Project Proposal: Food Delivery Web Application

1. Introduction

This document proposes the development of a data-driven web application for managing a Packaged Food Delivery System. This application will provide distinct interfaces for administrators, customers, and suppliers. The core goal is to create an efficient, secure, and user-friendly platform that streamlines the business's operations from order placement to delivery tracking, adhering to web development best practices.

2. Problem Statement

Managing a packaged food delivery service involves complex coordination between customer orders, multi-warehouse inventory, supplier interactions, and delivery logistics. The lack of a unified, digital system can lead to operational inefficiencies, including inaccurate stock data, delays in order processing, inconsistent customer service, and difficulty in monitoring overall business performance. This project seeks to mitigate these issues through a dedicated web application. **3. Proposed Solution**

A data-driven web application, centered around the existing database will serve as an excellent solution. Key features and technical considerations include:

- **Data-Driven Architecture:** The application's foundation will be its interaction with the database outlined in the Database Report. All functionalities, from displaying product listings to managing inventory and processing orders, will be driven by the data in the database.
- **Role-Based Access Control:** The system will implement strict users and role-based software functionalities. Users will register and log in as one of three roles:
    - **Administrator:** Full access to all system data (customers, suppliers, products, orders, inventory, warehouses, deliveries) and user management capabilities.
    - **Customer:** Ability to register, log in, browse products, manage a shopping cart, place orders, view order history, etc.
    - **Supplier:** Ability to register, log in, view products they supply, check relevant inventory levels, and add new items to supply.

- **State Management:** To ensure a smooth and responsive user experience, the web application will incorporate robust state management techniques.
- **Mandatory Validation:** Validation is mandatory for all data inputs. This includes clientside validation for immediate user feedback and comprehensive server-side validation to ensure data integrity, security, and adherence to the database constraints (e.g., data types, formats, required fields).
- **Data Security: Data security is essential**. The application will incorporate measures such as secure user authentication (SQL injection protection), authorization checks for all actions based on user roles, protection against common web vulnerabilities (e.g.,

  SQL injection, XSS) through input sanitization and prepared statements, and secure session management. HTTPS will be used for data transmission.

Use Case Model Actors:

1. **Customer**

Use Case Descriptions:

Register:

Allows a user to register as a customer.

Login:

Allows a registered customer to log in to the web application.

View Items:

Allows the customer to view items that can be ordered.

Manage Shopping Cart:

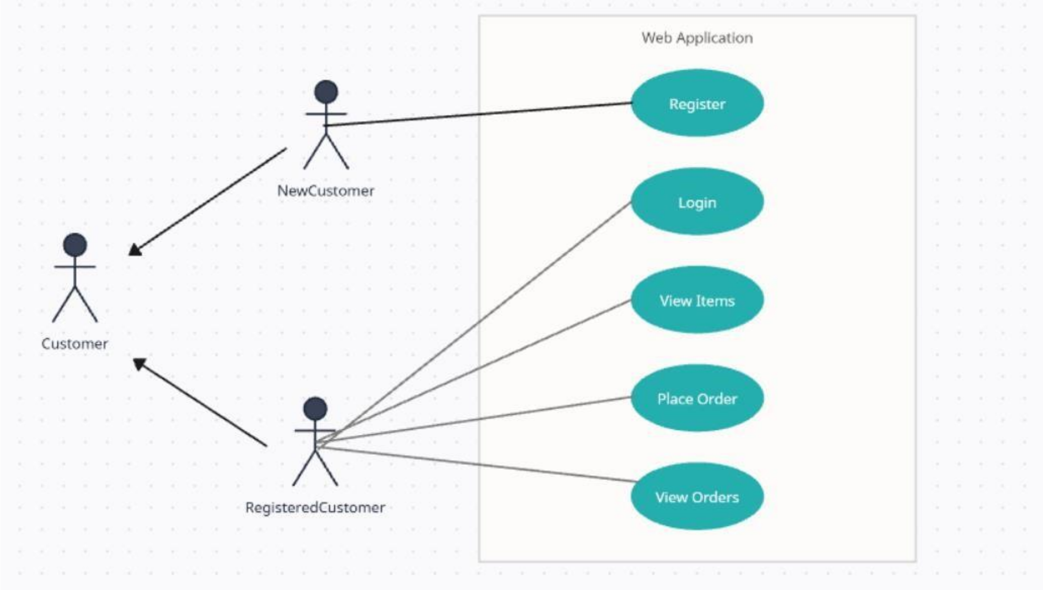
Add/remove items from shopping cart.

Place Order:

Allows the customer to place an order.

View Orders:

Allows the customer to view all their orders.



2. **Supplier**

Use Case Descriptions:

Register:

Allows a user to register as a supplier.

Login:

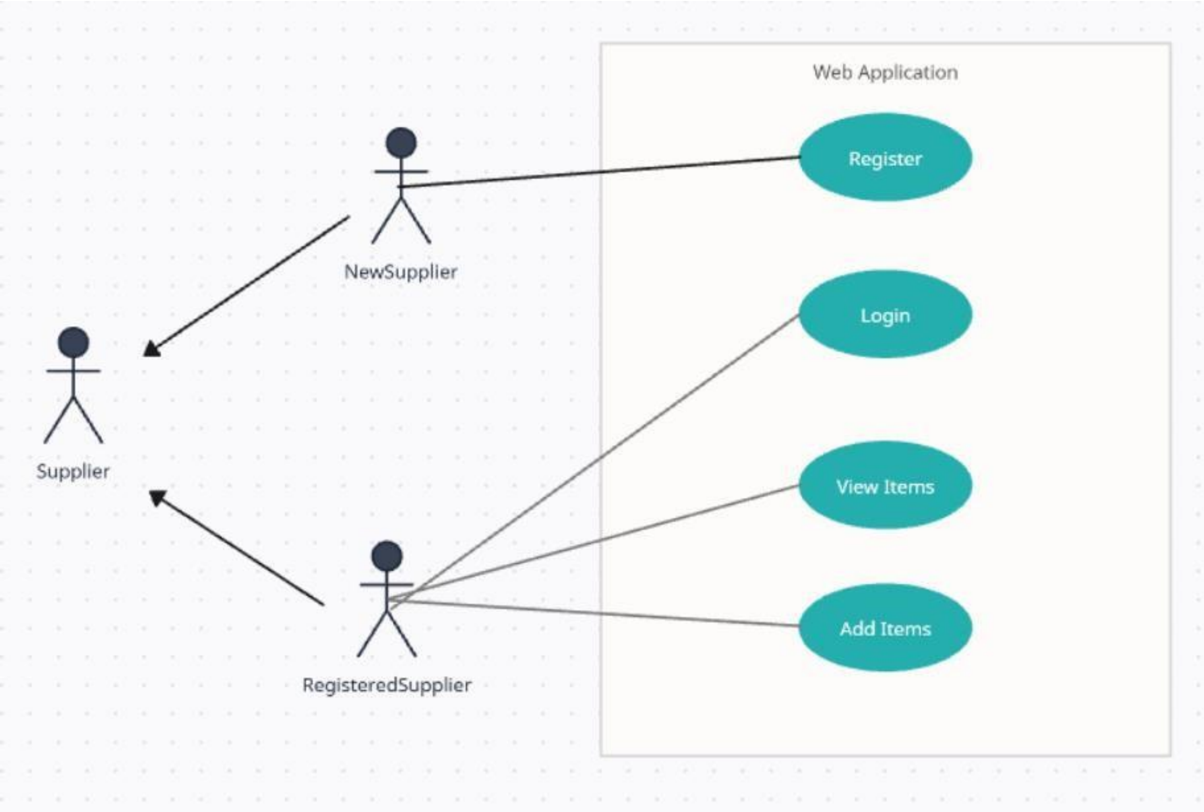
Allows a registered supplier to log in to the web application.

View Items:

Allows the supplier to see the items they supplied.

Add Items:

Allows the supplier to add new items.



3. **Administrator**

Use Case Descriptions:

Login:

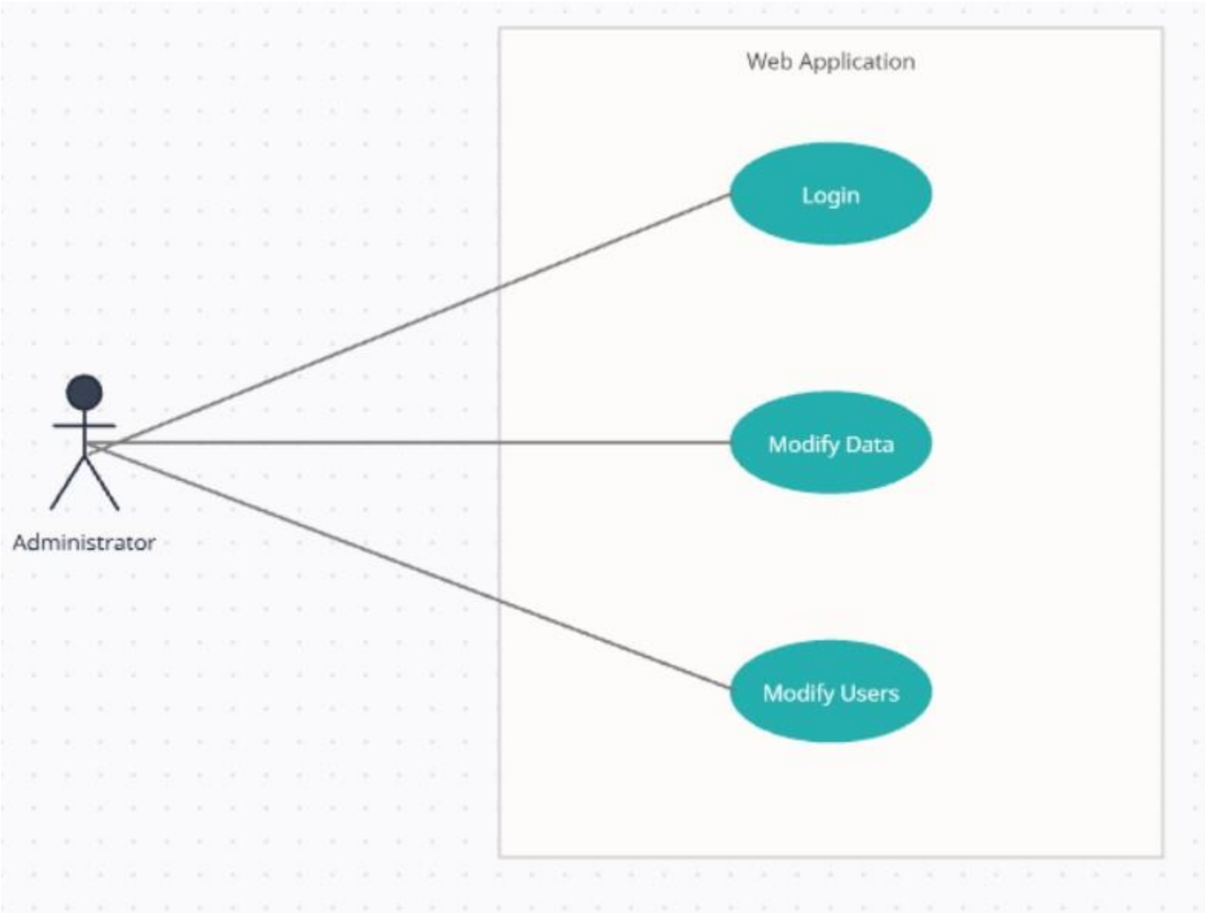
Allows a user to login as an administrator.

Modify Data:

Allows the administrator to modify all the data in the database.

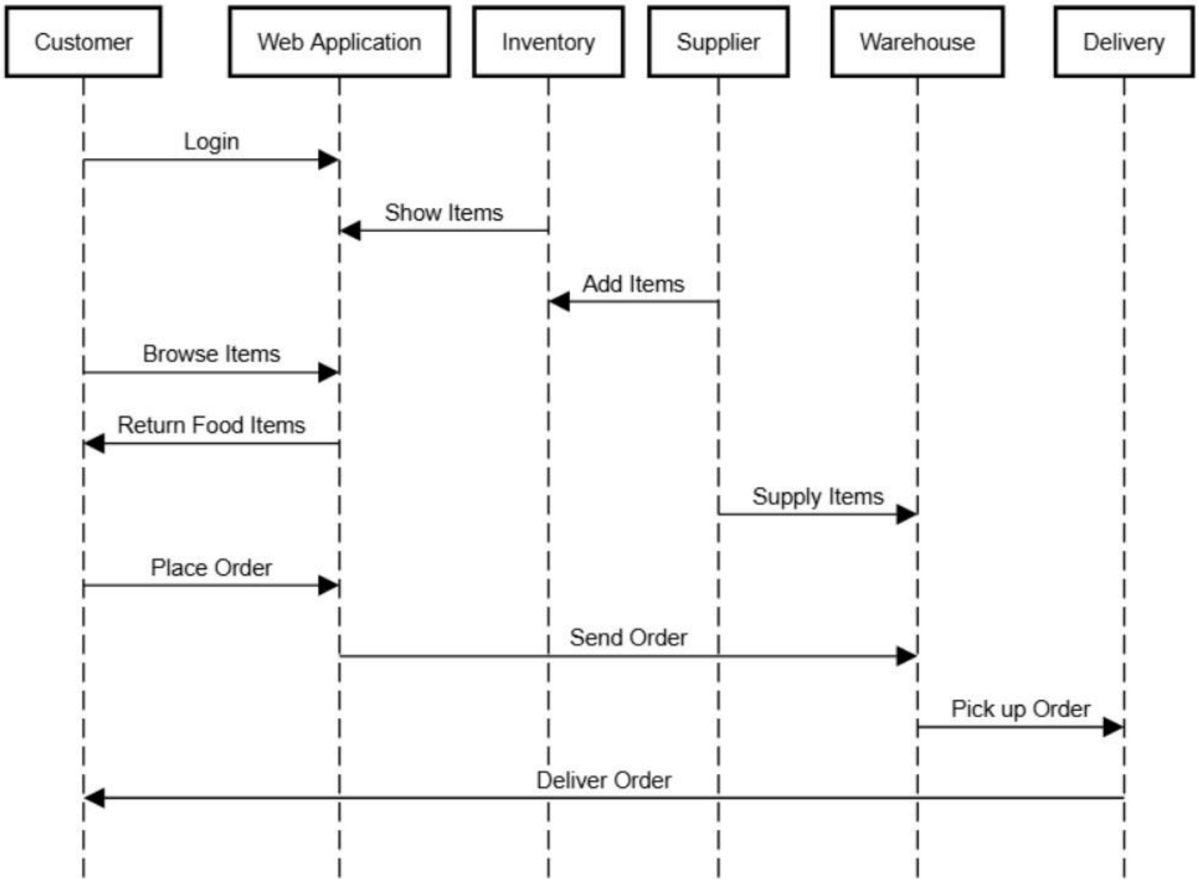
Modify User Permissions:

Allows the administrator to control what a user can and cannot see.

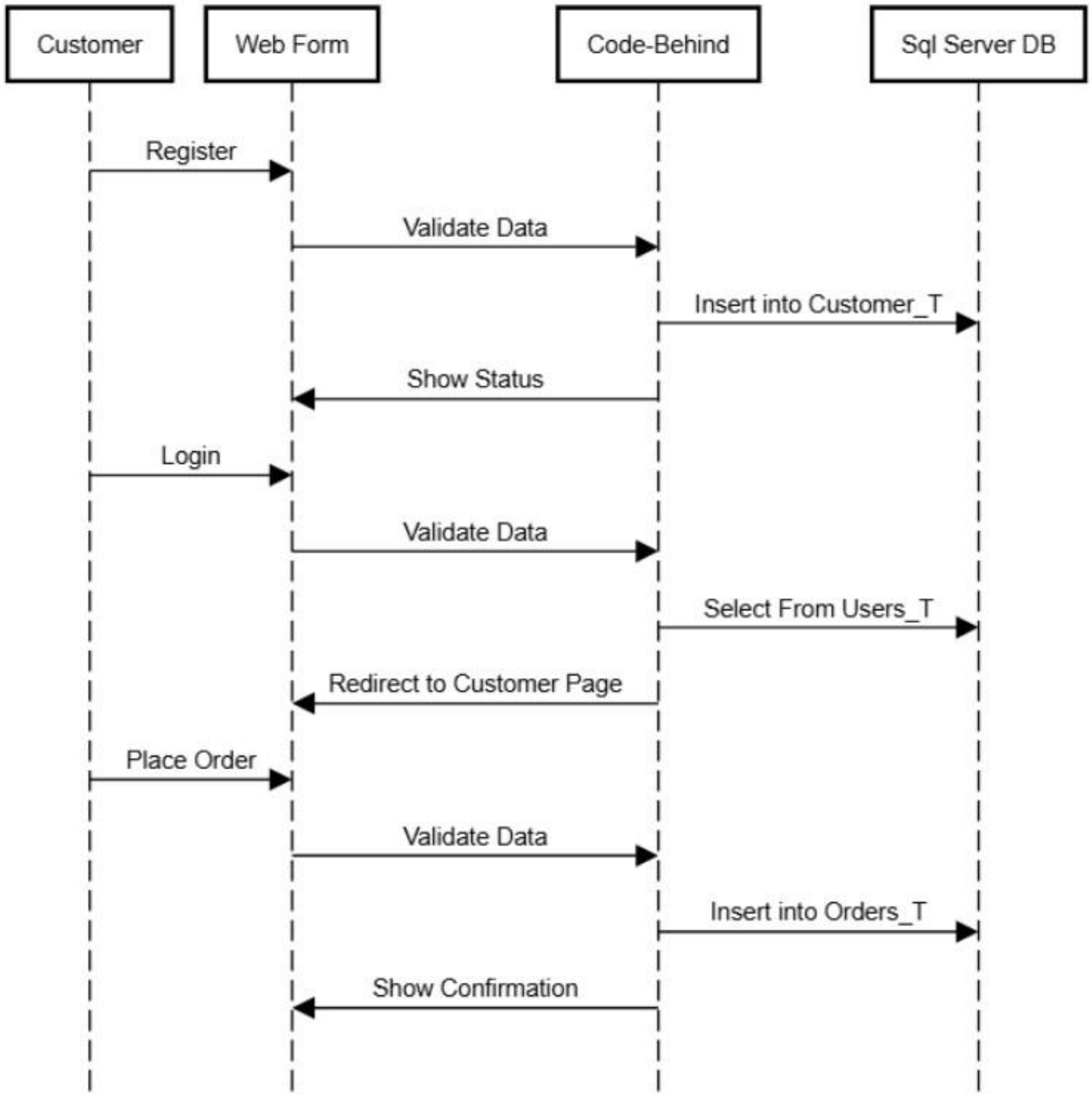


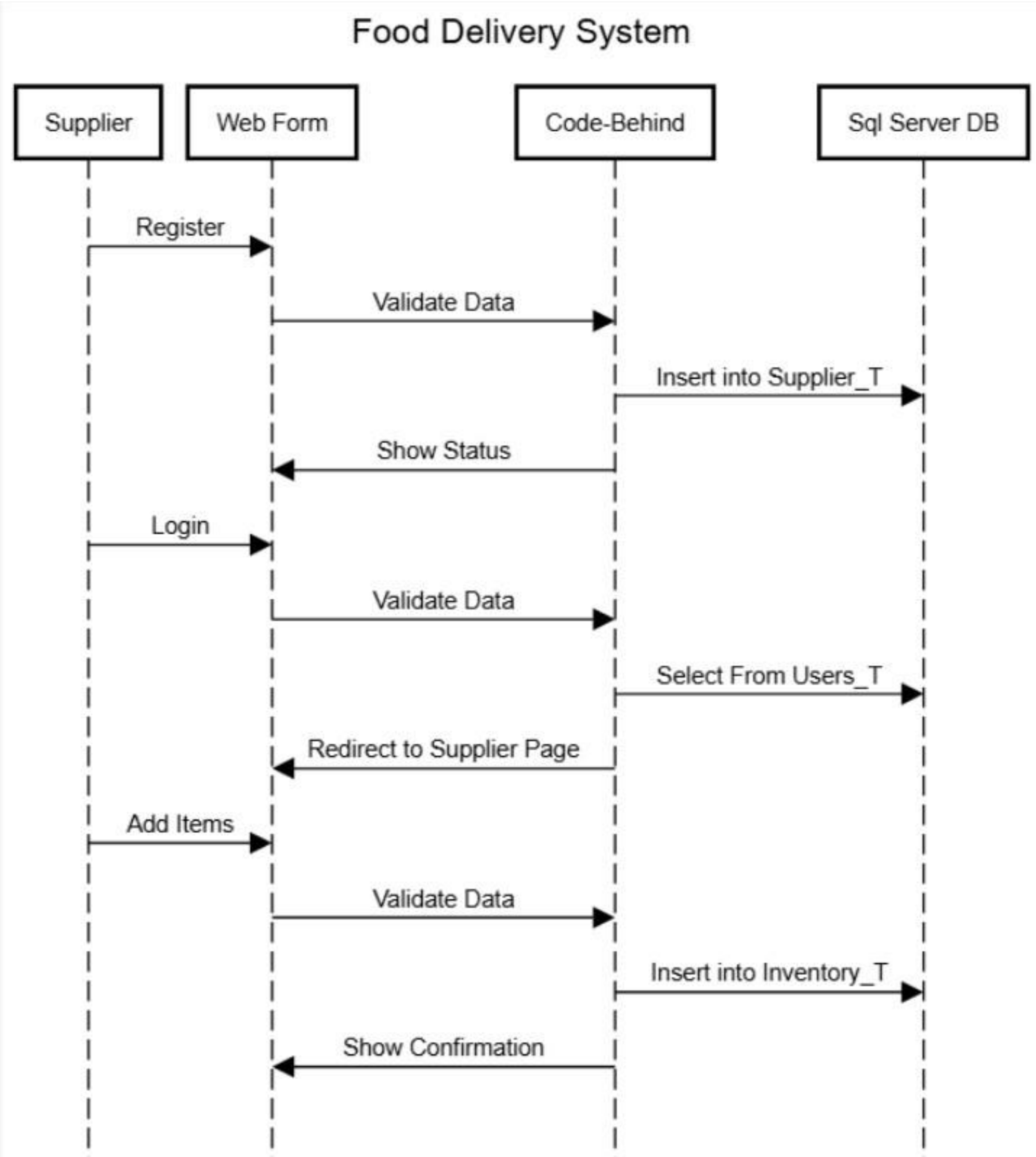
Sequence Diagram

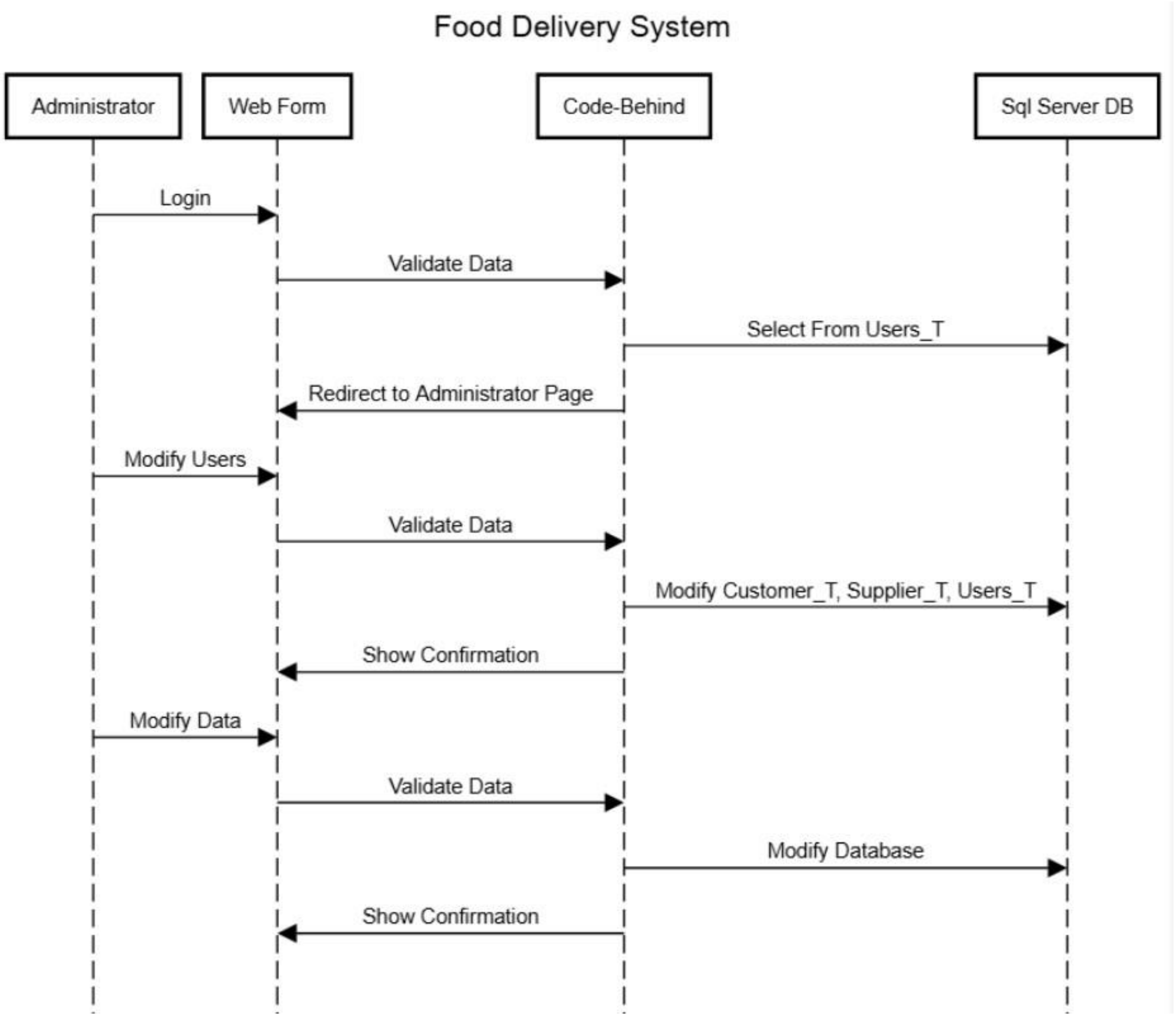
Food Delivery System



Food Delivery System




Food Delivery System
Supplier
Web Form
Code-Behind
Sql Server DB
Register
Validate Data
Insert into Supplier_T
Show Status
Login
Validate Data
Select From Users_T
Redirect to Supplier Page
Add Items
Validate Data
Insert into Inventory_T
Show Confirmation


Food Delivery System
Administrator
Web Form
Code-Behind
Sql Server DB
Login
Validate Data
Select From Users_T
Redirect to Administrator Page
Modify Users
Validate Data
Modify Customer_T, Supplier_T, Users_T
Show Confirmation
Modify Data
Validate Data
Modify Database
Show Confirmation

Software Requirements Specification

Food Delivery System Web Application

1. Introduction

1.1 Purpose

The purpose of this document is to define the functional and non-functional requirements for the Food Delivery System Web Application. This system enables customers to browse food items, add them to a shopping cart, manage their cart, and place orders, while administrators can manage products, categories, and order statuses.

1.2 Intended Audience

- Developers
- Project Managers
- QA/Testers
- End Users (Customers & Admins)

1.3 Scope

This system is a web-based food ordering application developed using **ASP.NET WebForms** and **VB.NET**. It supports customer registration and login, product browsing, cart management with quantity limits based on inventory, and order placement. Admins can manage users.

2. Overall Description

2.1 Product Perspective

This system is a standalone ASP.NET application following a multi-tier architecture It stores customer, product, and order data in a SQL Server database.

2.2 Product Features

- Customer registration, login, and session management
- Browse food items
- Shopping cart with inventory-aware quantity controls
- Order placement
- Admin dashboard

2.3 User Classes and Characteristics

- **Customer:** End-users who can browse food, add items to the cart, and place orders
- **Admin:** Manages food items, inventory, categories, and orders
- **Supplier:** Manages products

2.4 Operating Environment

- ASP.NET WebForms (.NET Framework)
- VB.NET backend
- SQL Server database
- Compatible with modern web browsers (Chrome, Edge, Firefox)

3. Functional Requirements

3.1 Customer Functionalities

- **FR1:** Register with email, password, name, and contact details.
- **FR2:** Log in using registered credentials. Session is stored in Session("Customer_ID").
- **FR3:** View list of food items by category.
- **FR4:** Add food items to cart. If item exists, increase quantity.
- **FR5:** Prevent quantity from exceeding available stock.

- **FR6:** Update or remove items in cart. If quantity = 0, remove item.
- **FR7:** Place an order with items in cart.
- **FR8:** View order history.

3.2 Admin Functionalities

- **FR9:** Log in using admin credentials.
- **FR10:** Add, update, or delete Users.

4. Non-Functional Requirements

4.1 Performance Requirements

- Response time for major operations (e.g., placing order) should be < 2 seconds.
- System should handle up to 100 concurrent users.

4.2 Security Requirements

- Passwords must be hashed.
- Role-based access control (Customer vs Admin).
- SQL injection prevention using parameterized queries.

4.3 Usability

- Simple, intuitive UI with Material Design or Bootstrap.
- Consistent navigation for both customers and admins.

4.4 Reliability & Availability

- System should be available 99% of the time.
- Graceful error handling with user-friendly messages.

5. Database Requirements

- **Customer Table:** ID, Name, Email, Password, Contact
- **Product Table:** ID, Name, CategoryID, Price, QuantityAvailable, ImagePath
- **Category Table:** ID, Name
- **Cart Table:** CartID, CustomerID, ProductID, Quantity
- **Order Table:** OrderID, CustomerID, OrderDate, TotalAmount, Status
- **OrderDetails Table:** OrderID, ProductID, Quantity, Price

6. External Interfaces

6.1 User Interface

- ASP.NET WebForms pages (Default.aspx, Login.aspx, Cart.aspx, AdminPanel.aspx, etc.)
- Client-side scripting with JavaScript/jQuery

6.2 Hardware Interface

- Standard desktop or mobile web browsers
- Hosted on local IIS or cloud VM (for deployment)

6.3 Software Interface

- SQL Server for database
- ASP.NET Membership or custom authentication module