

Gym Management System – Comprehensive MVC Project Documentation

NOTE: Unnecessary enterprise-level parts removed. Realistic for MVC Project. Includes optional enhancements (Email, In■App Notifications, Push Notifications, Wearables, etc.)

1. INTRODUCTION

A Gym Management System that allows trainers to assign personalized workout and nutrition plans, members to log daily progress, and both sides to receive notifications and analytics. MVC-based structure with scalable database design and background services for notifications.

2. SYSTEM MODULES OVERVIEW

1. Member Management
2. Trainer Management
3. Workout Management
4. Nutrition Meal Plans
5. Daily Logging (Meals + Workouts)
6. Progress Tracking & Analytics
7. Notifications (Email + In-App + Optional Push)
8. Authentication (Members & Trainers)
9. Optional Enhancements

3. BUSINESS RULES

- Every member is assigned to exactly one trainer.
- Trainer can manage multiple members.

- Trainer designs weekly workouts & meal plans.

- Members must log meals & workouts daily.

- System calculates adherence automatically.

- Notifications:

- Before meal time

- If a member misses logging

- Progress milestone achieved

- Trainer gets weekly status report on members

- Members cannot edit assigned plans.

- Trainers cannot edit member logs.

- Admin may exist optionally for managing users.

4. DATABASE DESIGN (Simplified)

Tables:

- Users (UserID, FullName, Email, PasswordHash, Role [Member/Trainer])

- Members (MemberID, UserID, Age, Height, Weight, TrainerID)

- Trainers (TrainerID, UserID, Specialization)

- Workouts (WorkoutID, TrainerID, Name, Description)

- WorkoutExercises (ExerciseID, WorkoutID, Name, Sets, Reps, RestTime)

- MealPlans (MealPlanID, TrainerID, Name)

- MealPlanItems (ItemID, MealPlanID, MealName, Calories, Protein, Carbs, Fat, Time)

- AssignedPlans (AssignmentID, MemberID, WorkoutID, MealPlanID, StartDate, EndDate)

- WorkoutLogs (LogID, MemberID, ExerciseID, Date, Completed)

- MealLogs (LogID, MemberID, ItemID, Date, Completed)

- Progress (ProgressID, MemberID, Weight, BodyFat, MuscleMass, DateLogged)

- Notifications (NotificationID, UserID, Message, IsRead, CreatedAt, Type)
- =====

5. CORE FEATURES DETAILS

=====

WORKOUT MODULE

- Trainers create workouts
- Members view assigned workouts
- Members log completion
- System tracks adherence rate

NUTRITION MODULE

- Trainers create meal plans
- Members log eaten meals
- Compare planned vs actual intake

PROGRESS TRACKING

- Members log weight or body stats weekly
- Graphs and analytics on dashboard
- Weekly progress summary sent to trainer

NOTIFICATIONS (FULL SYSTEM)

A. Email Notifications

- Before meal time (configurable)
- Weekly trainer report
- Milestone achievement congratulation
- Missing logs reminder

B. In-App Notifications

- Dashboard popups
- Notification bell (like Facebook)
- Trainer → Member or System → User

C. Optional Push Notifications

- Mobile app via Firebase API

6. SYSTEM ARCHITECTURE (MVC)

Controllers:

- MembersController
- TrainersController
- WorkoutsController
- NutritionController
- LogsController
- ProgressController
- NotificationController

Services:

- NotificationService
- EmailService (SMTP)
- AdherenceCalculator
- MealTimeScheduler (Background Service)
- ProgressAnalyzer

Views (Razor):

- Member Dashboard

- Trainer Dashboard

- Workout Views

- Meal Plan Views

- Analytics Charts

- Notification Center

7. BACKGROUND JOBS (REAL-TIME FEATURES)

A. Meal Time Reminder Job

- Runs every 5 minutes

- Checks upcoming meals

- Sends email + in-app notification

B. Daily Logging Check

- At midnight, check missing logs

- Send reminders

C. Weekly Progress Summary

- Trainer receives list of:

- Member progress

- Adherence %

- Missed logs

D. Optional:

- WebSocket/SignalR real-time updates

- Wearable sync job (Apple Health, Fitbit)

8. OPTIONAL ENHANCEMENTS (ADDED)

1. Email Notifications (SMTP)
 2. In-App Notification Center
 3. Push Notifications for Mobile
 4. AI-Based Meal Recommendation
 5. AI Workout Suggestion Engine
 6. Wearable Device Integration
 7. Water Intake Tracking
 8. Macro/Micro Nutrient Analysis
 9. Trainer–Member Chat System
 10. Video Library for Exercises
 11. Automated Weekly Progress PDF Export
-

9. RECOMMENDATIONS FOR REAL-TIME IMPLEMENTATION

- Use SignalR for real-time trainer dashboard updates.
 - Use background services (Quartz.NET / Hangfire) for scheduling.
 - Use caching (Redis) for fast analytics rendering.
 - Keep controllers thin; put heavy logic in services.
 - Use repository-pattern + DTOs for clean separation.
 - For mobile notifications, integrate Firebase Cloud Messaging.
-

10. CONCLUSION

This document is now a full, realistic, production-grade Gym Management System plan suitable for building a full MVC project with scalable architecture and optional real-time features.