**Gym Management System**

# Introduction:

The Gym Management System is a comprehensive software solution designed to manage a gym with multiple sites across London. It aims to boost the management of personal trainers and training sessions, providing functionalities for both gym administrators and members. The system uses Object-Oriented Programming (OOP) principles to ensure maintainability, and scalability.

# User Roles:

The Gym Management System is divided into two main user roles:

## Admin:

Admin can manage gym locations efficiently. It can also view trainers sorted alphabetically and by experience. Also, can add and remove a personal trainer. Admin can generate a report txt file for the training sessions booked on a specific date.

## Gym Member:

Gym member can check available personal trainers sorted by training session and personal trainer type. Member can book a session with a personal trainer of its own choice and can also cancel if needed.

# Functionalities:

This gym management system is designed to meet the requirements of a gym which has multiple locations and have different type of users. The functionalities are given below:

## Personal Trainers Management:

Admins can add new personal trainers, delete existing personal trainers and list trainers sorted by name or experience. Before adding a new personal trainer, its data is checked in the already existing data if same registration number exists it doesn’t add it. The system supports different types of trainers, uses inheritance and polymorphism.

## Training Sessions Management:

Users can book and cancel training sessions. Training session data is checked first then the session is booked if a session with same session code is not found. The system ensures that trainers are not double-booked by checking the availability before booking a session and also checks the gym location if it exists or not then book the session.

## Generating File:

This system allows the admin to generate a txt file for the sessions booked on a specific date which allows the admins to maintain the data of a specific date efficiently! The file is named with the date for the report which allows admin to generate multiple files and also allows him to sort them easily.

## Admin Functionalities:

Admins can add new gym locations, list trainers and generate reports on training sessions and add or remove personal trainers. The system provides sorted lists of trainers and detailed session reports.

## User Functionalities:

Users can list available trainers, filter trainers by type and sessionTime, and book/cancel sessions. The system provides a user-friendly interface for interacting with these functionalities

# Other Functionalities:

The system includes comprehensive input validation and is user-friendly. This includes:

## Input Validation:

Methods that require user input, such as adding a trainer or booking a session, include validation checks. For instance, when adding a trainer, the system validates the experience and fee to ensure they are positive numbers and also checks whether a trainer already exists with same reg no or not if found it doesn’t add trainer.

## User Feedback:

The system provides clear feedback to users on successful operations and errors. This includes confirmation messages for successful actions and error messages for invalid inputs or failed operations.

# Classes and Interfaces of System:

The system is designed using OOP principles. The classes and interfaces and their roles are as follows:

## Classes:

Following is an overview of the classes used in the system’s implementation:

### IGymManager Interface:

IGymManager interface allows admin to access following functionalities:

* + 1. Adding new gym location
    2. Adding a personel trainer
    3. Removing a personel trainer
    4. Viewing trainers alphabetically
    5. Viewing trainers by experience
    6. Generate a file for sessions booked

### IGymUser Interface:

IGymUser interface allows gym member or user to access following functionalities:

* + 1. Viewing available trainers by time available
    2. Viewing available trainers by time available and their type
    3. Booking a session
    4. Cancel a session

### GymManager Class:

GymManager class implements both the interfaces. It also manages the core functionalities of the application including managing and viewing personal trainers data by sorted methods, booking sessions and generating reports.

### PersonelTrainer Class:

It is an abstract base class for all types of trainers, encapsulating common properties like registration number, name, experience, certificates, styles and session fee. Includes derived classes:

* + 1. FitnessInstructor
    2. RelaxationSpecialist
    3. KarateInstructor

### TrainingSession Class:

Training session with properties like session code, member name, trainer, session time and gym location. It stores information of the session booked.

### GymLocation Class:

Represents a gym site with properties like site ID and location.

## Interfaces:

Following is an overview of the interfaces used in the system’s implementation:

### IGymManager Interface:

IGymManager interface allows admin to access following functionalities:

* + 1. Add new gym location
    2. Add a personel trainer
    3. Remove a personel trainer
    4. View trainers alphabetically
    5. View trainers by experience
    6. Generate a file for sessions booked

### IGymUser Interface:

IGymUser interface allows gym member or user to access following functionalities:

* + 1. View available trainers by time available
    2. View available trainers by time available and their type
    3. Book a session
    4. Cancel a session
    5. Get trainer by reg no

# OOP Concepts Used

These four core concepts Encapsulation, inheritance, polymorphism and abstraction play an important role in this application as defined below:

## Encapsulation:

Classes encapsulate data and methods, providing a clear interface for interaction while hiding internal implementation details. For example, the PersonelTrainer class encapsulates trainer details and provides methods to access and modify these details.

## Inheritance:

The system uses inheritance to create specific types of trainers from a base class. The PersonelTrainer class serves as a base class, with FitnessInstructor, RelaxationSpecialist and KarateInstructor as derived classes. This promotes code reuse and extensibility.

## Polymorphism:

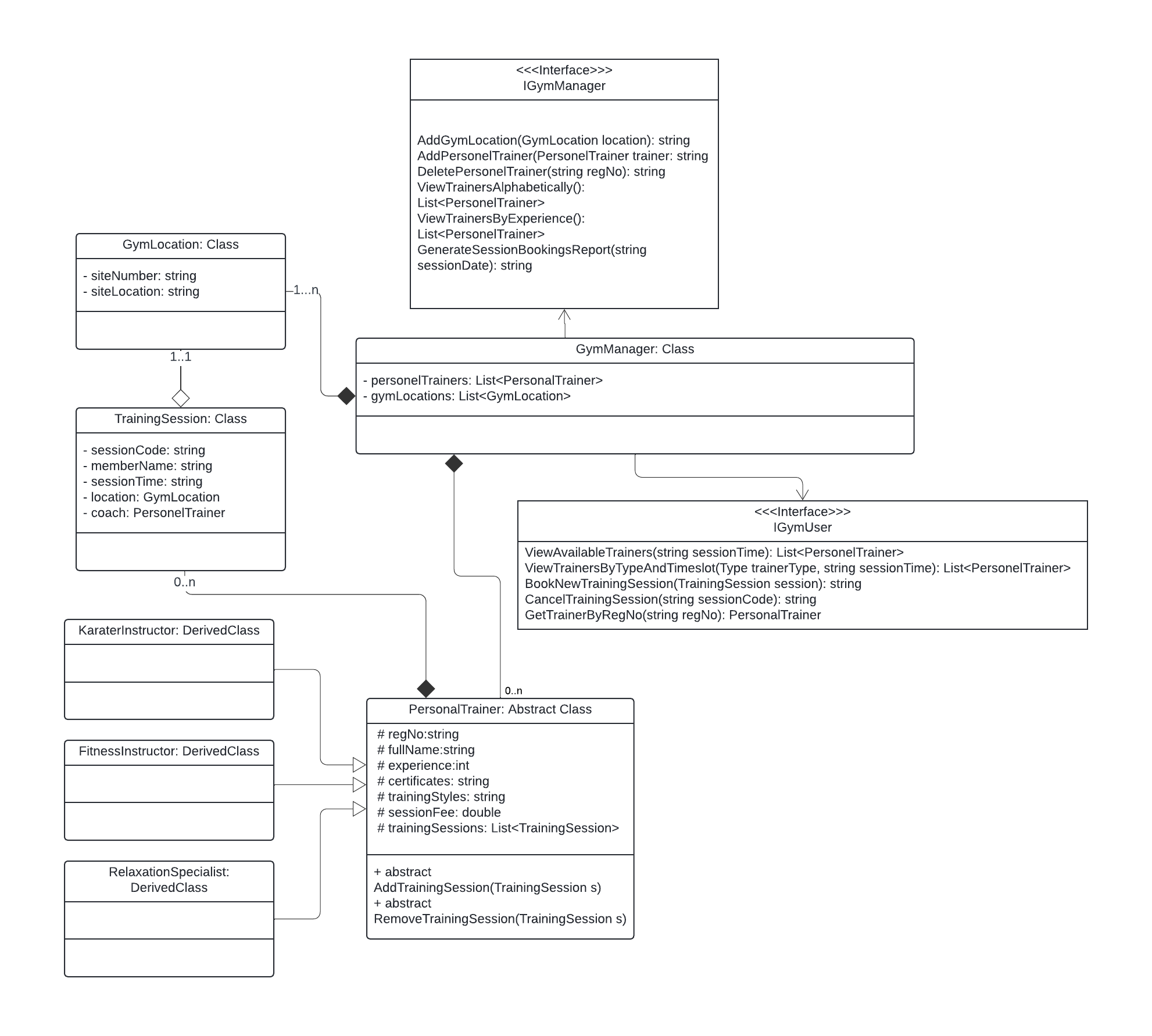
Polymorphism is achieved through method overriding in derived classes. The base class PersonelTrainer provides abstract methods that can be overridden by derived classes to provide specific implementations.

## Abstraction:

Abstraction in the is implemented through interfaces and abstract classes, which simplify complexity by focusing on essential functionalities without exposing the details of implementation. The IGymManager and IGymUser interfaces define fundamental operations for Admins and Gym Members respectively. This separation ensures clear responsibilities, enhancing the system's flexibility.

# UML Diagram Representation:

Here is an overview of the UML Diagram of the application which describes its flow:



## Relations:

The relations in each class are given as:

* + There is a One-to-many relationship between personel trainer and training sessions.
  + There is a one-to-one relationship between gymsite and training session and one session can be booked at one site only.
  + There is composition between session and trainer as session cannot exist without a trainer.
  + There is aggregation between gymsite and training session but if a gymsite does not exist session cannot be booked.

# Conclusion:

The Gym Management System is a solution for managing gym operations. It uses OOP principles to ensure maintainability. The system's design and implementation provide a comprehensive solution for managing personal trainers and training sessions, achieving the needs of both admins and gym members. The inclusion of error handling mechanisms farther enhances the system's reliability and user experience. Overall this system can help the gym’s to manage the sessions booked and members which can make it easier for them to handle it.