**CASE STUDY**

**GROCERY MANAGEMENT SYSTEM**

**Prepared by**

**Jaswant Singh**

**Pawan Ramteke**

**Bhupendra Pawar**

**Table of Contents**

## Introduction

* 1. [Purpose 1](#_TOC_250082)
  2. [Document Conventions 1](#_TOC_250081)
  3. [Intended Audience and Reading Suggestions 2](#_TOC_250080)
  4. [Product Scope 2](#_TOC_250079)

1. [Overall Description](#_TOC_250077)­­­
   1. [Product Perspective](#_TOC_250076) 3
   2. [Product Functions 4](#_TOC_250075)
   3. [User Classes and Characteristics 5](#_TOC_250074)

[External Interface Requirements](#_TOC_250071)

* 1. [User Interfaces 5](#_TOC_250070)
  2. [Software Interfaces 7](#_TOC_250068)
  3. [Communications Interfaces 8](#_TOC_250067)

1. [System Features](#_TOC_250066)
   1. [User Registration and Login 9](#_TOC_250065)
      1. [Description 10](#_TOC_250064)
      2. [Priority 10](#_TOC_250063)
      3. [Functional Requirements 11](#_TOC_250061)
2. [Other Nonfunctional Requirements](#_TOC_250036)
   1. [Performance Requirements 12](#_TOC_250035)
      1. [Response Time 12](#_TOC_250034)
      2. [Search Performance 12](#_TOC_250033)
      3. [Concurrent Users 12](#_TOC_250032)
   2. [Safety Requirements 12](#_TOC_250031)
      1. [Data Security 12](#_TOC_250030)
      2. [Authorization 12](#_TOC_250029)
      3. [Privacy 12](#_TOC_250028)
   3. [Security Requirements 12](#_TOC_250027)
      1. [Data Transmission 12](#_TOC_250026)
      2. [Vulnerability Prevention 13](#_TOC_250025)
      3. [Access Control 131](#_TOC_250024)
   4. [Software Quality Attributes 13](#_TOC_250023)
      1. [Usability 13](#_TOC_250022)
      2. [Reliability 13](#_TOC_250021)
      3. [Maintainability 13](#_TOC_250020)
      4. [Testability 13](#_TOC_250019)
      5. [Portability 13](#_TOC_250018)
      6. [Adaptability 13](#_TOC_250017)
3. [Other Requirements](#_TOC_250016)
   1. [Legal and Compliance Requirements 14](#_TOC_250015)
      1. [Data Protection and Privacy 14](#_TOC_250014)
      2. [Intellectual Property Rights 14](#_TOC_250013)
      3. [User Agreement 14](#_TOC_250012)
   2. [Documentation Requirements 14](#_TOC_250011)
      1. [User Manual 14](#_TOC_250010)
      2. [Technical Documentation 14](#_TOC_250009)
   3. [Development and Testing Requirements 14](#_TOC_250008)
      1. [Development Environment 14](#_TOC_250007)
      2. [Testing Environment 14](#_TOC_250006)
      3. [Test Cases 14](#_TOC_250005)
   4. [Deployment Requirements 15](#_TOC_250004)
      1. [Server Configuration 15](#_TOC_250003)
      2. [Database Setup 15](#_TOC_250002)
      3. [Release Management 15](#_TOC_250001)
4. [Appendix A: Glossary 15](#_TOC_250000)

Appendix B: Analysis Model 16

- ER Diagram 16

**Software Requirements Specification (SRS)**

## Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) is to provide a comprehensive and detailed description of the CRUD Console App. This document outlines the functional and non-functional requirements, system constraints, and user interface design to guide the development and implementation of the application.

The CRUD Console App is designed to offer a simple Command Line Interface (CLI) for users to perform CRUD (Create, Read, Update, Delete) operations on a collection of products and manage user accounts. It allows users to interact with the application by logging in, registering new accounts, viewing available products, placing orders, and viewing order history. Additionally, administrators have extended privileges, such as adding, updating, and deleting products, as well as user management.

.

## Document Conventions

This Software Requirements Specification (SRS) document follows the following conventions:

Priority: Each requirement statement is assigned its own priority level (High, Medium, or Low) based on importance and urgency.

Formatting: Consistent formatting using headings, subheadings, and bullet points for improved readability and organization.

Terminology: Standard industry terminology and abbreviations are used. Specialized terms are defined in the Glossary (Appendix A).

References: External references are listed in the References section (Section 1.5) with proper citations.

Use of placeholders: "TBD" (To Be Determined) is used when specific information is not yet available.

Notation: Concise and clear notation for presenting requirements with unique requirement IDs.

These conventions ensure clarity and consistency throughout the document, enabling readers to understand and navigate the SRS effectively.

## Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is intended for developers, project managers, testers, documentation writers, and stakeholders.

Developers will find technical requirements and system design information.

Project managers will gain insights into the project scope, objectives, and constraints. Testers will find functional and non-functional requirements for validation.

Documentation writers will discover system features and user documentation requirements. Stakeholders will get an overview of the system's functionalities, scope, and benefits.

The document is organized as follows:

Section 1: Introduction provides an overview of the document.

Section 2: Overall Description describes the product perspective, functions, user classes, operating environment, and constraints.

Section 3: External Interface Requirements details the user, hardware, software, and communication interfaces.

Section 4: System Features lists and describes the major features.

Section 5: Other Nonfunctional Requirements covers performance, safety, security, and quality attributes.

Section 6: Other Requirements includes additional requirements. Appendices include a glossary and optional analysis models.

Start with the introduction sections to understand the purpose, conventions, and audience.

## Product Scope

The product scope for the CRUD Console App includes a clear description of the functionalities, features, and limitations of the application. It outlines what the application aims to achieve and what it does not include. Below is the product scope for the CRUD Console App:

**Product Description**

The CRUD Console App is a simple command-line application designed to manage users, products, and orders. The primary goal of the application is to provide basic functionalities for users to log in, register, view products, add/update/delete products (admin only), view orders, and place new orders. The app operates on a MySQL database to store and retrieve data.

**Features**

The application offers the following key features:

- User Management:

- Users can log in using their username and password.

- New users can register by providing a unique username and password.

- Product Management:

- All users can view a list of available products.

- Admin users can add new products by providing product name and price.

- Admin users can update existing product details, including name and price.

- Admin users can delete products from the system.

- Order Management:

- Users can view their order history.

- Users can place new orders by selecting a product from the available list.

## Overall Description

## Product Perspective

The product perspective provides an overview of the CRUD Console App in relation to its interaction with external systems, users, and other software components. It describes the context in which the application operates and the relationships it has with various entities. Below is the product perspective for the CRUD Console App:

**Product Interaction**

The CRUD Console App interacts with the following entities:

**Users**: Users interact with the application through the command-line interface. They can log in, register, view products, place orders, and view their order history.

**Admin Users**: Admin users are a specific type of users with elevated privileges. They can add, update, and delete products in addition to regular user functionalities.

**MySQL Database**: The application interacts with a MySQL database for storing user credentials, product details, and order information.

**External Systems**

The CRUD Console App operates independently on a local system. It does not interact with external systems, APIs, or web services.

**User Roles**

The application distinguishes between two types of users:

**Regular Users**: Regular users can access basic functionalities such as logging in, registering, viewing products, and placing orders.

**Admin Users**: Admin users have additional privileges to manage products, including adding, updating, and deleting products.

**Data Storage**

The CRUD Console App stores user data, product information, and order details in a MySQL database. The database schema includes tables for users, products, and orders.

## Product Functions

The core functions of the Online Grocery Management Console Application include:

### User Registration and Login

The User Login and Registration module is a crucial part of the CRUD Console App, providing secure access to users and enabling new users to create accounts. This module outlines the requirements for user authentication, registration, and data security.

The User Login and Registration module will allow users to perform the following actions:

**User Login**: Existing users can log in to the application using their registered credentials.

**User Registration**: New users can create an account by providing a unique username and password.

**User Requirements**

The User Login and Registration module must meet the following user requirements:

**UR01**: Users should be able to log in using their unique username and password.

**UR02**: New users can register by providing a unique username and password.

**UR03**: The system should validate user credentials during login and registration.

### User Profile Creation and Management

After successful login, users can create and manage their profiles. Users can add details about

Products and orders .

## User Classes and Characteristics

## 

## This section of the SRS document outlines the user classes and their characteristics within the CRUD Console App. The User Classes and Characteristics describe different types of users interacting with the application and their respective attributes.

## The User Classes and Characteristics identify the various user roles and their specific characteristics relevant to the CRUD Console App. The application distinguishes between regular users and admin users, each having distinct privileges and behavior.

## User Classes and Characteristics-

## The CRUD Console App includes the following user classes and their corresponding characteristics:

## - Characteristics:

## - Regular users are individuals who have registered in the application but do not have admin

## privileges.

## - They can log in using their unique username and password.

## - Regular users can view a list of available products.

## - They can place orders for products from the available list.

## Admin Users-

## Characteristics-

## - Admin users are a specific type of user with elevated privileges in the application.

## - They are responsible for managing products, including adding, updating, and deleting

## products.

## - Admin users can log in using special admin credentials (e.g., username: "admin" and password: "admin123").

## - Admin users have access to additional functionalities not available to regular users.

## User Roles and Privileges-

## Regular Users-

## View Products: Regular users can view a list of available products in the application.

## Place Orders: Regular users can place orders for products from the available list.

## User Interfaces

The user interface of the Grocery Management System Console Application is console-based. Users interact with the application by entering commands and receiving responses displayed in the console.

## Login Interface

Users will be prompted to enter their login credentials (username and password) to access the application.

## Registration Interface

Users can register by providing their Username and password, whereas the admin doesn’t need to register since we have kept just 1 admin as default.

## Product Search Interface

Users can search for Products in the Grocery store which have been added by the admin .If users want to see other products the admin needs to add more products into the store.

## Order Submission Interface

Users can Submit their products order ,which than can be displayed by the admin.

## Software Interfaces

**Software Interface**

**User Interface**

The CRUD Console App interacts with users through a command-line interface (CLI) in the console. Users can input commands and receive textual output.

**Login**

- The user is prompted to enter their username and password to log in.

- Upon successful login, the user gains access to additional functionalities based on their admin status.

- For unsuccessful login attempts, appropriate error messages are displayed.

**Register**

- The user can register as a new user by providing a unique username and password.

- Upon successful registration, the user is added to the user database and can log in with the newly created credentials.

**View Products**

- The app displays a list of products with their IDs, names, and prices.

**Add Product (Admin Only)**

- Admin users can add new products by entering the product name and price.

- The new product is added to the product database.

**Update Product (Admin Only)**

- Admin users can update an existing product by entering the product ID.

- The user is prompted to enter the new product name and price.

- The product details are updated in the product database.

**Delete Product (Admin Only)**

- Admin users can delete an existing product by entering the product ID.

- The selected product is removed from the product database.

**View Orders**

- The app displays a list of orders with their IDs, user IDs, product IDs, and order dates.

**Add Order**

- Users can add an order by selecting a product from the list and entering its ID.

- The app records the order with the user ID, product ID, and the current date.

**Database Interface**

The CRUD Console App interacts with a MySQL database to store and retrieve user, product, and order data. The app uses Entity Framework Core to manage the database operations.

**User Entity**

- The User entity has attributes User Id, Username, Password, and Is Admin.

- User Id is the primary key of the User table.

- Username and Password are required fields used for authentication.

- Is Admin is a boolean value indicating whether the user has admin privileges.

**Product Entity**

- The Product entity has attributes Product Id, Name, and Price.

- Product Id is the primary key of the Product table.

- Name and Price are required fields representing the product's name and price, respectively.

**Order Entity**

- The Order entity has attributes Order Id, User Id, Product Id, and Order Date.

- Order Id is the primary key of the Order table.

- User Id and Product Id are foreign keys referencing the User and Product tables, respectively.

- Order Date stores the date and time when the order is placed.

**Database Connection**

- The app connects to the MySQL database using a connection string specifying the server, port, database name, username, and password.

**Error Handling**

- The app handles and displays appropriate error messages for various scenarios, such as invalid input, product not found, user authentication failure, etc.

**Admin Privileges**

- Certain functionalities, such as adding, updating, and deleting products, are restricted to admin users only.

- The app checks the admin status of the logged-in user before allowing access to admin-only functionalities.

**Data Validation**

- The app validates user inputs to ensure data integrity and prevent database-related issues.

## Communications Interfaces

The communication interface in the context of the CRUD Console App refers to how the application communicates with external systems, services, or users. As it is a console application, the communication interface primarily involves interactions with the user through the command-line interface and with the MySQL database for data storage and retrieval.

Let's outline the communication interfaces for the CRUD Console App:

**User Interaction**

The CRUD Console App communicates with users through a command-line interface (CLI) within the console. The CLI provides a text-based interaction where users can input commands and receive textual feedback.

**Database Interaction**

The CRUD Console App communicates with a MySQL database using Entity Framework Core to perform CRUD operations for user, product, and order data.

**Database Connection**

- The app establishes a connection to the MySQL database using the provided connection string.

- The connection string contains information about the server, port, database name, username, and password required to connect to the database.

**Data Retrieval**

- When the app needs to retrieve data (e.g., products, orders), it sends queries to the database using Entity Framework Core's LINQ queries.

- The app communicates with the MySQL server over the network to fetch the required data.

**Data Modification**

- When the app needs to add, update, or delete data (e.g., add a product, update an order), it sends corresponding database modification commands to the MySQL server.

- The app communicates with the MySQL server over the network to apply the changes to the database.

## System Features

## User Registration and Login

## Description

This feature allows users to register as Users by providing their username, password. Registered users can log in to the application using their credentials.

## Priority

High

## Functional Requirements

Certainly! Below are functional requirements for the CRUD Console App project:

**User Management**

**User Registration**

- The system shall allow users to register for a new account.

-Users shall provide a unique username and a password during registration.

-The system shall validate the uniqueness of the entered username.

-User passwords shall meet the specified criteria, such as minimum length and character –n restrictions.

- Upon successful registration, the system shall create a new user account with the provided credentials and set the user as a non-admin user by default.

-If the user attempts to register with an existing username, the system shall display an appropriate error message indicating the unavailability of the chosen username.

**User Login**

- The system shall allow registered users to log in using their username and password.

- The system shall verify the entered credentials against the user database for authentication.

- If the entered credentials match an existing user account, the system shall grant access to the user and display a welcome message.

- If the entered credentials do not match any existing user account, the system shall display an appropriate error message indicating the login failure.

- The system shall differentiate between regular users and admin users based on the user's admin status.

- Admin users shall have additional privileges, including the ability to add, update, and delete products.

- By default, newly registered users shall be assigned a non-admin status.

Product Management

**View Products**

- The system shall allow users to view a list of products.

- The list shall display product IDs, names, and prices.

**Add Product (Admin Only)**

- The system shall allow admin users to add new products to the system.

- Admin users shall enter the product name and price to add a new product.

- Upon successful addition, the system shall update the product database with the new product details.

**Update Product (Admin Only)**

- The system shall allow admin users to update existing products.

- Admin users shall provide the product ID of the product to be updated.

- The system shall prompt for the new product name and price.

- Upon successful update, the system shall update the product database with the modified product details.

- If the provided product ID does not match any existing product, the system shall display an appropriate error message indicating that the product was not found.

**Delete Product (Admin Only)**

- The system shall allow admin users to delete existing products.

- Admin users shall provide the product ID of the product to be deleted.

- Upon successful deletion, the system shall remove the product from the product database.

- If the provided product ID does not match any existing product, the system shall display an appropriate error message indicating that the product was not found.

**Order Management**

**View Orders**

- The system shall allow users to view a list of orders.

- The list shall display order IDs, user IDs, product IDs, and order dates.

**Add Order**

- The system shall allow users to add new orders.

- Users shall select a product from the list of products to place an order.

- Upon order placement, the system shall record the order with the user ID, product ID, and the current date.

**Order Date**

- The system shall automatically timestamp the order date when a user places an order.

**Error Handling**

- The system shall handle and display appropriate error messages for various scenarios, such as invalid input, product not found, user authentication failure, etc.

- Error messages shall be informative and user-friendly to help users understand and resolve issues.

# Other Nonfunctional Requirements

## Performance Requirements

## Response Time

The application should respond to user interactions promptly. User commands and inputs should be processed within a maximum of 2 seconds to ensure a smooth user experience.

## Search Performance

The search functionality should return results within a reasonable time frame, depending on the database size and complexity of search criteria. The search response time should be kept under 3 seconds.

## Concurrent Users

The application should support multiple concurrent users without significant performance degradation.

## Safety Requirements

## Data Security

The application should ensure the security and confidentiality of user data, including login credentials and personal information. User passwords should be stored securely using encryption or hashing techniques.

## Authorization

Access to certain features, such as job posting, should require additional authorization to prevent unauthorized usage.

## Privacy

The application should enforce communication between users and products within the application to maintain user privacy and security.

## Security Requirements

## Data Transmission

User data should be transmitted securely over the internet using encryption protocols such as SSL/TLS to protect it from unauthorized access.

## Vulnerability Prevention

The application should implement measures to prevent common security vulnerabilities, such as SQL injection and cross-site scripting (XSS).

## Access Control

The system should have mechanisms in place to prevent unauthorized access to sensitive data and features.

## Software Quality Attributes

## Usability

The application should have an intuitive and user-friendly console interface to enhance user experience and engagement.

## Reliability

The system should function consistently and accurately to maintain user trust.

1

## Maintainability

The application should be designed in a modular and organized manner for ease of maintenance and updates.

## Testability

The system should be designed to facilitate testing and debugging of features and functionalities.

## Portability

The application should be compatible with different web browsers and devices.

## Adaptability

The system should be adaptable to changing business requirements and evolving technologies.

# Other Requirements

## Legal and Compliance Requirements

## Data Protection and Privacy

The application should comply with relevant data protection and privacy regulations to safeguard user information.

## Intellectual Property Rights

The application should not infringe on any intellectual property rights, and proper licenses should be obtained for any third-party components used.

## User Agreement

The application should have a clear and accessible user agreement outlining terms and conditions for users.

## Documentation Requirements

## User Manual

The application should have a comprehensive user manual explaining how to use the application, its features, and functionalities.

## Technical Documentation

The development team should maintain technical documentation describing the system architecture, database schema, and code structure.

## Development and Testing Requirements

## Development Environment

The development team should use .NET C# and MS SQL Server for the development of the application.

## Testing Environment

A dedicated testing environment should be set up to conduct thorough testing of the application.

## Test Cases

The development team should create test cases to verify the functionality and performance of the application.

## Deployment Requirements

## Server Configuration

The server hosting the application should be properly configured to support the application's software and hardware requirements.

## Database Setup

The MSSQL database should be set up and configured to accommodate the application's data storage needs.

## Release Management

The application should have a defined release management process for deploying updates and new features.

# Appendix A: Glossary

Here is a glossary of key terms and abbreviations used throughout this document:

.NET C#: A programming framework developed by Microsoft for building Windows applications and web services.

MSSQL: Microsoft SQL Server, a relational database management system developed by Microsoft.

GUI: Graphical User Interface, a visual way of interacting with software using graphical elements.

HTTP: Hypertext Transfer Protocol, the foundation of data communication on the World Wide Web.

HTTPS: Hypertext Transfer Protocol Secure, an encrypted version of HTTP for secure data transmission over the internet.

TCP/IP: Transmission Control Protocol/Internet Protocol, the set of communications protocols used for internet communication.

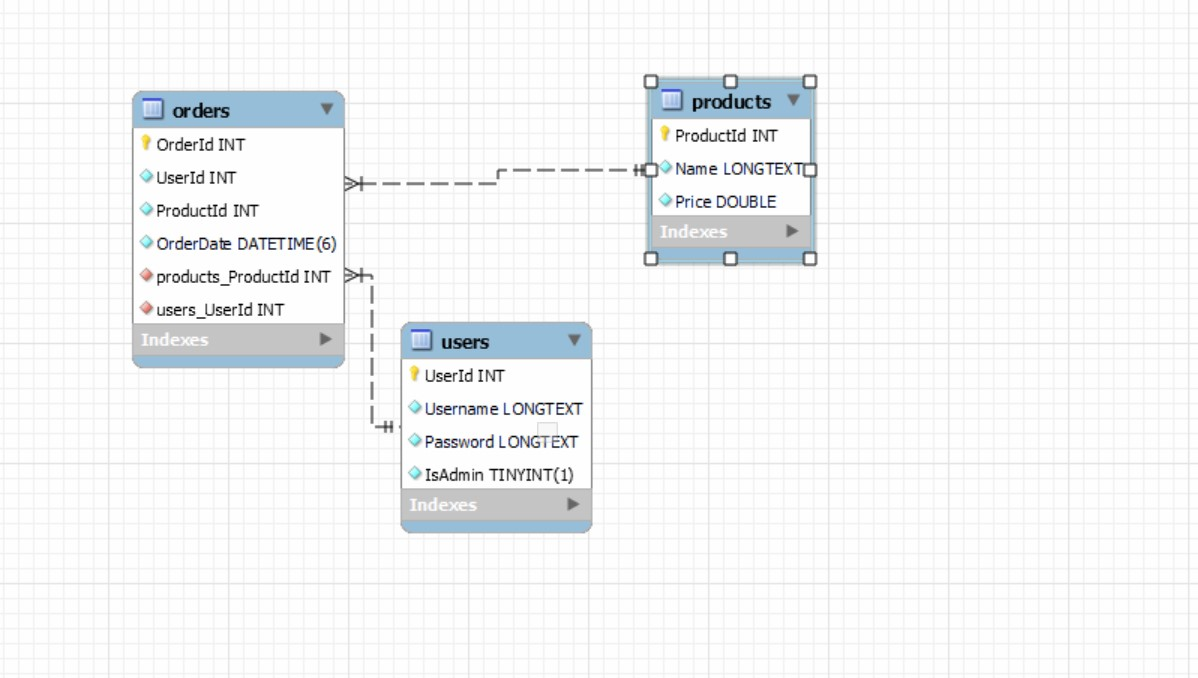
SQL Injection: A type of cybersecurity attack where malicious SQL code is inserted into an application's input fields to manipulate the database.

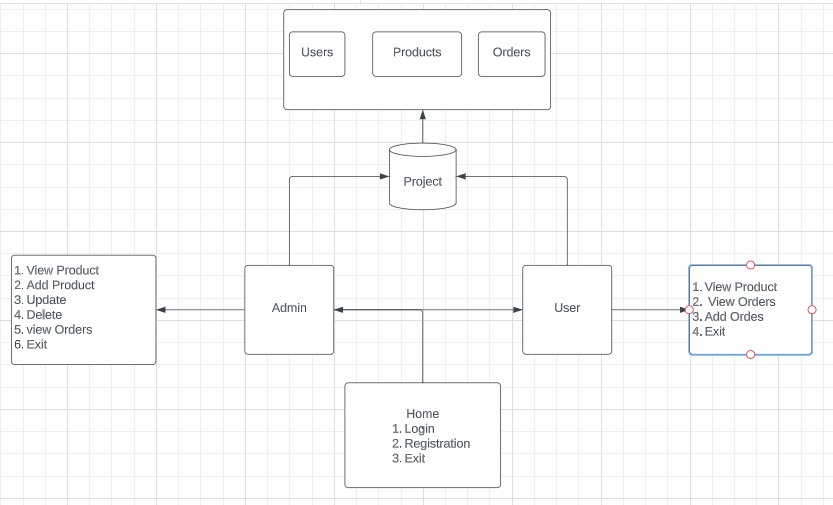
XSS: Cross-Site Scripting, a security vulnerability where attackers inject malicious scripts into web pages viewed by other users.

Its just a simple Console app where i have Register and Login Functionality, if i login as user i can edit my details, see All available products in market, add or remove a product can be done by admin.

# Appendix B: Analysis Model ER Diagram

# 





## Outputs:

