Mess Management System Project Report

MUHAMMAD ALI NASIR

May 21, 2025

Contents

1 Summary

The Mess Management System is a web-based application designed to streamline the operations of a mess facility, catering to both administrative and customer needs. Built using ASP.NET Web Forms with a VB.NET backend and SQL Server database, the system employs a multi-tier architecture with role-based access control to ensure secure and efficient management. The MMS facilitates user management, product and menu scheduling, order processing, billing, payment tracking, and customer credit management. Key features include automated bill generation, real-time menu display, and a robust credit-based payment system, all integrated with a secure, user-friendly interface. This report provides a comprehensive overview of the system's requirements, design, implementation, testing, and potential enhancements.

2 System Requirements Analysis

The MMS was developed to address the operational challenges of managing a mess facility, including menu planning, order processing, billing, and user management. The requirements were gathered through stakeholder interviews and operational analysis.

2.1 Functional Requirements

- **User Management**: Admin can add, view, and delete users; customers can self-register.
- **Product Management**: Manage product lines and products with price history tracking.
- **Menu Management**: Schedule and display daily menus with meal details.
- **Order Management**: Process customer orders and integrate with billing.
- **Billing and Payments**: Generate monthly bills, track payment status, and manage customer credits.
- **Quiz System**: Support quizzes for user engagement (optional feature).

2.2 Non-Functional Requirements

- **Security**: Role-based access control, input validation, and SQL injection prevention.
- **Performance**: Handle multiple concurrent users with real-time data updates.
- **Usability**: Intuitive dashboards and navigation for both admin and customers.
- **Scalability**: Support additional users and features without performance degradation.

3 Database Design Documentation

The MMS uses a SQL Server database with Entity Framework Code First approach to ensure data integrity and scalability. The schema includes core tables for user management, product catalog, menu scheduling, orders, billing, and auxiliary features like quizzes and price history.

3.1 Database Schema

Below is a simplified representation of the core tables and their relationships:

```
-- User Management
CREATE TABLE REGISTERUSER_T (
    UserID INT PRIMARY KEY IDENTITY,
    Username NVARCHAR(50) UNIQUE,
    Password NVARCHAR(100),
    RoleID INT,
    FOREIGN KEY (RoleID) REFERENCES ROLE_T(RoleID)
);

CREATE TABLE ROLE_T (
    RoleID INT PRIMARY KEY IDENTITY,
    RoleName NVARCHAR(50)
);

-- Product Management
CREATE TABLE PRODUCTLINE_T (
    ProductLineID INT PRIMARY KEY IDENTITY,
    LineName NVARCHAR(100),
    Category NVARCHAR(50),
    Quantity INT
);

CREATE TABLE PRODUCT_T (
    ProductID INT PRIMARY KEY IDENTITY,
    ProductName NVARCHAR(100),
    ProductLineID INT,
    Price DECIMAL(10,2),
    FOREIGN KEY (ProductLineID) REFERENCES PRODUCTLINE_T(
       ProductLineID)
);

-- Menu Scheduling
CREATE TABLE AVAILABLEMEAL_T (
    MealID INT PRIMARY KEY IDENTITY,
    ProductID INT,
    FOREIGN KEY (ProductID) REFERENCES PRODUCT_T(ProductID)
);

CREATE TABLE MEALSCHEDULE_T (
    ScheduleID INT PRIMARY KEY IDENTITY,
    MealID INT,
```

```
    MealDate DATE,
    MealType NVARCHAR(50),
    FOREIGN KEY (MealID) REFERENCES AVAILABLEMEAL_T(MealID)
);

-- Order Management
CREATE TABLE ORDERS_T (
    OrderID INT PRIMARY KEY IDENTITY,
    UserID INT,
    OrderDate DATE,
    FOREIGN KEY (UserID) REFERENCES REGISTERUSER_T(UserID)
);

CREATE TABLE ORDERLINE_T (
    OrderLineID INT PRIMARY KEY IDENTITY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    FOREIGN KEY (OrderID) REFERENCES ORDERS_T(OrderID),
    FOREIGN KEY (ProductID) REFERENCES PRODUCT_T(ProductID)
);

-- Billing and Payments
CREATE TABLE BILL_T (
    BillID INT PRIMARY KEY IDENTITY,
    UserID INT,
    TotalAmount DECIMAL(10,2),
    Status NVARCHAR(20),
    Month NVARCHAR(20),
    FOREIGN KEY (UserID) REFERENCES REGISTERUSER_T(UserID)
);

CREATE TABLE PAYMENT_T (
    PaymentID INT PRIMARY KEY IDENTITY,
    BillID INT,
    Amount DECIMAL(10,2),
    PaymentDate DATE,
    FOREIGN KEY (BillID) REFERENCES BILL_T(BillID)
);

CREATE TABLE USER_CREDITS (
    CreditID INT PRIMARY KEY IDENTITY,
    UserID INT,
    CreditAmount DECIMAL(10,2),
    FOREIGN KEY (UserID) REFERENCES REGISTERUSER_T(UserID)
);

CREATE TABLE DEFAULTERLIST_T (
    DefaulterID INT PRIMARY KEY IDENTITY,
    UserID INT,
    AmountDue DECIMAL(10,2),
```

```
    FOREIGN KEY (UserID) REFERENCES REGISTERUSER_T(UserID)
);

-- Price History
CREATE TABLE PRICEHISTORY_T (
    PriceHistoryID INT PRIMARY KEY IDENTITY,
    ProductID INT,
    OldPrice DECIMAL(10,2),
    NewPrice DECIMAL(10,2),
    ChangeDate DATE,
    FOREIGN KEY (ProductID) REFERENCES PRODUCT_T(ProductID)
);
```

3.2 Key Relationships

- REGISTERUSER_T links to ROLE_T via RoleID for role-based access.
- PRODUCT_T is associated with PRODUCTLINE_T through ProductLineID.
- AVAILABLEMEAL_T references PRODUCT_T for menu items.
- MEALSCHEDULE_T links to AVAILABLEMEAL_T for scheduling.
- ORDERS_T and ORDERLINE_T manage order details with references to REGISTERUSER_T and PRODUCT_T.
- BILL_T and PAYMENT_T track billing and payments, linked to REGISTERUSER_T.
- USER_CREDITS and DEFAULTERLIST_T manage customer credits and defaulters.

3.3 ER Diagram Description

An Entity-Relationship (ER) diagram would depict entities as rectangles (e.g., REGISTERUSER_T, PRODUCT_T), relationships as diamonds, and attributes as ovals. Foreign key relationships (e.g., UserID, ProductID) are shown as lines connecting entities. For example, REGISTERUSER_T connects to ROLE_T via RoleID, and ORDERS_T connects to ORDERLINE_T and REGISTERUSER_T.

4 System Architecture Explanation

The MMS employs a multi-tier architecture to ensure modularity, scalability, and maintainability:

- **Presentation Layer**: ASP.NET Web Forms with HTML/CSS and server controls for dynamic UI.
- **Business Logic Layer**: VB.NET handles core functionalities like user management, order processing, and billing.
- **Data Access Layer**: Entity Framework Code First manages database operations with SQL Server.

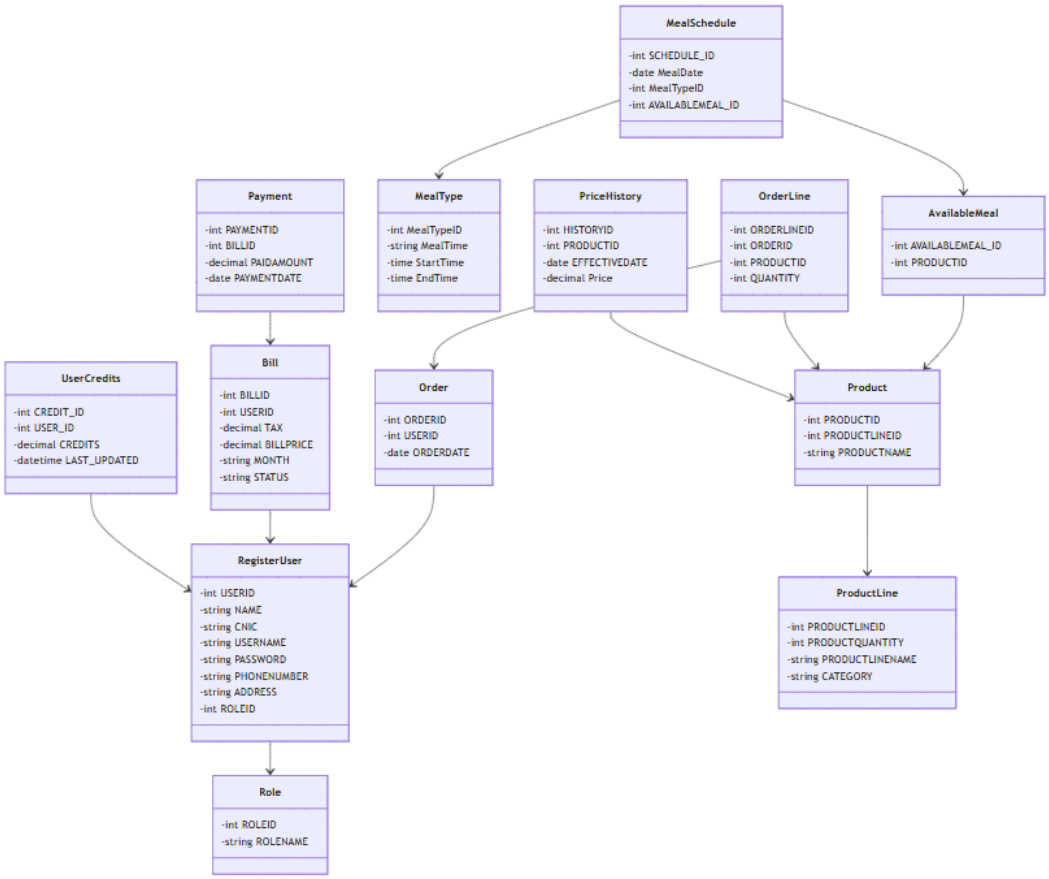


Figure 1: Class Diagram of the Mess Management

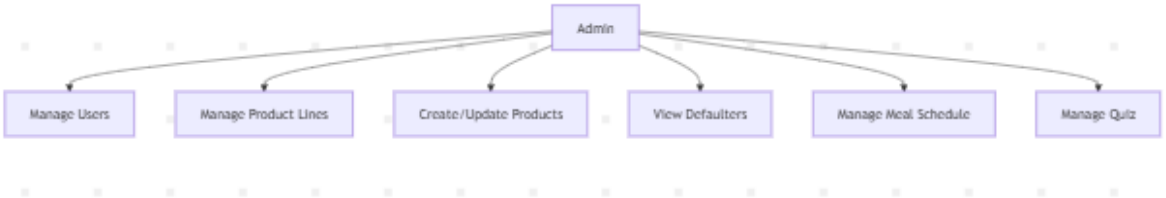


Figure 2: Usecase Diagram of the Mess Management

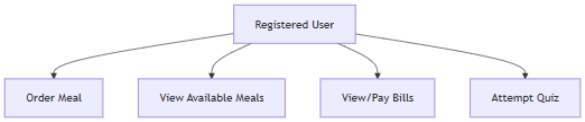


Figure 3: USECASE DIAGRAM OF CUSTOMER

- **Database Layer**: SQL Server stores data with triggers and constraints for integrity.

4.1 System Flow Diagram Description

A system flow diagram would illustrate the user interaction flow:

- **Customer**: Logs in → Views menu → Places order → Views/pays bills → Checks credits.
- **Admin**: Logs in → Manages users/products/menu → Generates bills → Tracks payments.

The diagram would use arrows to show navigation between pages (e.g., Login.aspx to CustomerDashboard.aspx or AdminDashboard.aspx).

5 Module-wise Functionality Description

5.1 Admin Module

- **User Management**:
  - Add new users with role assignment.
  - View user list with details (UserID, Username, Role).
  - Delete users by UserID.
- **Product Management**:
  - Manage product lines (add, view, delete).
  - Manage products with price history tracking.
- **Menu Management**:
  - Schedule meals with date, type, and products.
  - Display scheduled menus.
- **Billing System**:
  - Generate monthly bills for users.
  - View bills with status (Paid/Unpaid/Overpaid).
- **Order Management**:
  - Process customer orders and integrate with billing.

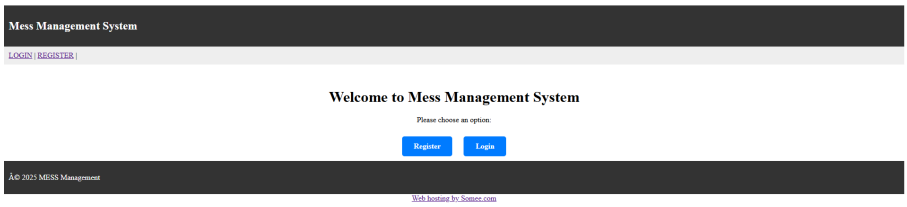


Figure 4: home page

5.2 Customer Module

- **Credit Management**:
  - View real-time credit balance.
  - Deduct credits for payments.
- **Menu Viewing**:
  - Display daily menus with meal details and prices.
- **Billing and Payments**:
  - View personal bill history.
  - Pay bills using credit balance.
- **Order Placement**:
  - Place orders from available menu items.

6 User Interface Design

The UI is built using ASP.NET Web Forms with a master page for consistent navigation. Key features:

- **Admin Dashboard**: Centralized interface for managing users, products, menus, and bills.
- **Customer Dashboard**: Displays menu, credits, bills, and order placement options.
- **Responsive Design**: CSS ensures compatibility across devices.
- **Navigation**: Menu bar with role-based access to modules.

6.1 UI Mockup Description

A UI mockup would show:

- **Login Page**: Username/password fields with role-based redirection.

Mess Management System

LOGIN | REGISTER |

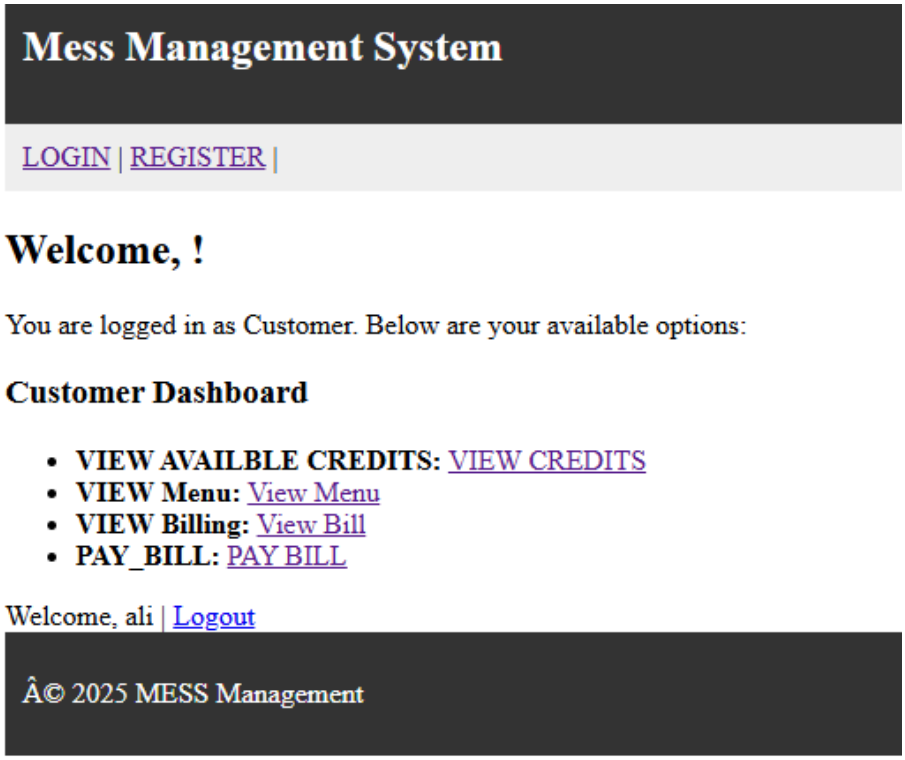
Welcome, admin!

You are logged in as Administrator. Below are your management options:

Admin Dashboard

- **Manage Users:** Manage users
- **Manage Products:** Manage Products
- **Manage Menu:** Manage Menu
- **Billing:** Manage Bills
- **Orders:** Manage Orders
- **Reports:**

Welcome, admin | Logout



- **Admin Dashboard**: Grid views for users, products, and bills; forms for adding/scheduling.
- **Customer Dashboard**: Menu table, credit balance, and bill payment interface.

7 Testing and Validation

The system underwent rigorous testing to ensure functionality and reliability:

- **Unit Testing**: Tested individual components (e.g., bill generation, order processing) using MSTest.
- **Integration Testing**: Verified module interactions (e.g., order-to-billing integration).
- **User Acceptance Testing**: Conducted with sample users to validate usability.
- **Security Testing**: Checked for SQL injection and unauthorized access vulnerabilities.

8 Security Implementation

- **Authentication**: Username/password with hashed passwords.
- **Authorization**: Role-based access control using session variables.

- **Input Validation**: Server-side validation to prevent injection attacks.
- **Session Management**: Secure session handling with timeouts.

9 Future Enhancements

- **Mobile App Integration**: Develop iOS/Android apps for customer access.
- **Notification System**: Email/SMS alerts for bill due dates and menu updates.
- **Advanced Reporting**: Generate analytical reports for mess operations.
- **Inventory Management**: Enhance stock tracking with low-stock alerts.

10 Conclusion

The Mess Management System successfully addresses the operational needs of a mess facility, providing a robust, secure, and user-friendly platform for managing users, products, menus, orders, and billing. The multi-tier architecture, combined with ASP.NET Web Forms, VB.NET, and SQL Server, ensures scalability and maintainability. Future enhancements will further improve its functionality and user reach. This report documents the system's comprehensive design, implementation, and validation, serving as a complete reference for stakeholders.