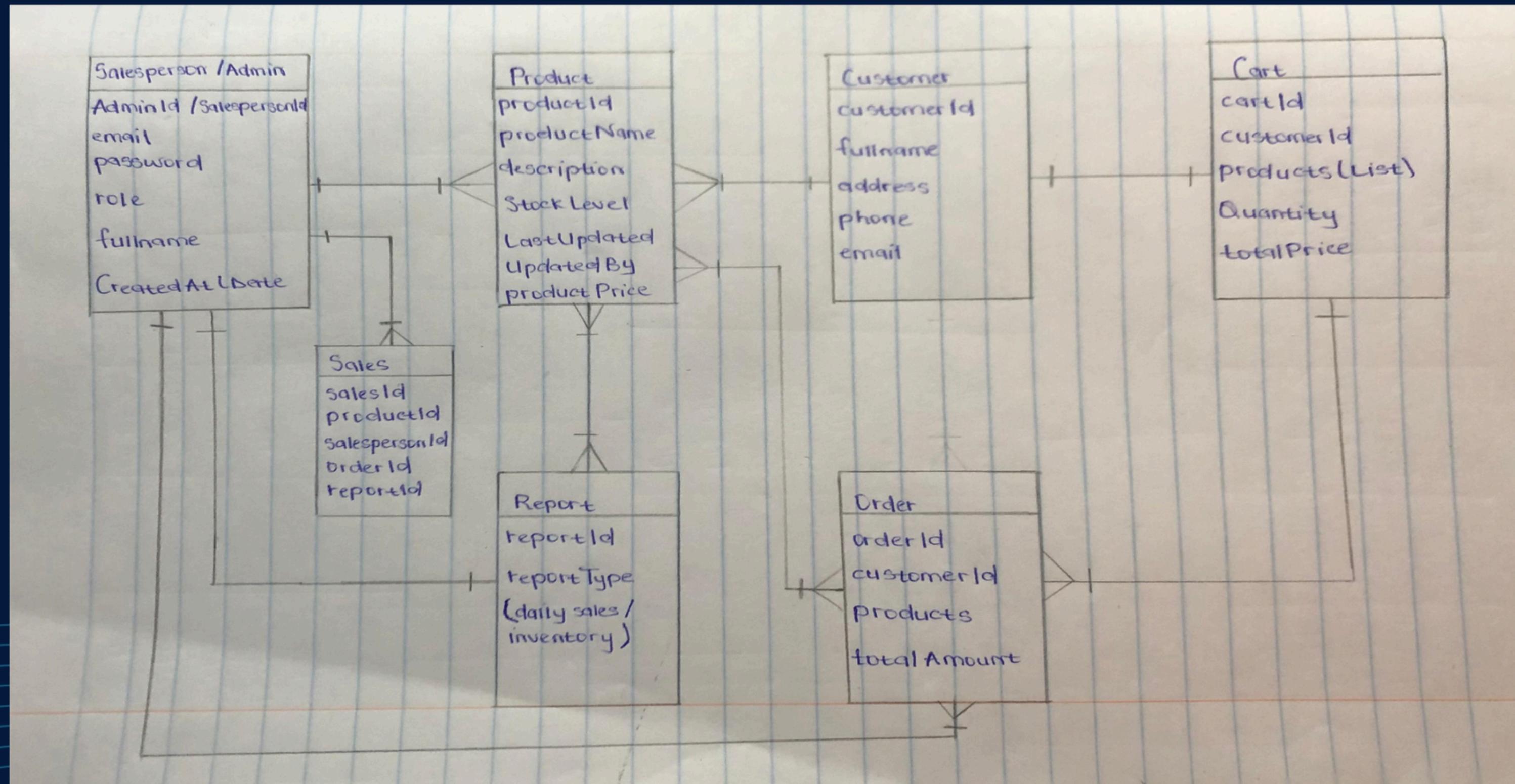


ACTIVITY DIAGRAM



SEQUENCE DIAGRAMS

SYSTEM OBJECT DIAGRAMS



EVIDENCE OF INFORMATION GATHERING

Survey Response:

- How often do you shop at BestBrightness?

Occasionally

- Which products do you purchase most frequently? (Select all that apply):

Cleaning Detergents

Tissue Papers

- How satisfied are you with our current inventory availability?

Satisfied

- How do you rate the ease of finding products in our store?

Easy

- How would you rate our customer service?

Good

- Do you use our store's online platform for browsing or purchasing products?

Yes

- If yes, how satisfied are you with the online shopping experience?

Satisfied

- What improvements would you like to see in our online platform? (Select all that apply):

Easier Navigation

Faster Checkout Process

- How likely are you to recommend BestBrightness to others?

Likely



Best Brightness Shop

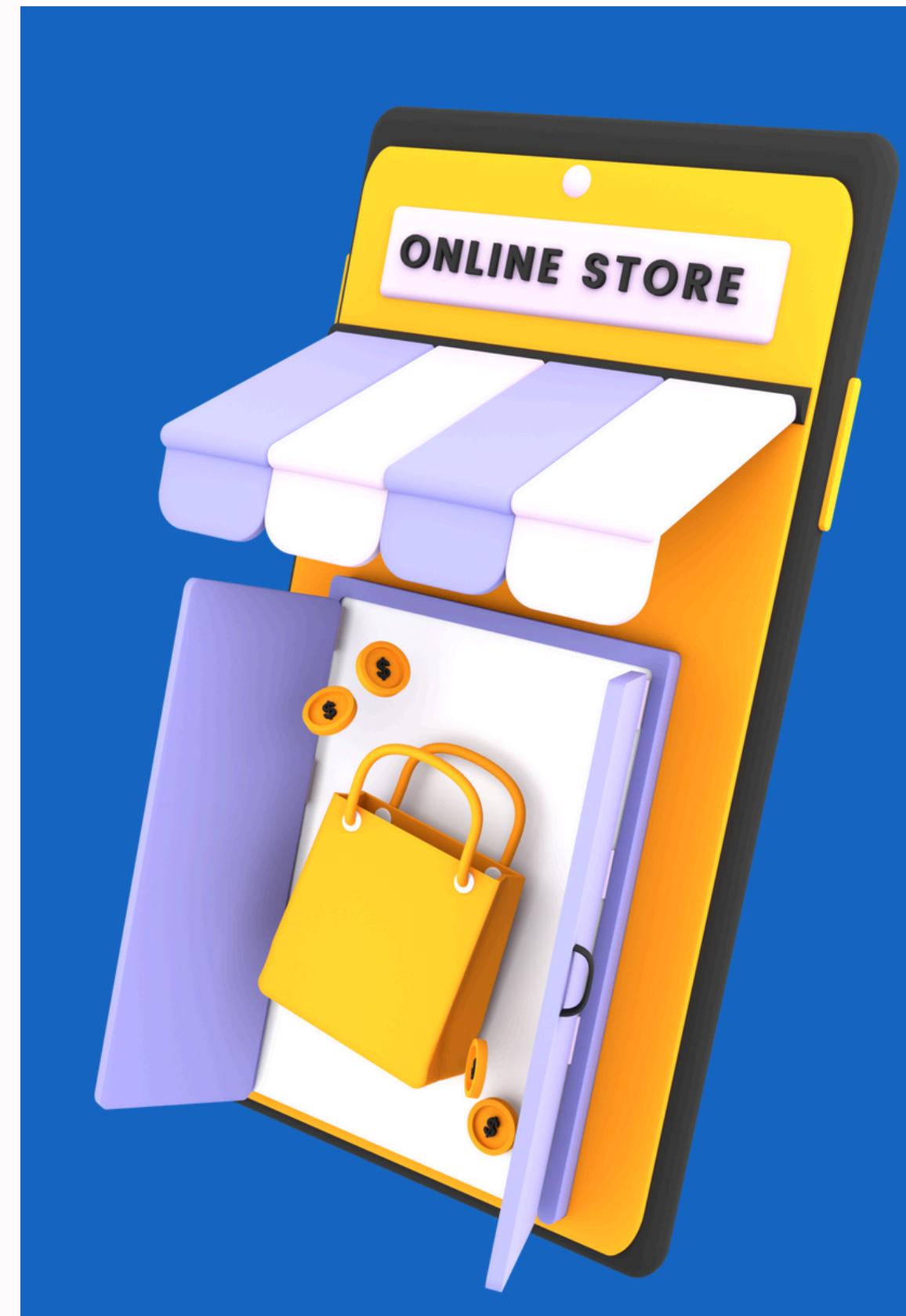
IMPLEMENTATION

www.bestbrightness.com



Content

- 01** Overview
- 02** Implementation Environment
- 03** Implementation Approach
- 04** Coding and Development
- 05** Database Implementation
- 06** Testing and Debugging
- 07** Security Considerations
- 08** Deployment
- 09** User Manual & Documentation
- 10** Future Enhancements



Overview

The system aims to streamline Best Brightness Pongola Shop's operations through a web-based solution. Key features include inventory management for real-time stock tracking, customer management to enhance service, and sales tracking with detailed performance reports.

Implemented Features

01

INVENTORY MANAGEMENT



The system tracks stock levels in real-time, allowing the shop to monitor inventory status. Basic automated notifications for low stock.

02

CUSTOMER MANAGEMENT



Database store basic customer profiles with contact details, payments and order histories.

03

SALES TRACKING



The system records sales transactions, providing insights into daily sales performance, identify top-selling products and generates basic reports.

E-COMMERCE PLATFORM:



Customers can view products, add items to their shopping cart, and easily proceed to checkout for purchase.

04

Limitations

01

INVENTORY AUTOMATION



While low-stock notifications are in place, full automation for reordering from suppliers has not yet been integrated.

Limited to basic customer info; Future upgrades will be segmentation and personalized recommendations to enhance customer engagement.



02

03

E-COMMERCE



The system currently lacks guest checkout options, which can deter new customers who prefer a quick purchasing process.

The system lacks advanced tools like trend prediction, limiting its ability to analyze consumer behavior and anticipate market shifts.

SALES ANALYTICS



04

Future Developments

01

E-COMMERCE EXPANSION



Implement a streamlined guest checkout process to boost conversion rates, and express checkout for added convenience.

SUPPLIER INTEGRATION



Automation of inventory restocking, allowing for direct communication with suppliers when stock levels are low.

03

ENHANCED CRM TOOLS



Future upgrades will enhance engagement with advanced segmentation and personalized recommendations.

02

COMPREHENSIVE REPORTING & ANALYTICS



Future developments will incorporate advanced trend prediction tools to enhance consumer behavior analysis and better anticipate market shifts.

04

IMPLEMENTATION ENVIRONMENT

Hardware Specifications, Software & Tools, and Programming Languages

Development Workstations

Processor:

Intel Core i5 (or equivalent) for optimal performance during development.

RAM:

16GB DDR4 to handle the demands of compiling code, running development environments, and testing.

Storage:

512GB SSD for fast access to development files and tools

Graphics:

Integrated GPU sufficient for typical ASP.NET development tasks.

OS:

Windows 10/11 for compatibility with Visual Studio and other Microsoft development tools.

Servers

Development Server:

Local and Cloud server running SQL Server 2019

Testing Server:

Sufficient hardware and security, along with a dedicated testing database. Focus on performance & functional.

Database Server:

It manages data storage, retrieval, and processing, ensuring efficient data handling.

Development Tools

Visual Studio 2022:

IDE for developing the ASP.NET project.

SQL Server (SSMS):

Database management, querying, and performance tuning.

Coding

C-Shape (C#)

Backend logic and ASP.NET Core MVC.

CSHTML/CSS

Designing the frontend of the web application.

JavaScript:

Client-side scripting and interactions (along with libraries like jQuery).

Third-Party Libraries

Bootstrap 5:

Frontend CSS framework used for responsive and mobile-first design.

Version Control:

Git was used to manage the development process, allowing for efficient tracking of changes, collaboration, and rollback if needed.

IMPLEMENTATION APPROACH

Methodology

Methodology

The development of the Best Brightness Pongola Shop system followed the *Agile methodology, a flexible, iterative approach to software development.*

Justification for Agile

Agile provided flexibility to adapt to changing business needs with frequent sprints, allowing for early testing and continuous feedback. This ensured timely development and refinement of key features like inventory management and sales tracking.

Justification for Agile

Phase 1: Requirements Gathering and Planning

Gathered business requirements from stakeholders and created a project backlog with user stories for key features like inventory, customer, and sales management.

Phase 2: Data Layer Development

Designed the database schema for products, customers, and transactions. Integrated Entity Framework for database interactions and seeded sample data for testing.

Phase 3: User Interface Development

Designed and implemented the web interface using **ASP.NET Core MVC** and **Bootstrap**, focusing on user-friendliness and creating basic product browsing and shopping cart pages.

Phase 4: Core Feature Development

Developed inventory management, customer profile tracking, and sales logging with reports. Launched the initial e-commerce product browsing and cart functionality.

Phase 5: Testing and Integration

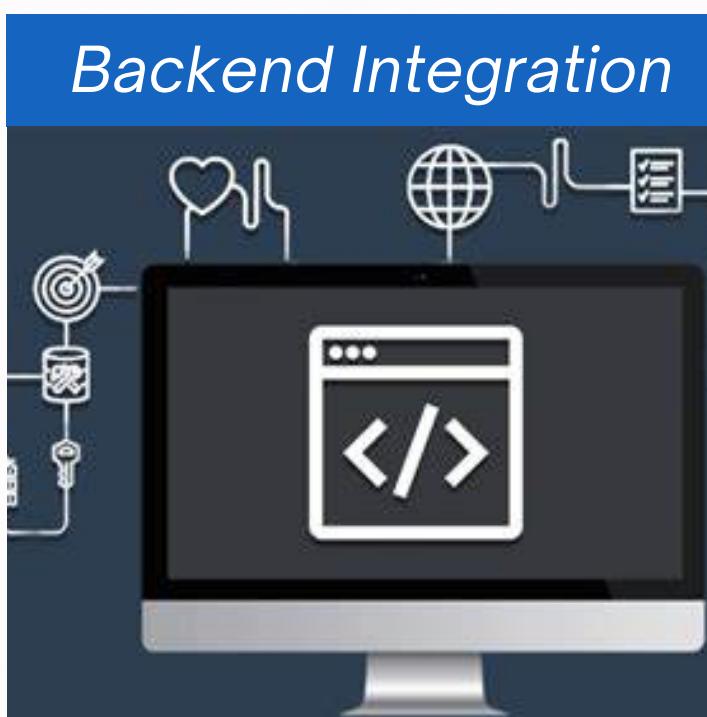
Conducted unit and integration testing to ensure all components worked seamlessly, followed by user acceptance testing with stakeholders.

Phase 6: Final Refinement & Deployment Preparation

Fixed bugs, optimized performance, prepared user documentation, and configured the system for deployment, including database backups and environment setup.

Division of Work

Team members were assigned specific tasks based on their expertise. One focused on database design and backend integration, while another handled frontend development and user interface design.



Mr. MS Mthethwa

Miss. NA Nxumalo

Mr. NT Mgabhi & BZS Khumala

Mr. N Mbambo

CODING AND DEVELOPMENT

Core Functionality:

The system manages sales, inventory, and suppliers with **CRUD operations**, includes predictive analytics for inventory forecasting, secure user authentication, and APIs for accessing data.

Modularization:

The system is divided into modules for authentication, sales and inventory management, forecasting, supplier management, and reporting. *Common utilities handle logging and error management across all modules.*

Project Structure

Controllers:

Handle **HTTP** requests and define actions (e.g., SalesController, InventoryController).

Models:

Define core data structures like Product, Order, Supplier, used with Entity Framework.



Views

Razor views for UI, organized by related controller (e.g., Views/Sales, Views/Inventory).

Services:

Business logic, like InventoryService for stock management and ForecastingService for analytics.

Project Structure

Design Patterns:

MVC, Repository Pattern, Dependency Injection, and asynchronous programming for better performance and maintainability.

DATABASE IMPLEMENTATION

Schema Implementation

Tables and Structure:

Payments

•	PaymentId (PK, int, not null)
•	CustomerId (int, not null)
•	OrderId (int, not null)
•	CardName (nvarchar(50), not null)
•	CardNumber (nvarchar(16), not null)
•	ExpiryDate (nvarchar(5), not null)
•	CVV (nvarchar(3), not null)
•	BillingAddress (nvarchar(max), not null)
•	PaymentDate (datetime2(7), not null)

QuotationRequests

•	Id (PK, int, not null)
•	FullName (nvarchar(max), not null)
•	Email (nvarchar(max), not null)
•	ProductId (int, not null)
•	Quantity (int, not null)
•	Message (nvarchar(max), not null)

Quotations

•	QuotationId (PK, int, not null)
•	ProductDetails (nvarchar(max), not null)
•	Quantity (int, not null)
•	CustomerName (nvarchar(100), not null)
•	CustomerEmail (nvarchar(100), not null)
•	CustomerPhone (nvarchar(20), not null)
•	CustomerAddress (nvarchar(200), not null)
•	RequestDate (datetime, not null)
•	Feedback (nvarchar(500), null)

Reviews

•	Id (PK, int, not null)
•	ProductId (int, not null)
•	Content (nvarchar(max), not null)
•	DateSubmitted (datetime2(7), not null)
•	Rating (decimal(18,2), not null)

WarehouseRegisterViewModels

•	Id (PK, int, not null)
•	Name (nvarchar(max), not null)
•	Email (nvarchar(max), not null)
•	Password (nvarchar(max), not null)
•	ConfirmPassword (nvarchar(max), null)

WarehouseLoginViewModels

•	Id (PK, nvarchar(450), not null)
•	Email (nvarchar(450), not null)
•	Password (nvarchar(max), not null)

DATABASE IMPLEMENTATION

Schema Implementation

Tables and Structure:

Customers

Column Name	Data Type	Allow Nulls
CustomerId	int	<input type="checkbox"/>
FirstName	nvarchar(50)	<input type="checkbox"/>
LastName	nvarchar(50)	<input type="checkbox"/>
Email	nvarchar(255)	<input type="checkbox"/>
Password	nvarchar(100)	<input type="checkbox"/>
RegistrationDate	datetime	<input type="checkbox"/>
Phone	nvarchar(100)	<input checked="" type="checkbox"/>
Address	nvarchar(200)	<input checked="" type="checkbox"/>
DateOfBirth	datetime	<input checked="" type="checkbox"/>
Gender	nvarchar(10)	<input checked="" type="checkbox"/>

Carts:

- **UserId (PK, nvarchar(255), not null)**
- **CustomerId (int, not null)**
- **Discount (decimal(18,2), not null)**

CartItems:

- **Id (PK, int, not null)**
- **ProductId (int, not null)**
- **ProductName (nvarchar(255), not null)**
- **ImagePath (nvarchar(255), null)**
- **Price (float, not null)**
- **Quantity (int, not null)**
- **CartUserId (nvarchar(255), not null)**
- **Total (Computed, float, not null)**
- **CartId (nvarchar(255), null)**
- **Discount (float, null)**

Orders

- **OrderId (PK, int, not null)**
- **OrderDate (datetime2(7), not null)**
- **TotalAmount (float, not null)**
- **CustomerId (int, not null)**
- **ShippingAddress (nvarchar(200), not null)**
- **Subtotal (float, not null)**
- **Tax (float, not null)**
- **Total (float, not null)**
- **IsConfirmed (bit, null)**
- **Feedback (nvarchar(max), null)**
- **IsCanceled (bit, not null)**
- **DeliveryDate (datetime2(7), null)**

OrderItem:

- **OrderItemId (PK, int, not null)**
- **OrderId (int, not null)**
- **ProductId (int, not null)**
- **ProductName (nvarchar(max), not null)**
- **Price (float, not null)**
- **Quantity (int, not null)**

DATABASE IMPLEMENTATION

Schema Implementation

Tables and Structure:

WarehouseUsers

•	Id (PK, int, not null)
•	Name (nvarchar(max), not null)
•	Email (nvarchar(max), not null)
•	PasswordHash (nvarchar(max), not null)

WarehouseProducts

•	Id (PK, nvarchar(450), not null)
•	Name (nvarchar(max), not null)
•	Quantity (int, not null)
•	Supplier (nvarchar(max), not null)
•	ReceivedDate (datetime2(7), not null)
•	ImageUrl (nvarchar(max), not null)
•	OriginalQuantity (int, not null)

WarehouseOrders

•	Id (PK, nvarchar(450), not null)
•	ProductId (nvarchar(max), not null)
•	Quantity (int, not null)
•	Supplier (nvarchar(max), not null)
•	OrderDate (datetime2(7), not null)
•	Status (nvarchar(max), not null)

Pickups

•	Id (PK, nvarchar(450), not null)
•	ProductId (nvarchar(max), not null)
•	Quantity (int, not null)
•	PickupDate (datetime2(7), not null)
•	Status (nvarchar(max), not null)
•	DeclineReason (nvarchar(max), not null)

Stocks

•	Id (PK, nvarchar(450), not null)
•	Name (nvarchar(max), not null)
•	OriginalQuantity (int, not null)
•	Quantity (int, not null)
•	Supplier (nvarchar(max), not null)
•	ReceivedDate (datetime2(7), not null)
•	ImageBase64 (nvarchar(max), not null)

StockItems

•	StockItemId (PK, int, not null)
•	ProductName (nvarchar(max), not null)
•	QuantityOrdered (int, not null)
•	UnitPrice (decimal(18,2), not null)

DATABASE IMPLEMENTATION

Schema Implementation Tables and Structure:

Suppliers

•	Id (PK, nvarchar(450), not null)
•	Name (nvarchar(max), not null)
•	ContactInfo (nvarchar(max), not null)
•	Address (nvarchar(max), not null)
•	AverageDeliveryTime (float, null)
•	AverageRating (float, null)
•	FulfillmentAccuracy (float, null)

SupplierRatings

•	Id (PK, int, not null)
•	SupplierId (FK, nvarchar(450), not null)
•	StoremanId (nvarchar(max), not null)
•	Rating (int, not null)
•	Comments (nvarchar(max), not null)
•	RecordedDate (datetime2(7), not null)

SupplierPerformances

•	Id (PK, int, not null)
•	SupplierId (FK, nvarchar(450), not null)
•	OrderId (int, not null)
•	DeliveryTime (float, not null)
•	FulfillmentAccuracy (bit, not null)
•	ProductQualityRating (float, null)
•	RecordedDate (datetime2(7), not null)

	Column Name	Data Type	Allow Nulls
•	Id	int	<input type="checkbox"/>
•	ProductId	int	<input type="checkbox"/>
•	Quantity	int	<input type="checkbox"/>
•	PricePerUnit	decimal(18, 2)	<input type="checkbox"/>
•	TotalPrice	decimal(18, 2)	<input type="checkbox"/>
•	SaleDate	datetime2(7)	<input type="checkbox"/>
•	SalespersonId	nvarchar(MAX)	<input type="checkbox"/>
•	Invoiceld	int	<input checked="" type="checkbox"/>

OrderItem

	Column Name	Data Type	Allow Nulls
•	OrderItemId	int	<input type="checkbox"/>
•	OrderId	int	<input type="checkbox"/>
•	ProductId	int	<input type="checkbox"/>
•	ProductName	nvarchar(MAX)	<input type="checkbox"/>
•	Price	float	<input type="checkbox"/>
•	Quantity	int	<input type="checkbox"/>

DATABASE IMPLEMENTATION

Schema Implementation

Tables and Structure:

Column Name	Data Type	Allow Nulls
StockLevel	int	<input type="checkbox"/>
Price	decimal(18, 2)	<input type="checkbox"/>
LastUpdatedDate	datetime	<input type="checkbox"/>
LastUpdatedBy	nvarchar(100)	<input checked="" type="checkbox"/>
SalespersonId	nvarchar(MAX)	<input type="checkbox"/>
Category	nvarchar(MAX)	<input type="checkbox"/>
Rating	decimal(18, 2)	<input type="checkbox"/>
ReviewCount	int	<input type="checkbox"/>
QuotationRequestId	int	<input checked="" type="checkbox"/>
Discount	decimal(5, 2)	<input checked="" type="checkbox"/>
ImagePath	nvarchar(255)	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
StockItemId	int	<input type="checkbox"/>
ProductName	nvarchar(MAX)	<input type="checkbox"/>
QuantityOrdered	int	<input type="checkbox"/>
UnitPrice	decimal(18, 2)	<input type="checkbox"/>

Column Name	Data Type	Allow Nulls
OrderId	int	<input type="checkbox"/>
OrderDate	datetime2(7)	<input type="checkbox"/>
TotalAmount	float	<input type="checkbox"/>
CustomerId	int	<input type="checkbox"/>
ShippingAddress	nvarchar(200)	<input type="checkbox"/>
Subtotal	float	<input type="checkbox"/>
Tax	float	<input type="checkbox"/>
Total	float	<input type="checkbox"/>
IsConfirmed	bit	<input checked="" type="checkbox"/>
Feedback	nvarchar(MAX)	<input checked="" type="checkbox"/>
IsCanceled	bit	<input type="checkbox"/>
DeliveryDate	datetime2(7)	<input checked="" type="checkbox"/>



QUERIES AND PROCEDURES

```
CREATE TABLE [dbo].[Products]([Id] [int] IDENTITY(1,1) NOT NULL,[Name] [nvarchar](100) NOT NULL,[Description] [nvarchar](max) NULL,[StockLevel] [int] NOT NULL,[Price] [decimal](18, 2) NOT NULL,[LastUpdatedDate] [datetime] NOT NULL,[LastUpdatedBy] [nvarchar](100) NULL,[SalespersonId] [nvarchar](max) NOT NULL,[Category] [nvarchar](max) NOT NULL,[Rating] [decimal](18, 2) NOT NULL,[ReviewCount] [int] NOT NULL,[QuotationRequestId] [int] NULL,[Discount] [decimal](5, 2) NULL,[IsDamaged] [bit] NULL,[IsMarkedAsDamaged] [bit] NULL,[DamagedStockLevel] [int] NULL,[UndamagedStockLevel] [int] NULL,[LossAmount] [decimal](18, 2) NULL,[ProductImage] [varbinary](max) NOT NULL,PRIMARY KEY CLUSTERED([Id] ASC)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]GO
```

IMPLEMENTATION ENVIRONMENT

Data Management

Data Validation

Server-Side: ASP.NET Core validation using data annotations (e.g., `[Required]`, `[Range]`) and custom checks in controllers.

Client-Side: Used JavaScript and HTML5 for *immediate user feedback*, complemented by ASP.NET Core's built-in validation.

Data Storage

Database: SQL Server was selected for its robustness. Proper indexing and foreign key constraints ensured data integrity and query performance.

Data Consistency: Employed SQL transactions for atomic operations and maintained referential integrity.

Data Retrieval:

Data Access Layer: Used Entity Framework Core for simplified data management and LINQ queries for efficient data retrieval.

API Integration: Developed APIs for seamless data exchange between the ASP.NET application and external modules.



Backup and Recovery

Backup Mechanisms

Automated Backups: Scheduled full and differential backups with SQL Server Agent, stored securely.

On-Demand Backups: Manual backups before major operations to prevent data loss.

Recovery Mechanisms

Restoration Process: Utilized SQL Server Management Studio for data recovery and enabled point-in-time recovery using transaction logs.

Disaster Recovery Plan: Offsite backups and regular testing of recovery processes ensured data safety and quick restoration capabilities.

DATABASE IMPLEMENTATION

Testing and Debugging

Unit Testing :

Overview

Unit testing validates that individual components or functions of the system work correctly in isolation, ensuring each part performs its intended function effectively.

Testing Frameworks:

For ASP.NET Core, unit tests were conducted using frameworks such as xUnit, NUnit, or MSTest.

Testing Approach:

ASP.NET Core:

Controllers were tested for expected results and scenario handling, services confirmed correct business logic implementation, and data access methods ensured proper database interaction.

Integration Testing:

Overview

Integration testing verifies that different system modules work together as intended, ensuring combined functionalities are correctly implemented.

Testing Approach:

ASP.NET Core:

Database interactions were tested for proper execution of CRUD operations and data retrieval, while end-to-end scenarios validated complete workflows across multiple components.

DATABASE IMPLEMENTATION

Testing and Debugging

User Testing/Acceptance Testing

Overview

User testing, or acceptance testing, confirms that the system meets end users' needs and expectations through real user interactions to validate functionality and usability.

Testing Approach:

Scenario-Based Testing:

: Users performed specific tasks, such as generating forecasts and downloading files, to verify the system's functionality.

Feedback Collection:

Collected user feedback on usability, performance, and feature accuracy.

Acceptance Criteria:

Forecast Accuracy:

Ensured forecasts generated matched expected results based on historical data.

UI Functionality:

Confirmed that user interface elements, such as download buttons and visualizations, functioned correctly and provided accurate data

SECURITY CONSIDERATIONS

Authentication:

ASP.NET Identity:

Used ASP.NET Identity for user authentication, providing secure login and account management with usernames and hashed passwords.

Password Hashing:

Passwords are securely hashed using a strong algorithm (e.g., PBKDF2) before storage, ensuring user passwords remain protected even if the database is compromised.

Authorization:

Role-Based Access Control (RBAC):

Implemented role-based authorization to restrict access to various parts of the application based on user roles (e.g., admin, manager, customer). Users are assigned roles, and permissions are enforced based on these roles.

Claims-Based Authorization:

Implemented claims-based authorization for granular user permissions, defining user attributes and permissions to ensure flexible and secure access control.

Encryption:

Data at Rest:

Sensitive data in the database is encrypted with Transparent Data Encryption (TDE) or column-level encryption in SQL Server to prevent unauthorized access.



Encryption of Sensitive Information:

Specific fields such as credit card numbers or personal identification details are encrypted before storing them in the database to add an extra layer of security.

Input Validation:

Server-Side Validation:

User inputs are validated on the server side to prevent injection attacks like SQL injection and XSS, ensuring that only valid data is processed.

Client-Side Validation:

Complementary client-side validation provides immediate feedback to users, enhancing the experience and catching errors before they reach the server.



SECURITY CONSIDERATIONS



Data Masking:

Masking Sensitive Data:

Sensitive data, such as user credit card numbers, is masked in user interfaces and logs to prevent unauthorized exposure.



Penetration Testing:

Internal Penetration Testing:

Performed by the development team to identify potential security issues from within the organization, ensuring that internal threats are also considered.



DEPLOYMENT

Deployment Plan

The system was deployed to production through a planned process, ensuring stability and performance, with the server set up, the ASP.NET Core app configured, and all dependencies integrated.

Server Setup:

The production server was set up with the required hardware and software to support the ASP.NET Core application and its predictive analytics components.

Application Deployment:

The application was deployed using a CI/CD pipeline for a smooth process, which included building, testing, and deploying to the production server.



Static Files and Resources:

All static files, including forecast plots and CSVs, were placed in wwwroot to ensure proper serving by the application.

Static Files and Resources:

Environment-specific settings like connection strings, API keys, and logging levels were configured via environment variables and files to ensure proper behavior in production.

Testing and Validation:

After deployment, the system was tested in production to confirm all functionalities, including forecasting, visualizations, and file downloads, worked as expected.

DEPLOYMENT

Installation/Setup Instructions

Server Preparation:

- Installed the ASP.NET Core runtime.
- Set up the web server (IIS, Nginx) with firewalls and SSL for HTTPS.



Testing:

- Verified the deployment by accessing the live application.



Application Deployment:

- Deployed via CI/CD, pulling the latest code, building, and deploying to the server.



Monitoring and Maintenance:

- Set up tools to monitor performance, server health, and error logs.
- Scheduled regular backups and updates to maintain system stability and security.



Environment Configuration:

Set environment-specific settings in appsettings.json and environment variables.



FUTURE ENHANCEMENTS

E-commerce Checkout and Payment Integration:

Online Payment Processing:

Integrated a secure payment gateway (e.g., Stripe, PayPal) for customers to make payments directly on the website.

Order Confirmation and Receipt Generation:

Implemented functionality to send order confirmation emails and generate electronic receipts for completed purchases.

Advanced Reporting and Analytics:

Detailed Sales Reports:



Created advanced reports and dashboards to analyze sales trends, customer behavior, and inventory turnover.



Best
Brightness

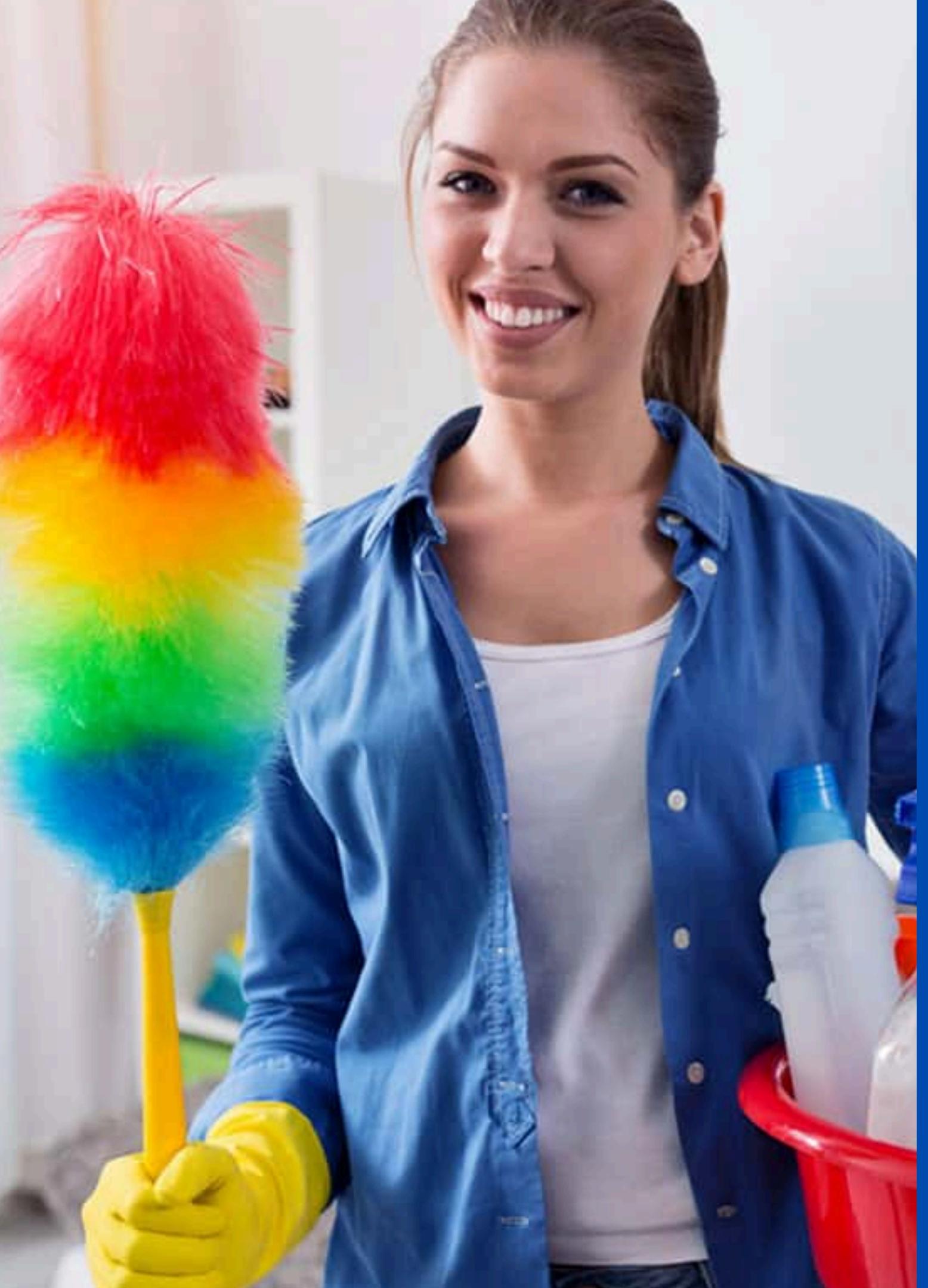
SYSTEM DESIGN

Presented by: Innovation Nexus



Overview

- Introduction 01
- System Architecture 02
- External Interfaces 03
- Database Design 04
- User Interface Design 05
- Application Components Design 06
- Design Considerations 07
- Clients 08
- Projects 09





Introduction

Purpose

To develop a streamlined system for managing the Bright Brightness Shop, focusing on inventory, sales, customer management, and quotations.



Scope

- Inclusions: Inventory tracking, sales processing, customer database, quotation management.
- Exclusions: Mobile app development, external integrations.



Overview

The presentation will cover system architecture, external interfaces, database design, user interface design, and key design considerations like performance, security, and scalability.

High-Level System Architecture

System Overview:

- A 3-tier architecture is implemented, separating concerns between:
 - a. Presentation Layer: User interfaces via web browsers.
 - b. Business Logic Layer: ASP.NET Core handling application logic.
 - c. Data Access Layer: SQL Server managing data storage and retrieval.

Hardware Architecture

- User Devices: Laptops, desktops accessing the system via a web browser.
- Server: Hosting the system on a central server with SQL Server as the database engine.
- Network Devices: Routers, switches, Ethernet cables, and Wi-Fi connections.

Software Architecture

- Backend: ASP.NET Core framework for business logic and data handling.
- Frontend: HTML, CSS, JavaScript for web-based user interface.
- Database: SQL Server, utilizing Entity Framework for object-relational mapping.
- Web Hosting: IIS for serving the web application.

Network Architecture

- Communication:
- HTTPS for secure communication between clients and the server.
- TCP/IP for internal network communications.
- Network Infrastructure:
- Connected through Ethernet and Wi-Fi.

External Interfaces - Hardware

- User Devices: Laptops, desktops interacting with the system through input/output peripherals.
- Server Interaction: HTTP requests to and from the server.
- Protocols: USB connections for local device setups.

External Interfaces - Software

- APIs: REST APIs for interaction between the frontend and backend.
- Entity Framework: For interaction with the SQL Server database.
- Data Formats: JSON for data exchange between system components.

External Interfaces - Communication and Security

- Communication:
- HTTPS for secure external communication.
- TCP/IP for internal communications.
- System Security:
- ASP.NET Identity for role-based access control (RBAC).
- SSL/TLS encryption for secure data transmission.
- Input validation and regular backups.

Database Design - Physical Design and Data Security

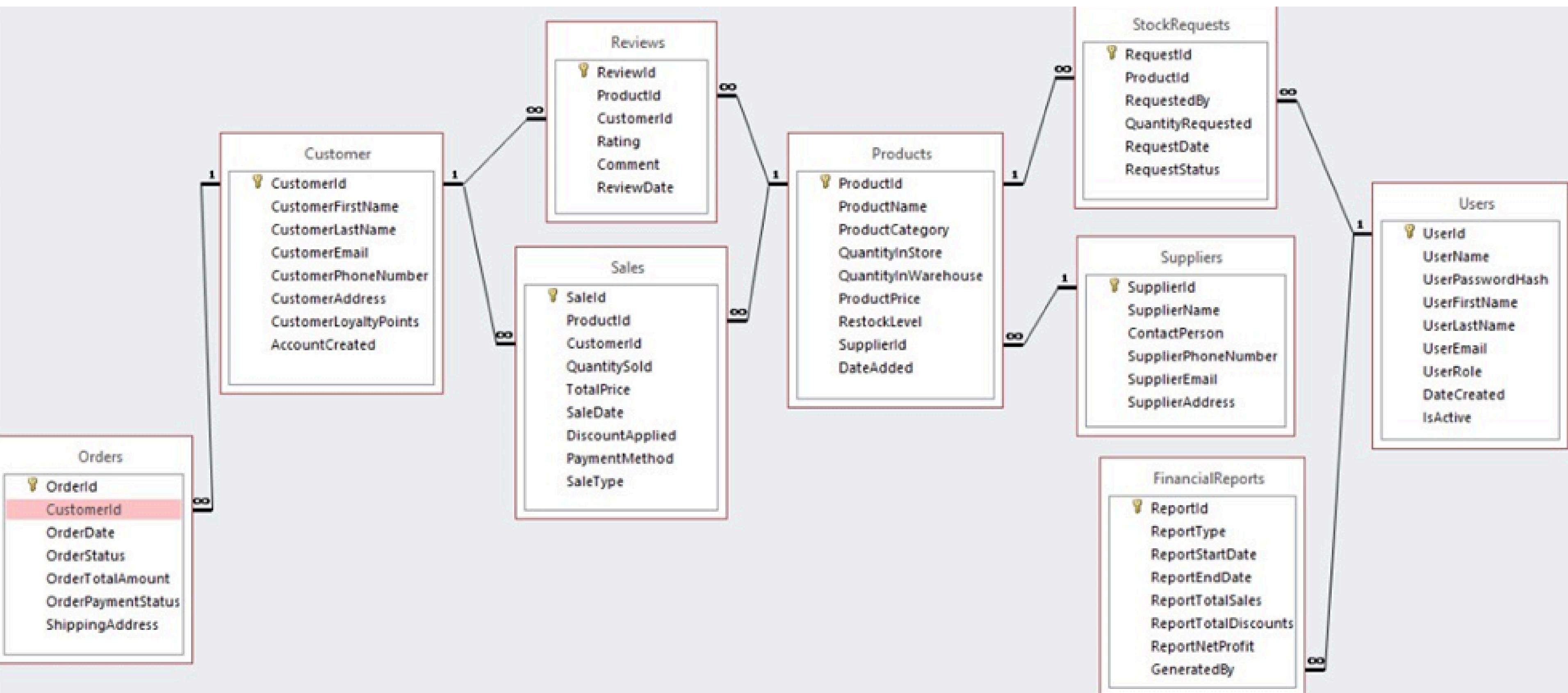
- **Physical Design:**
 - SQL Server as the primary database engine.
 - Data storage includes organized tables and indexed fields for faster access.
- **Data Security:**
 - Encryption of sensitive data (e.g., customer payment info).
 - Role-based access control for administrators, managers, and staff.
 - Regular backups to ensure data recovery.

Database Design - Data Requirements

- Data Dictionary:
- Key fields: Products, Sales, Customers, Quotations.
- Relationships: Products linked to sales and quotations.
- Example Fields: ProductID, CustomerID, SaleID, etc.

Database Design - Schema

- Schema Overview:
- Tables: Products, Sales, Customers, Quotations.
- Relationships: Foreign keys linking sales to products and customers.





Field Name	Data Type	Description (Optional)
UserId	AutoNumber	Unique identifier for each user.
UserName	Short Text	Username for login.
UserPasswordHash	Short Text	Hashed password for security.
UserFirstName	Short Text	User's first name.
UserLastName	Short Text	User's last name.
UserEmail	Short Text	User's email address.
UserRole	Short Text	Role of the user (e.g., Admin, Store Manager, Salesperson).
DateCreated	Date/Time	Date when the user account was created.
IsActive	Yes/No	Indicates if the user account is active.

Field Properties

General Logos

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

User Interface Design

- **Detailed User Interface Design:**
- Mockups:
 - Home page showing sales and inventory summaries.
 - Sales menu with interactive features for processing orders.
 - Customer management dashboard.
- **User Experience:**
- Designed for easy navigation, with a minimalistic layout to reduce complexity.
- Accessibility features for users with different needs.

[BestBrightness](#) [Home](#) [Privacy](#)

Hello, Thulani Mgabhi

[Online](#) | [Sales](#) | [Inventory](#) | [Report](#) | [Profile](#) | [Logout](#)

Welcome to Best Brightness



Daily Sales Tracking

Efficiently track daily sales with our intuitive dashboard.



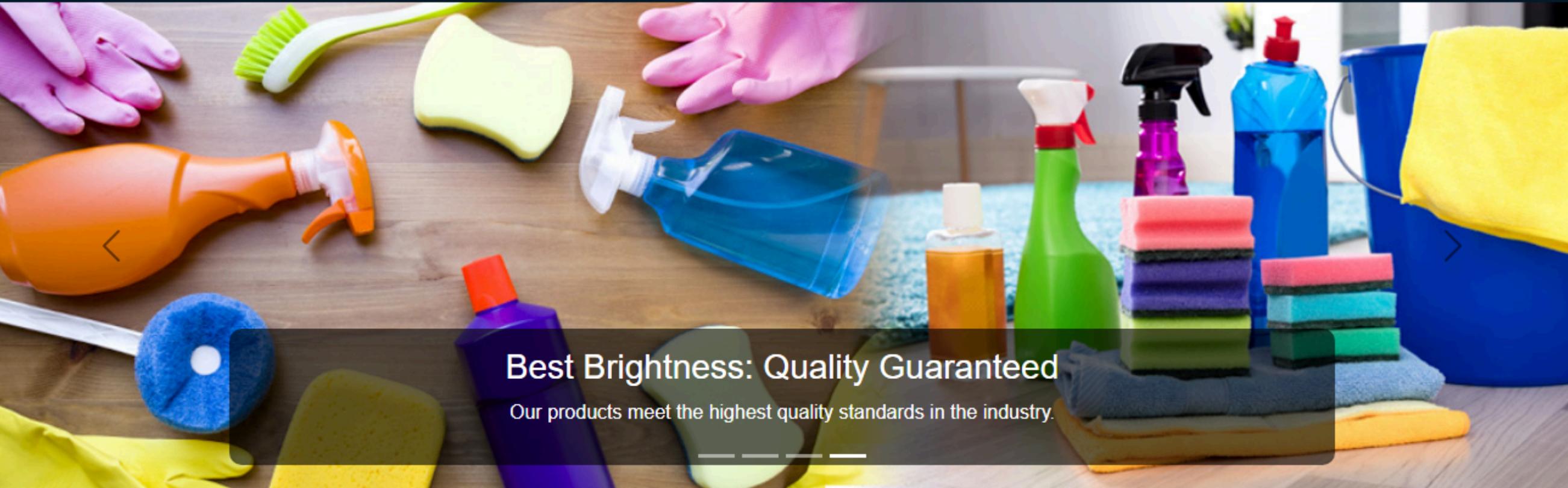
Manage Inventory

Keep your products organized and always in stock.



Monitor Employee Performance

Optimize your team's performance with insightful analytics.



Why Shop with BestBrightness



Fast Delivery

Nationwide shipping with guaranteed timely delivery.



Customer Support

Dedicated support team available to help you 24/7



Top Quality Products

We provide high-quality products for all your cleaning needs.

[Contact Us](#)

Shop 12, Pongola Shopping Centre, Jan Mielie St, Pongola, 2380

Email: info@bestbrightness.com

[Privacy Policy](#) [Terms of Service](#)

© 2024 Best Brightness. All rights reserved.





CLEANING SERVICE

Transform your space with top-tier cleaning expertise. Experience unmatched cleanliness and hygiene standards tailored to your needs. Elevate your environment today!

Call Us :
+27-63-456-7890

House Cleaning
 Office Cleaning
 Apartment Cleaning

@ingoudecompany www.ingoudecompany.com

Stainless Steel Vacumm - 2200W
R 3399,00 **R 1189,65**
0,00 ★ 0 Reviews
[Buy Now](#)

Windolene 750ml
R 65,00 **R 61,10**
4,90 ★ 10 Reviews
[Buy Now](#)

48V Portable Lithium Battery High Pressure Cleaner
R 149,00
2,80 ★ 10 Reviews
[Buy Now](#)

Dettol 6x500ml
R 149,00 **R 126,65**
4,90 ★ 10 Reviews
[Buy Now](#)

Handy Andy Probiotic
R 499,00
0,00 ★ 0 Reviews
[Buy Now](#)

Duck 500ml
R 37,00
5,00 ★ 1 Reviews
[Buy Now](#)

Sales

Salesperson: 207649 - Vergy Wax

Total Sales Price: R19 448,84

Product	Quantity	Total Price	Sale Date	Sale Time
Stainless Steel Vaccum - 2200W	3	R7 497,00	2024/09/20	18:45
Handy Andy Professional Refill Floor Cleaner and Degreaser 5L	1	R129,63	2024/09/20	18:45
Dettol 6x500ml	1	R279,65	2024/09/20	18:55
Handy Andy Professional Refill Floor Cleaner and Degreaser 5L	2	R259,26	2024/09/20	18:55
Stainless Steel Vaccum - 2200W	2	R4 998,00	2024/09/20	18:55
Windolene 750ml	60	R2 932,80	2024/09/20	19:45
Domestos Summer Multipurpose Stain Removal Thick Bleach Cleaner 6x500ml	30	R3 352,50	2024/09/20	19:45

Inventory

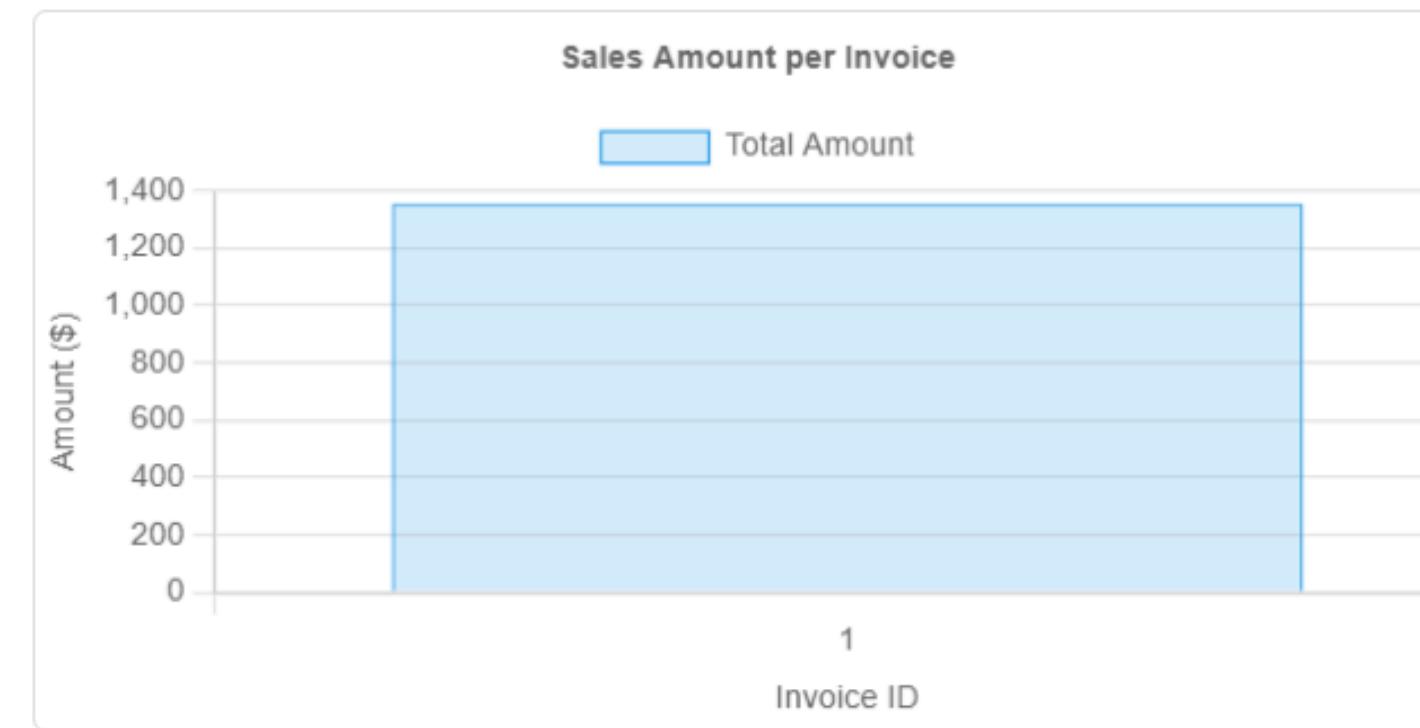
The following products are low in stock: Pledge Wood Polish Lemon Oil 300m

Product Image	Product Name	Description	Stock Level	Price	Last Updated By	Last Updated Date	Actions
	Stainless Steel Vacuum - 2200W	Flexible hose with 80 cm length is perfect for reaching tight corners and difficult-to-access areas, trust in the vacuum's durability and long-lasting use in your home.	46	R3 399,00	Admin	2024/09/29 18:27	<button>Update Stock</button>
	Windolene 750ml	Windolene has a Powerful 6-Action Shine System. Glass & Shiny Surface Cleaner - Trigger	49	R65,00	Admin	2024/09/29 18:40	<button>Update Stock</button>
	48V Portable Lithium Battery High	A powerful and portable solution for all your vehicle cleaning needs.	49	R149,00	Admin	2024/09/29 18:43	<button>Update Stock</button>

Reports Overview

[Download CSV](#)[Print Reports](#)

Sales Report

[Daily Sales](#)[All Sales](#)

All Sales Report

Inventory Report



Product	Stock Level	Needs Restocking
Stainless Steel Vacum - 2200W	46	Sufficient Stock
Windolene 750ml	49	Sufficient Stock
48V Portable Lithium Battery High	49	Sufficient Stock
Dettol 6x500ml	50	Sufficient Stock
Handy Andy Probiotic	45	Sufficient Stock
Duck 500ml	49	Sufficient Stock
Mr. Sheen Fast Acting Oven Cleaner 300ml x2	49	Sufficient Stock
Domestos 6x500ml	50	Sufficient Stock
Pledge Wood Polish Lemon Oil 300m	0	Low Stock - Restock Needed
Handy Andy Professional Refill Floor Cleaner and Degreaser 5L	50	Sufficient Stock
Ultimate Home Cleaning Combo	39	Sufficient Stock
Spray Mop with Brush Cleaner Water	40	Sufficient Stock
5x Citronol	40	Sufficient Stock
Dettol 1.5l	40	Sufficient Stock
Anti-Bacterial Wipes - 2000 Pack	38	Sufficient Stock



Welcome, Thulani Mgabhi

Warehouse Manager Dashboard

Search pickups, suppliers, stock...

Search

Total Stock Available

The total stock in your warehouse is: 1420

Stock levels are sufficient.

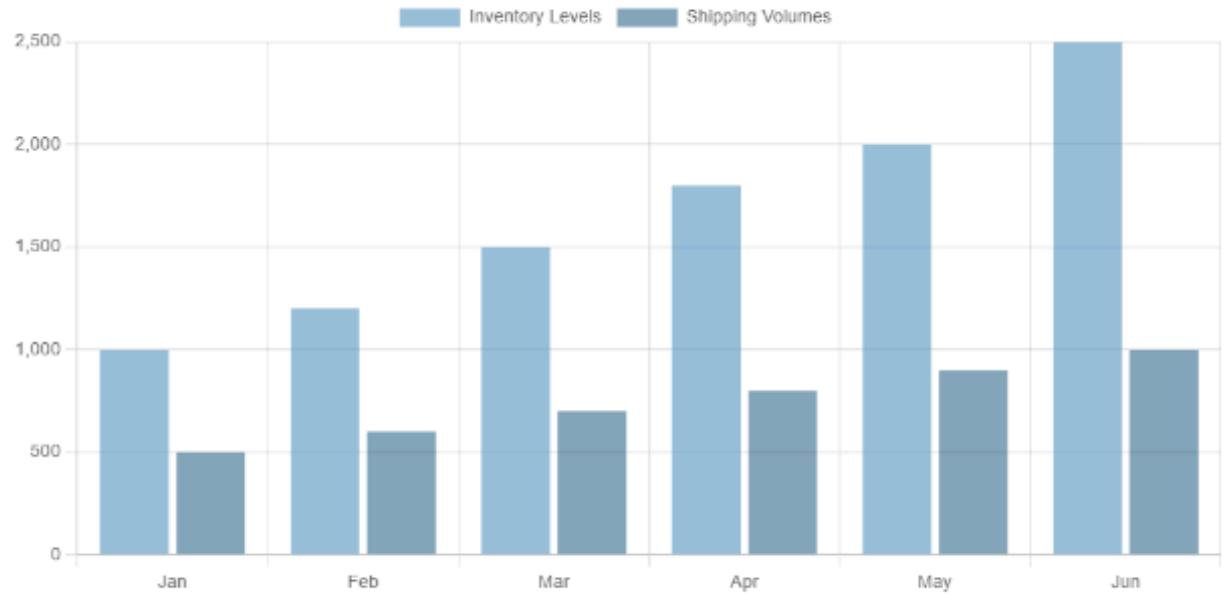
Manage Stock

Manage Pickups

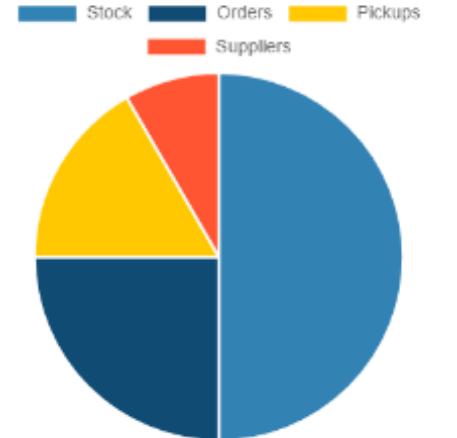
Handle Pickup Requests

Go to Pickups

Inventory and Shipping Statistics



Warehouse Overview



Create Orders

Place new orders for suppliers

Create Order

Suppliers

View and manage suppliers

Manage Suppliers

Record Sale

Product

Dettol 6x500ml

Quantity

2

Sale Date

2024/09/30

Salesperson ID

403518

Price Per Item

149.00

Discount (%)

15.00

Total Price

253.30

Add to Cart

https://localhost:7276/Sales/Cart

Brightness Home Privacy

Cart

Sale added to cart.

Product Name	Original Price	Discount (%)	Discounted Price	Quantity	Total Price	Action
Dettol 6x500ml	R149,00	15	R126,65	2	R253,30	<button>Remove</button>

Total Amount: R253,30

Amount Paid

300

Change

47.00

[Record Sale](#)

[Go to Record Sale](#)



Search here



3

Daily Orders



8

Sales



1

Quotations



R 10641,26

Earning



Recent Products

Product Image

Price

Stock Level

Discount

View All



R3 399,00

46

65,00%



R65,00

49

6,00%



R149,00

49

No Discount



R149,00

50

15,00%



R499,00

45

No Discount



R37,00

49

No Discount



R195,00

49

48,00%



R149,00

50

25,00%



R73,00

0

65,00%



R79,00

50

No Discount



R203,00

39

35,00%



R145,00

40

No Discount

Damaged Products

View All

Product	Quantity	Loss
	2	998,00
	1	145,00

Application Components Design

- **Component Overview:**
- Inventory Management: Tracks stock levels, damaged products.
- Sales Processing: Manages sales transactions and discounts.
- Customer Management: Tracks customer information and order history.
- Quotations: Generates and stores quotations for bulk orders.
- **Detailed Design:**
- Each component is modular, with clearly defined data structures and interactions with the database.

Manage Damaged Products

ID	Name	Description	Stock		Damaged		Undamaged Stock Level	Loss Amount	Price	Category	Is Damaged	Actions
			Level	Level	Stock	Stock Level						
1	Stainless Steel Vacuum - 2200W	Flexible hose with 80 cm length is perfect for reaching tight corners and difficult-to-access areas, trust in the vacuum's durability and long-lasting use in your home.	46	46	46	46	46	R3 399,00	No	<button>Update</button> <button>Mark as Damaged</button>	<input type="checkbox"/>	
2	Windolene 750ml	Windolene has a Powerful 6-Action Shine System. Glass & Shiny Surface Cleaner - Trigger	49	49	49	49	49	R65,00	No	<button>Update</button> <button>Mark as Damaged</button>	<input type="checkbox"/>	
3	48V Portable	A powerful and	49	49	49	49	49	R149,00	No	<input type="checkbox"/>	<input type="checkbox"/>	

Best Brightness Pongola Shop

Shop 12, Pongola Shopping Centre

Jan Mielie St, Pongola, 2380

Email: info@bestbrightness.co.za

Phone: +27 34 345 6789

Invoice Details

Invoice Date: 30 Sep 2024

Invoice Number: INV-16

Salesperson: Thulani Mgabhi (mrgeetwoo@gmail.com)

Amount Paid: R300,00

Change: R46,70

Product Name	Price	Quantity	Total Price
Dettol 6x500ml	R149,00	2	R253,30
Total Amount:			R253,30

Thank you for your purchase!

Design Considerations - Performance

- **Performance:**
 - Caching strategies for frequently accessed data.
 - Efficient database queries to reduce load times.
- **Scalability:**
 - System can handle additional users and more data as the business grows.
 - Support for load balancing to distribute traffic across servers.

View All Salespersons

[Register New Salesperson](#)

Salesperson ID	First Names	Last Name	Email	Password	Actions
434762	Vumani Sphe	Mthethwa	sphe@sphe.com	eda7bed3	Edit Delete

View All Employees

[Register New Employee](#)

Employee ID	First Names	Last Name	Email	Password	Actions
789959	Sanele	Shobede	sanele.shobede@gmail.com	5d0575b9	Edit Delete

Design Considerations - Security

- **Security:**
 - Implemented through SSL/TLS encryption for data in transit.
 - Secure login with multi-factor authentication (MFA) for administrators.
 - Regular security audits and vulnerability testing.

Login

mrmgabhint@gmail.com

.....

Log In

Don't have an account? [Register here](#)

Welcome!



First Name

Last Name

Email

Password

Confirm Password

Register

Register!



Only administrators are allowed to register. If you are not an admin, please ask an administrator to register you.



Quality You Can Trust

Login

Email:

mrgeetwoo@gmail.com

Password:

.....

Invalid login attempt.

[Login](#)

Don't have an account? [Register Here](#)



Quality You Can Trust

Register

Name

Email

Password

Confirm Password

Register

Already have an account? [Login Here](#)

[Log In](#) [Sign Up](#)



Log In



mrgeetwoo@gmail.com



.....

Login

[Forgot your password?](#)

[Log In](#)

[Sign Up](#)



Sign Up

 First Name

 Last Name

 mrgeetwoo@gmail.com



[Register](#)