**Technical Design Specification Document**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to provide a detailed technical design for a Customer Profile Management Web Application using ASP.NET MVC with Razor. The application will allow users to perform CRUD (Create, Read, Update, Delete) operations on customer profiles and their associated addresses.

The proper design for Customer Creation Form should have multiple addresses form. But there are only two addresses for Home and Office Addresses.

The proper design for Customer Update Form should have multiple addresses form and can update those addresses too. Now there are not multiple addresses in Customer Update Form.

Another way for proper design should have two Web pages for Customer Creation Form and Customer Addresses Creation Form.

There should have (Create, Read, Update, Delete) for two Web pages of Customer Creation Form and Customer Addresses Creation Form.

**1.2 Scope**

This document covers the following aspects:

* System Architecture
* Database Design
* Application Components
* User Interface Design
* Security Considerations
* Testing Plan

**2. System Architecture**

**2.1 Overview**

The system architecture consists of:

* Client Side: Browser-based user interface developed using Razor views and HTML/CSS.
* Server Side: ASP.NET MVC application hosted on a web server.
* Database: SQL Server database to store customer and address information.

**2.2 Technology Stack**

* **Frontend:** HTML, CSS, JavaScript, jQuery, Razor
* **Backend:** ASP.NET MVC, C#
* **Database:** SQL Server
* **Tools:** Visual Studio 2022, Entity Framework 7

**3. Database Design**

**3.1 Database Schema**

**Tables:**

* **Customer**
  + Id (int, Primary Key)
  + Name (nvarchar (255))
  + Photo (nvarchar(100))
  + Email (nvarchar(100))
  + Phone (nvarchar(100))
  + Fax (nvarchar(100))
  + Internet (nvarchar(100))
  + Remarks (nvarchar(255))
* **Address**
  + Id (int, Primary Key)
  + CustomerId (int, Foreign Key)
  + AddressType (nvarchar(100))
  + AddressLine (nvarchar(100))
  + City (nvarchar(100))
  + Country (nvarchar(100))
  + PostalCode (nvarchar(100))

**Relationships:**

* **Customer-Address:** One-to-Many relationship

**4. Application Components**

**4.1 Models**

Define the Customer, CustomerAndAddresses, CustomerAndListAddress and Address models.

**4.2 Controllers**

* **CustomerController:** Handles customer-related actions (Index, Create, Edit, Delete).
* **HomeController:** Handles address-related actions (if needed).

**4.3 Views**

* **Index.cshtml:** Displays a list of customers.
* **Create.cshtml:** Form for creating a new customer. (There should have confirmation message for user create or not. Now there is not included because of time limitation.)
* **Edit.cshtml:** Form for editing an existing customer. (There should have confirmation message for user delete or not. Now there is not included because of time limitation.)
* **Delete.cshtml:** Confirmation view for deleting a customer. (There should have confirmation message for user delete or not. Now there is not included because of time limitation.)

**5. User Interface Design**

**5.1 Layout**

* **Header:** Application name and navigation links.
* **Main Content:** Forms and tables for managing customers and addresses.
* **Footer:** Footer information.

**5.2 Forms**

* **Customer Form:** Fields for Name, Photo, Email, Phone, Fax, Internet, and Remarks.
* **Address Form:** Fields for Address Type, Address Line, City, Country, and Postal Code.

**6. Security Considerations**

* **Authentication:** Implement user authentication using ASP.NET Identity. (not included in the project but it should have)
* **Authorization:** Restrict access to customer management pages based on user roles. (not included in the project but it should have)
* **Validation:** Server-side and client-side validation for form inputs. (not completely included but it should have)
* **Data Encryption:** Encrypt sensitive data, such as passwords and customer details. (not included but it should have)

**7. Testing Plan**

**7.1 Unit Testing**

* **Controllers:** Test CRUD operations.
* **Models:** Test data validation.

**7.2 Integration Testing**

* **Database Integration:** Test database CRUD operations.
* **User Interface:** Test user interactions and form submissions. User Interface should have competitive design but I do not have much time for this. The UI is not proper design now. The proper design is Customer Creation Form should have multiple addresses, Home Address, Office Address and more. Users can choose their desire addresses. Now there is only limit two addresses.

**7.3 User Acceptance Testing (UAT)**

* Conduct UAT with end-users to ensure the application meets business requirements.

**8. Conclusion**

This technical design specification outlines the architecture, database design, application components, user interface design, security considerations, and testing plan for the Customer Profile Management Web Application.