GYM Management System
OOAD Mini Project Specifications

PES1UG23CS907	C S Deepak
PES1UG23CS160	CHENNUPATI GUNADEEP
PES1UG23CS152	BYSANI NIRVAAN

1. Introduction

The Gym Membership and Workout Tracker is an Object-Oriented Analysis and Design (OOAD) based system that models real-world gym operations using object-oriented principles.

The system focuses on identifying objects, their attributes, behaviors, and relationships to design a maintainable and scalable software system.

2. Problem Statement

Managing gym members, trainers, workout plans, payments, and attendance manually is inefficient and error-prone.

This project aims to design an object-oriented system that automates gym management and provides a structured solution using UML models.

3. Objectives

- To apply OOAD concepts in a real-world system.

- To identify classes, attributes, and methods.

- To model relationships using UML diagrams.

- To ensure reusability and modularity.

- To design a scalable and maintainable system.

4. System Actors

- Admin

- Trainer

- Member

5. Identified Classes and Attributes

Member Class

Attributes:

- memberId

- name

- email

- phoneNo

- address

- dob

- joinDate

- gender

Methods:

- registerMember()

- updateProfile()

- viewWorkoutPlan()

- markAttendance()

- makePayment()

Trainer Class

Attributes:

- trainerId

- trainerName

- phoneNo

- email

- dob

Methods:

- createWorkoutPlan()

- updateWorkoutPlan()

- assignExercise()

Package Class

Attributes:

- packageId

- packageName

- price

- duration

Methods:

- updatePackage()

- viewPackageDetails()

WorkoutPlan Class

Attributes:

- planId

- durationWeeks

- goal

Methods:

- addExercise()
- removeExercise()
- updateGoal()

Exercise Class

Attributes:

- exerciseId
- exerciseName
- muscleGroup
- defaultSets
- defaultReps

Methods:

- updateExercise()

Equipment Class

Attributes:

- equipmentId
- name
- quantity

Methods:

- updateQuantity()

Attendance Class

Attributes:

- date
- checkInTime

- checkOutTime

Methods:

- markCheckIn()

- markCheckOut()

Payment Class

Attributes:

- paymentId

- amount

- mode

- timestamp

Methods:

- processPayment()

- generateReceipt()

6. Relationships (OOAD View)

- Member subscribes to Package (Association)

- Member assigned to Trainer (Association)

- Trainer creates WorkoutPlan (Association)

- Member has WorkoutPlan (Aggregation)

- WorkoutPlan contains Exercise (Composition)

- Exercise requires Equipment (Association)

- Member makes Payment (Association)

- Member marks Attendance (Association)

7. UML Diagrams Used

- Use Case Diagram
- Class Diagram
- Activity Diagram
- Sequence Diagram
- ER Diagram (for reference)
- Component Diagram

8. Use Case Description (Sample)

Use Case: Mark Attendance

Actor: Member

Description: Member records daily attendance by checking in and checking out.

Pre-condition: Member must be registered.

Post-condition: Attendance record is stored.

9. Object-Oriented Principles Applied

Encapsulation:

Each class encapsulates its data and methods.

Inheritance:

Future extensions like PremiumMember can inherit from Member.

Polymorphism:

Different payment modes implement the same processPayment() method.

Abstraction:

Only essential features are exposed to the user.

10. Non-Functional Requirements

- Scalability
- Security
- Maintainability
- Performance
- Usability

11. Conclusion

This OOAD mini project successfully models a real-world gym management system using object-oriented principles and UML diagrams.

The design ensures modularity, reusability, and future scalability.