Gourmet Shop Customer Interface

Overview

We will be creating a mock website using WinForms to simulate customer buying experience. With group members Sarah Harlan, Rayleen Roberts, and Broc Keeton we will be dividing work equally to complete the full Software Development Life Cycle.

Project Objectives

Develop a Customer WinForms Application

- Implement **user authentication (login/logout)**

- Display **available products** dynamically

- Enable **filtering by supplier**

- Allow **multi-item purchases**

Update the DataAccess Library

- Implement **database queries** to support product filtering and order processing

- Ensure **secure data transactions** for purchasing

Enhance GourmetShop for Managerial Insights

- Implement **sales tracking features**

- Enable **filtering by customer or product**

Comprehensive Documentation

- Clearly define project plan, task assignments, and development workflow

- Provide inline code comments for maintainability

Diagrams

Login Diagram

Welcome

| Username |
| Password |

Login

Main Dashboard Diagram

Search by Supplier ID: Search... 🔍

Search

Data Grid View
of
Product List

View Cart   Add to Cart
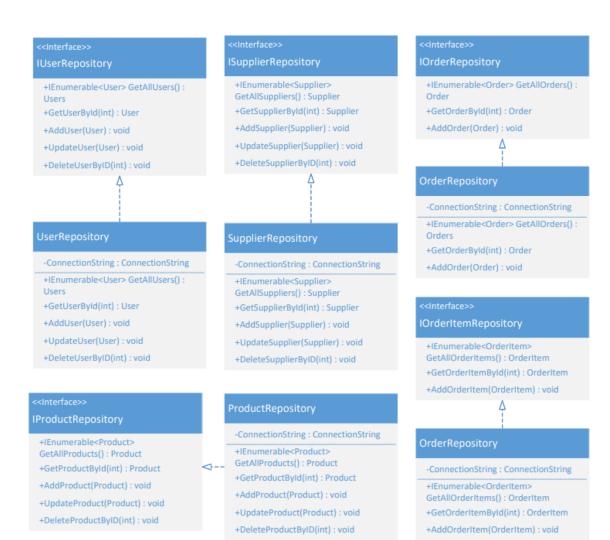
Cart View Diagram

Subtotal:

Tax:

_____

Total Amount Due:

DataGridView to
Display Items in Cart

Remove Last

Back to Products

Buy

UML Class Diagram

<<Interface>>
IUserRepository

+IEnumerable<User> GetAllUsers() : Users
+GetUserById(int) : User
+AddUser(User) : void
+UpdateUser(User) : void
+DeleteUserByID(int) : void

<<Interface>>
ISupplierRepository

+IEnumerable<Supplier> GetAllSuppliers() : Supplier
+GetSupplierById(int) : Supplier
+AddSupplier(Supplier) : void
+UpdateSupplier(Supplier) : void
+DeleteSupplierByID(int) : void

<<Interface>>
IOrderRepository

+IEnumerable<Order> GetAllOrders() : Order
+GetOrderById(int) : Order
+AddOrder(Order) : void

UserRepository

-ConnectionString : ConnectionString

+IEnumerable<User> GetAllUsers() : Users
+GetUserById(int) : User
+AddUser(User) : void
+UpdateUser(User) : void
+DeleteUserByID(int) : void

SupplierRepository

-ConnectionString : ConnectionString

+IEnumerable<Supplier> GetAllSuppliers() : Supplier
+GetSupplierById(int) : Supplier
+AddSupplier(Supplier) : void
+UpdateSupplier(Supplier) : void
+DeleteSupplierByID(int) : void

OrderRepository

-ConnectionString : ConnectionString

+IEnumerable<Order> GetAllOrders() : Orders
+GetOrderById(int) : Order
+AddOrder(Order) : void

<<Interface>>
IOrderItemRepository

+IEnumerable<OrderItem> GetAllOrderItems() : OrderItem
+GetOrderItemById(int) : OrderItem
+AddOrderItem(OrderItem) : void

<<Interface>>
IProductRepository

+IEnumerable<Product> GetAllProducts() : Product
+GetProductById(int) : Product
+AddProduct(Product) : void
+UpdateProduct(Product) : void
+DeleteProductByID(int) : void

ProductRepository

-ConnectionString : ConnectionString

+IEnumerable<Product> GetAllProducts() : Product
+GetProductById(int) : Product
+AddProduct(Product) : void
+UpdateProduct(Product) : void
+DeleteProductByID(int) : void

OrderRepository

-ConnectionString : ConnectionString

+IEnumerable<OrderItem> GetAllOrderItems() : OrderItem
+GetOrderItemById(int) : OrderItem
+AddOrderItem(OrderItem) : void

Database Diagram

Order
- 🔑 Id
- OrderDate
- OrderNumber
- UserId
- TotalAmount

Users
- 🔑 UserId
- FirstName
- LastName
- City
- Country
- Phone
- UserLogin
- UserPassword

OrderItem
- 🔑 Id
- OrderId
- ProductId
- UnitPrice
- Quantity

Product
- 🔑 Id
- ProductName
- SupplierId
- UnitPrice
- Package
- IsDiscontinued

Users_Roles
- 🔑 UserRoleID
- UserID
- RoleID

Supplier
- 🔑 Id
- CompanyName
- ContactName
- ContactTitle
- City
- Country
- Phone
- Fax

Roles
- 🔑 RoleID
- RoleDescription