Personal Workout Management System (PWMS-CRM) Application - Software Requirements Specification

**Description**

Personal Workout Management System (PWMS) is the consummate personal fitness application that provides a Personal Trainer with the essential tools he or she needs in order to become a successful personal trainer. PWMS provides personal trainers the ability to add and edit their clientele list and also the ability to create and assign re-usable 5-Step Workout plans.

**Purpose**

The purpose of this document is to provide a guide for developers and testers who are responsible for the development of this CRM platform. It provides the information necessary to design, develop, and test the software.

**Scope**

This document serves as a description of the required improvements to this CRM project. It contains a workflow description, functional and nonfunctional requirements, and appropriate diagrams to fully describe the system.

**System Overview**

This enterprise project Uses the MVC architectural design pattern which consists of three major components and also a Models layer that is used between the three major Components.

The **Presentation Layer (View)** is the Web Portal, which is what the user sees and interacts with. In this case this would be the ASP.NET MVC View application ending in .cshtml.

The **business logic layer(Contoller)** coordinates the application, processes the interaction by the user, and makes calculations and decisions based on the user’s input. The business layer processes and sends data between the Data Access Layer and the Presentation Layer. The controller has the classes that contains the business logic and perform the mentioned calculations and transfer of data between layers.

The **data access layer (Class)** is responsible for sending data from the database to the business logic layer for calculations and vice-versa. The Data Access Layer is represented by a class library that has methods using ADO.NET Framework to connect to the SQL Server Database.

The **Model Layer (Class)**  is used between all of the three major layers to transfer information between the three layers. The Model Layer consists of classes that represent the databases with properties that are columns in your database. and also the TraineeViewModel Consists of classes that get rendered in the View.

This architecture has been chosen to keep separate components in order for scalability. The Presentation Layer(View) should only interact with the Business Layer(Controller), and the Data Layer(DAL) should only interact with the database and the Business Layer(Controller). The Business Layer is responsible for any calculations and processing of data between the Data Layer and the Presentation Layer.

**Use Cases**

* **All Users (Personal Trainers)**
* Workout Plan Management
* Create a 5-Step Workout Plan for a Trainee
* Update a 5-Step Workout Plan for a Trainee
* Delete a 5-Step Workout Plan for a Trainee
* Trainee Management
* Create a Trainee to the database
* Update the data stored for an Trainee
* Delete a Trainee from the database
* Assign a Workout to a Trainee
* View an Assigned Workout per Trainee

**Use Case Diagrams**



**Business Requirements**

**BR1: Ability to Add a Trainee:** After the User provides the required fields, the page should return to the Trainee Listing to view the added Trainee

**BR2: Ability to Edit an Already Existing Trainee:**After the User edits the required fields, the page should return to the Trainee Listing to view changes

**BR3: Ability to Delete an Existing Trainee:**The User should be able to confirm if he/she truly wants to delete the Trainee

**BR4: Ability to Assign a Workout to a Trainee:**Each Trainee can only be assigned 1 Workout at a time, however the User should have the ability to assign/unassign Workouts as they wish

**BR5: Ability to Add a Workout:** After the User provides the required fields, the page should return to the Workout listing to view the added workout; Each Workout should have exactly 5 exercises including repetitions and sets.

**BR6: Ability to Edit an Already Existing Workout:** After the User edits the required fields, the page should return to the Workout Listing to view changes; Each Workout should have exactly 5 exercise including repetitions and sets.

**BR7:** **Ability to Delete an Existing Workout:** The User should be able to confirm if he/she truly wants to delete the Workout

**BR8: Ability to View Assigned Workout per Trainee:** The User should be able to see the Workout Plan assigned to the Trainee, including Name of Workout and Exercises; A Workout can be assigned to multiple Trainees, however a Trainee can only be assigned 1 Workout at a time.

**Functional Requirements**

**FR1: All Fields in the Add Trainee form are required**

**FR2: All Fields in the Edit Trainee form are required**

**FR3: All Fields in the Add Workout form are required**

**FR4: All Fields in the Edit Workout form are required**