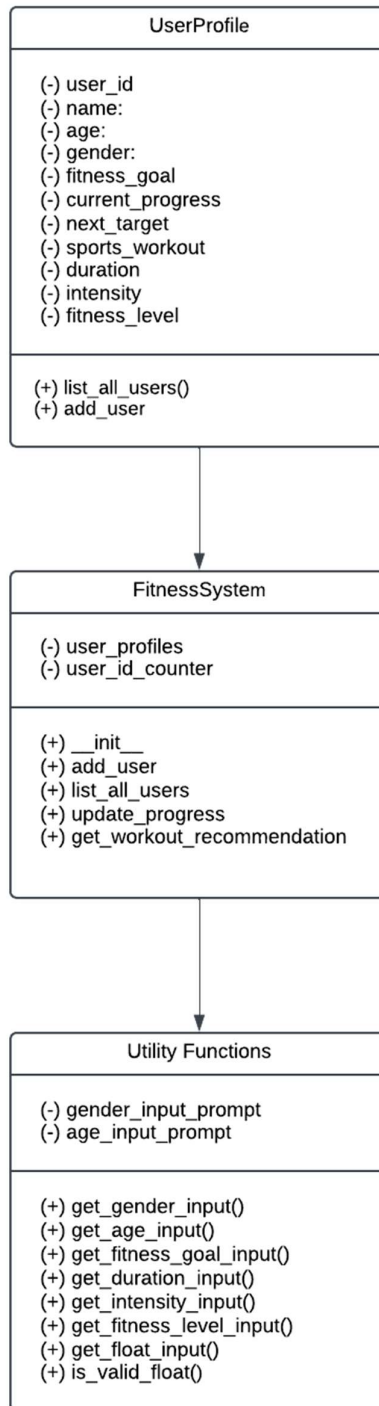


Section A

1.0 This section displays the development of a class diagram used for workout tracking and recommendation system.



2.0 This section explains the class diagram based on Question 1.0.

The UserProfile Class is the core of the system which shows the individual's user profile. The UserProfile consists of attributes which capture the main user's information and data. The user_id is used for each user to track them individually. The name is to add the user's name and age is to understand the physical and factors of the person. Gender is used to classify the person. Fitness_goals is applied to understand the fitness objective of the person and apply recommendations to the person. The current_progress and next_target helps user on their fitness journey through tracking down current progress and future targets. The sports_workout, duration, intensity, fitness_level, and score keeps the preference, ability, and the data of the person.

Next, the FitnessSystem Class handles user profiles and communication within the system. The system includes various attributes and methods. user_profiles is used to cater instances which helps to managing user data and storing it. As for, user_id_counter, integers are used here to generate unique user IDs for new profiles. The function of add_user allows addition of new user profiles. The list_all_users is used to list their profiles and the person is able to update the progress or receiving workout recommendations when needed. The update_progress is created for users to keep their profile data updated in fitness progress and next target. Then, get_workout_recommendation enables workout recommendations depending on the user's fitness level, goal, and intensity in working out.

Furthermore, utility function is used to collect and validate user inputs. The gender_input_prompt and age_input_prompt collects prompt for obtaining gender and age inputs, ensuring more engagement with the user. The get_gender_input() and get_age_input() authenticate gender and age inputs, enhancing user experience. The get_fitness_goal_input(), get_duration_input(), get_intensity_input(), and get_fitness_level_input() are used to improve user preferences related to fitness goals, workout duration, intensity, and fitness levels, ensuring the system tailors recommendations appropriately. As for get_float_input(prompt) and is_valid_float(value_str), it is used to store numeric inputs for improving data accuracy.

3.0 This section discusses on strategies for generating personalized workout and recommendations.

Fitness profiles are important as it is the core to initialise personalized workout recommendation which depends on the individual user's unique fitness profile. This can be achieved through performing evaluation of the individual's fitness status which involves workout assessments, body fit test which are recommended to be done by the individual. The assessments should be based on the level of endurance, strength, and performance of the person. Individuals are encouraged to understand and explore their fitness objectives to focus on losing weight, building muscle, enhancing stamina, or body composition. This is important as it will affect the selection of workout recommendation.

Besides that, the workout history of one's is crucial in providing tailored workout recommendations based on analysing the user's past workout history. The individuals are encouraged to record and track their workouts such as type of workout, duration, and intensity. The recorded resources stored in the system can will improve the recommendations to the person. The records can be based on the body weight of the person, the BMI of the person, the distance which the person ran or even muscle mass if it is muscle building.

In conjunction to that, one can building personalized workout plans based on the recommendations. Workout plans will include well detailed workout routines which focus on the fitness goal of the person. The aim is to allow the user to be with clear and understanding with their fitness objectives.

The variety of sports and progression are important as mixture of different exercises will improve fitness. Furthermore, workout plans should have increase in intensity for improving the user's fitness level.

Section B

This section displays an interactive user interface made from console of python for the workout tracking and recommendation system which includes a dashboard according to the project scenario.

The console is created to enable new users to create fitness profiles, including information such as age, gender, and their desired fitness objectives. Besides that, it allows users to input their information such as sports or workout the person previously has done and record the duration and intensity the person has been training with. The interface will also ask the age and gender of the person and to store the fitness goals and current progress to encourage the person and motivate the person to reach their goal more effectively. A unique ID is created for new users when they key in their details.

```
Fitness System Menu:
1. View user profile
2. Add a new user
3. Exit
Please choose 1-3: 2
Enter Name: Tre
Enter Age: 25
Enter Gender (Male/Female): Female
Select a fitness goal:
1. Weight Loss
2. Muscle Building
3. Stamina
4. General Composition
Enter the number corresponding to your fitness goal (1-4): 1
Enter Sports/Workout: Running
Enter Duration (minutes): 30
Intensity Level:
1. High
2. Medium
3. Low
Enter the number corresponding to your intensity level (1-3): 2
Fitness Level:
1. Beginner
2. Intermediate
3. Advanced
4. Expert
5. Professional
Enter the number corresponding to your fitness level (1-5): 2
Enter current weight (kg): 50
Enter next target weight (kg): 48
New user profile has been added with User ID: 1
```

The interface also allows existing users to modify their current progress and update their new target to improve their fitness journey.

```
Fitness System Menu:
1. View user profile
2. Add a new user
3. Exit
Please choose 1-3: 1
Enter User ID to view user profile: 101
Hi, Alex Johnson, what can I do for you?
Options:
1. Update current progress and next target in fitness goal
2. Workout recommendation
Enter the number corresponding to your choice (1-2): 1
Updating current progress and next target for Alex Johnson
Current Progress: weight: 85 kg
Next Target: 80 kg
Enter new current progress for Weight Loss: 78
Enter new next target for Weight Loss: 75
Current Progress and Next Target updated for Alex Johnson.
```

The interface also contains a list of routines for working out and exercise based on the user's goals and fitness level. As shown in the image below, there are about 20 workout and routines provided to cater for various individual catering for each of their needs depending on what they require in their fitness goals. Workout routines are provided based on the levels of the individual tailoring for their needs for each kind of goals. If the workout does not satisfy the person enough, they are recommended to try the more intense workout recommendation which is suggested by the interface.

```
Fitness System Menu:
1. View user profile
2. Add a new user
3. Exit
Please choose 1-3: 1
Enter User ID to view user profile: 102
Hi, Sara Smith, what can I do for you?
Options:
1. Update current progress and next target in fitness goal
2. Workout recommendation
Enter the number corresponding to your choice (1-2): 2
Here are workout recommendations for Sara Smith:
Options:
1. Regular Recommendation
2. Intense Recommendation
Enter the number corresponding to your choice (1-2): 1
Advanced weightlifting routines with progressive overload.
```

There are 10 people stored evaluate the performance to evaluate the performance of the console.

```
user_data = [
  {"user_id": 101, "name": "Alex Johnson", "age": 28, "gender": "Male", "fitness_goal": "Weight Loss", "sports_workout": "S"},
  {"user_id": 102, "name": "Sara Smith", "age": 32, "gender": "Female", "fitness_goal": "Muscle Building", "sports_workout": "S"},
  {"user_id": 103, "name": "Michael Brown", "age": 22, "gender": "Male", "fitness_goal": "Stamina", "sports_workout": "S"},
  {"user_id": 104, "name": "Taylor Emanuel", "age": 35, "gender": "Female", "fitness_goal": "General Composition", "sports_workout": "S"},
  {"user_id": 105, "name": "Bruce Lee", "age": 40, "gender": "Male", "fitness_goal": "Muscle Building", "sports_workout": "S"},
  {"user_id": 106, "name": "Jessica Lea", "age": 25, "gender": "Female", "fitness_goal": "Weight Loss", "sports_workout": "S"},
  {"user_id": 107, "name": "Peggy Carter", "age": 30, "gender": "Male", "fitness_goal": "Stamina", "sports_workout": "S"},
  {"user_id": 108, "name": "Lara Milson", "age": 27, "gender": "Female", "fitness_goal": "General Composition", "sports_workout": "S"},
  {"user_id": 109, "name": "Robert Loh", "age": 35, "gender": "Male", "fitness_goal": "Muscle Building", "sports_workout": "S"},
  {"user_id": 110, "name": "Elizabeth Hugh", "age": 29, "gender": "Female", "fitness_goal": "Weight Loss", "sports_workout": "S"},
  {"user_id": 111, "name": "James Jamerson", "age": 24, "gender": "Male", "fitness_goal": "Stamina", "sports_workout": "S"},
  {"user_id": 112, "name": "Angela Rawat", "age": 31, "gender": "Female", "fitness_goal": "General Composition", "sports_workout": "S"}
]

intensity": "Moderate", "fitness_level": "Intermediate", "current_progress": "weight: 85 kg", "next_target": "80 kg"},
ur", "intensity": "High", "fitness_level": "Advanced", "current_progress": "muscle mass: 24%", "next_target": "26%"},
tensity": "Low", "fitness_level": "Beginner", "current_progress": "running distance: 5 km", "next_target": "7 km"},
hour", "intensity": "Moderate", "fitness_level": "Intermediate", "current_progress": "bmi: 23", "next_target": "22"},
hours", "intensity": "High", "fitness_level": "Expert", "current_progress": "muscle mass: 27%", "next_target": "29%"},
intensity": "Moderate", "fitness_level": "Beginner", "current_progress": "weight: 75 kg", "next_target": "next target: 70 kg"},
intensity": "High", "fitness_level": "Advanced", "current_progress": "running distance: 15 km", "next_target": "18 km"},
mins", "intensity": "Moderate", "fitness_level": "Intermediate", "current_progress": "bmi: 24", "next_target": "23"},
r", "intensity": "High", "fitness_level": "Professional", "current_progress": "muscle mass: 30%", "next_target": "32%"},
ng", "duration": "40 mins", "intensity": "High", "fitness_level": "Intermediate", "current_progress": "weight: 68 kg", "next_target": "65 kg"},
5 hours", "intensity": "High", "fitness_level": "Advanced", "current_progress": "running distance: 20 km", "next_target": "25 km"},
r", "intensity": "Moderate", "fitness_level": "Intermediate", "current_progress": "bmi: 22", "next_target": "21"}
```

The image below displays the 20 types of routine or workouts stored in the interface for users to access to depending on the fitness goals.

```
def get_workout_recommendation(self, user):
    print(f"Here are workout recommendations for {user.name}:")

    # Define workout recommendations based on fitness goals and levels
    regular_recommendations = {
        "Beginner": {
            "Weight Loss": "Beginner's cardio workouts like brisk walking or cycling for 30 minutes a day.",
            "Muscle Building": "Bodyweight exercises like push-ups, squats, and lunges.",
            "Stamina": "Start with light jogging and gradually increase the distance.",
            "General Composition": "Yoga and Pilates for flexibility and core strength.",
        },
        "Intermediate": {
            "Weight Loss": "Interval training with a mix of cardio and strength exercises.",
            "Muscle Building": "Strength training with weights and compound exercises.",
            "Stamina": "HIIT workouts and longer runs or cycling sessions.",
            "General Composition": "High-intensity interval workouts and functional training.",
        },
        "Advanced": {
            "Weight Loss": "Advanced HIIT workouts and circuit training.",
            "Muscle Building": "Advanced weightlifting routines with progressive overload.",
            "Stamina": "Long-distance running or intense cycling sessions.",
            "General Composition": "High-intensity functional training and core workouts.",
        },
        "Expert": {
            "Weight Loss": "Extreme cardio and circuit workouts.",
            "Muscle Building": "Advanced bodybuilding routines and powerlifting.",
            "Stamina": "Marathon training and advanced endurance exercises.",
            "General Composition": "Advanced functional training and agility workouts.",
        },
        "Professional": {
            "Weight Loss": "Customized high-intensity routines with a focus on calorie burn.",
            "Muscle Building": "Professional bodybuilding and strength training programs.",
            "Stamina": "Professional endurance training and triathlon preparation.",
            "General Composition": "Customized workouts designed by fitness professionals.",
        }
    }
```

```
intense_recommendations = {
    "Beginner": {
        "Weight Loss": "Start daily cardio with activities like brisk walking or cycling for 30 minutes.",
        "Muscle Building": "Begin with bodyweight exercises like push-ups, squats, and lunges.",
        "Stamina": "Begin stamina-building with light jogging, then increase your distance gradually.",
        "General Composition": "Work on core strength and flexibility through yoga and Pilates."
    },
    "Intermediate": {
        "Weight Loss": "Try interval training, combining cardio and strength exercises.",
        "Muscle Building": "Lift weights and do compound exercises for muscle growth.",
        "Stamina": "Do HIIT workouts and longer runs or cycling sessions.",
        "General Composition": "Focus on high-intensity interval workouts and functional training."
    },
    "Advanced": {
        "Weight Loss": "Do advanced HIIT workouts and circuit training to burn calories.",
        "Muscle Building": "Follow advanced weightlifting routines with progressive overload.",
        "Stamina": "Challenge yourself with long-distance runs and intense cycling sessions.",
        "General Composition": "Engage in high-intensity functional training and core workouts."
    },
    "Expert": {
        "Weight Loss": "Try extreme cardio and circuit workouts for fat loss.",
        "Muscle Building": "Become a true strength titan with advanced bodybuilding routines and powerlifting.",
        "Stamina": "Train for marathons and advanced endurance exercises.",
        "General Composition": "Master advanced functional training and agility workouts for a sculpted physique."
    },
    "Professional": {
        "Weight Loss": "Engage in customized high-intensity routines for maximum calorie burn.",
        "Muscle Building": "Join professional bodybuilding and strength training programs.",
        "Stamina": "Prepare for triathlons with professional endurance training.",
        "General Composition": "Follow customized workouts crafted by fitness professionals for an exceptional physique."
    }
}
```


The interface also includes a function which encourages users to improving workout consistency and long-term fitness goals.

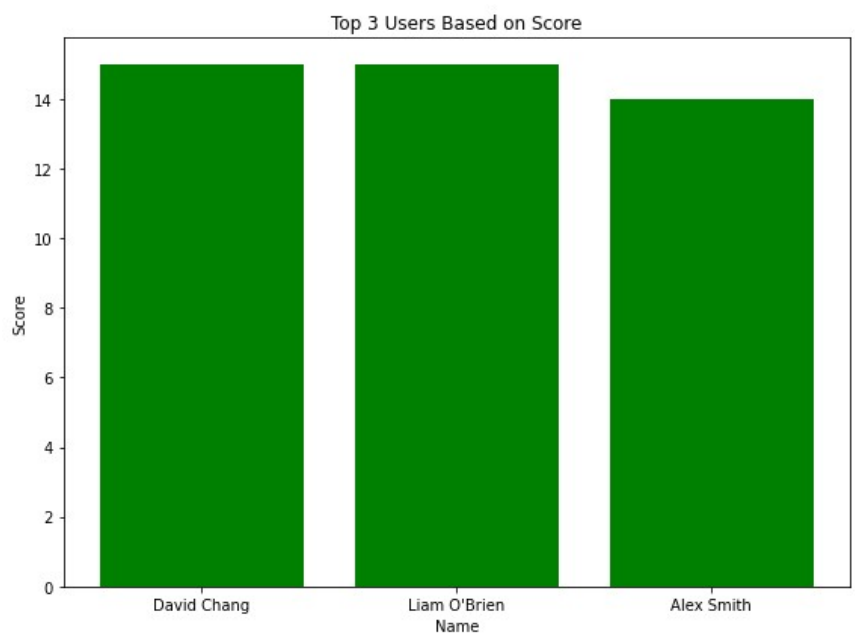
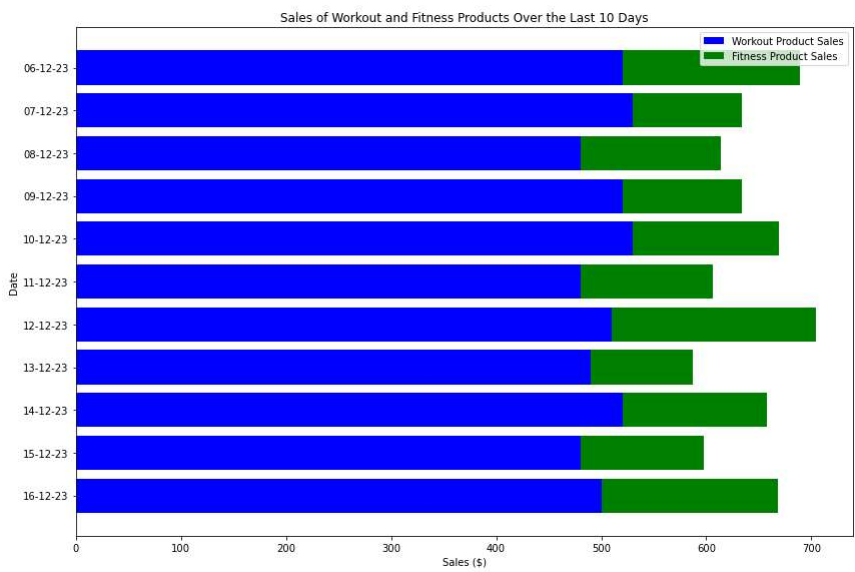
```
Fitness System Menu:
1. View user profile
2. Add a new user
3. Exit
Please choose 1-3: 1
Enter User ID to view user profile: 101
Hi, Alex Johnson, what can I do for you?
Options:
1. Update current progress and next target in fitness goal
2. Workout recommendation
3. Get Guidance for workout consistency and long-term goals
Enter the number corresponding to your choice (1-3): 3
Guidance for Alex Johnson (Weight Loss):
To achieve your weight loss goal:
- Maintain a balanced diet with a calorie deficit.
- Incorporate cardio workouts and strength training.
- Stay consistent with your exercise routine.
```

```
# Check if the user's fitness goal is related to weight loss or muscle building
if user.fitness_goal in ("Weight Loss", "Muscle Building"):
    # Provide generic guidance for weight loss or muscle building
    if user.fitness_goal == "Weight Loss":
        print("To achieve your weight loss goal:")
        print("- Maintain a balanced diet with a calorie deficit.")
        print("- Incorporate cardio workouts and strength training.")
        print("- Stay consistent with your exercise routine.")
    elif user.fitness_goal == "Muscle Building":
        print("To build muscle effectively:")
        print("- Focus on progressive resistance training.")
        print("- Consume enough protein to support muscle growth.")
        print("- Allow for proper recovery between workouts.")
elif user.fitness_goal in ("Stamina", "General Composition"):
    # Provide guidance for stamina or general composition
    if user.fitness_goal == "Stamina":
        print("To improve your stamina:")
        print("- Incorporate regular cardiovascular exercises.")
        print("- Gradually increase the duration and intensity of your workouts.")
        print("- Include interval training to boost endurance.")
    elif user.fitness_goal == "General Composition":
        print("For general composition improvement:")
        print("- Combine strength and flexibility exercises.")
        print("- Include yoga or Pilates for core strength and flexibility.")
        print("- Maintain a balanced diet to support overall health.")
else:
    print("Guidance for this fitness goal is not available.")
```

This section demonstrates a development of an administrative console which is accessible through a unique login key (12345) which is used for system administrators.

```
Enter the login key: 12345
Admin Console Menu
1. Display Workout and Fitness Product Sales Chart
2. Display Top Users by Score
3. Gracefully Shut Down the System / Exit
Please choose 1-3: |
```

The system provides an access to sales of past 10 days through a chart to visualise the amount of workout and fitness products sold. Besides that, the top three fitness users will be spotted and generated in the chart.



The console will shut down at the call of 3 to end.

```
Admin Console Menu
1. Display Workout and Fitness Product Sales Chart
2. Display Top Users by Score
3. Gracefully Shut Down the System / Exit
Please choose 1-3: 2
Admin Console Menu
1. Display Workout and Fitness Product Sales Chart
2. Display Top Users by Score
3. Gracefully Shut Down the System / Exit
Please choose 1-3: 3
Shutting down the system for maintenance or updates...
```

This section shows the screenshots of the codes.

```
class UserProfile:
    def __init__(self, user_id=None, name=None, age=None, gender=None, fitness_goal=None, sports_workout=None, duration=None,
                  intensity=None, fitness_level=None, score=None, current_progress=None, next_target=None):
        self.user_id = user_id
        self.name = name
        self.age = age
        self.gender = gender
        self.fitness_goal = fitness_goal
        self.current_progress = current_progress
        self.next_target = next_target
        self.sports_workout = sports_workout
        self.duration = duration
        self.intensity = intensity
        self.fitness_level = fitness_level
        self.score = score

class FitnessSystem:
    def __init__(self):
        self.user_profiles = []
        self.user_id_counter = 1 # Initialize the counter for generating unique user IDs

    def list_all_users(self):
        user_id = input("Enter User ID to view user profile: ")
        found = False
        try:
            user_id = int(user_id) # Convert the input to an integer
            for user in self.user_profiles:
                if user.user_id == user_id:
                    print(f"Hi, {user.name}, what can I do for you?")
                    print("Options:")
                    print("1. Update current progress and next target in fitness goal")
                    print("2. Workout recommendation")
                    print("3. Get Guidance for workout consistency and Long-term goals") # Added option for guidance
                    option = input("Enter the number corresponding to your choice (1-3): ")
                    if option == "1":
                        self.update_progress(user)
                    elif option == "2":
                        self.get_workout_recommendation(user)
                    elif option == "3": # Call provide_guidance for option 3
                        self.provide_guidance(user)
                    else:
                        print("Invalid option. Please choose 1, 2, or 3.")
                        found = True
                        break
        except ValueError:
            pass

        if not found:
            print(f"User with User ID {user_id} not found.")

    def add_user(self, user):
        if user.user_id is None:
            user.user_id = self.user_id_counter
            self.user_id_counter += 1
        self.user_profiles.append(user)

    def update_progress(self, user):
        print(f"Updating current progress and next target for {user.name}")
        if user.fitness_goal in ("Weight Loss", "Muscle Building", "Stamina", "General Composition"):
            print(f"Current Progress: {user.current_progress}")
            print(f"Next Target: {user.next_target}")

            # Allow the user to update the current progress and next target
            current_progress = input(f"Enter new current progress for {user.fitness_goal}: ")
            next_target = input(f"Enter new next target for {user.fitness_goal}: ")

            # Update the user's current progress and next target
            user.current_progress = current_progress
            user.next_target = next_target
            print(f"Current Progress and Next Target updated for {user.name}.")

    def get_workout_recommendation(self, user):
        print(f"Here are workout recommendations for {user.name}:")

        # Define workout recommendations based on fitness goals and levels
        regular_recommendations = {
            "Beginner": {
                "Weight Loss": "Beginner's cardio workouts like brisk walking or cycling for 30 minutes a day.",
                "Muscle Building": "Bodyweight exercises like push-ups, squats, and lunges.",
                "Stamina": "Start with light jogging and gradually increase the distance.",
                "General Composition": "Yoga and Pilates for flexibility and core strength.",
            },
            "Intermediate": {
                "Weight Loss": "Interval training with a mix of cardio and strength exercises.",
                "Muscle Building": "Strength training with weights and compound exercises.",
                "Stamina": "HIIT workouts and longer runs or cycling sessions.",
                "General Composition": "High-intensity interval workouts and functional training.",
            }
        }
```

```

        "Stamina": "HIIT workouts and longer runs or cycling sessions.",
        "General Composition": "High-intensity interval workouts and functional training.",
    },
    "Advanced": {
        "Weight Loss": "Advanced HIIT workouts and circuit training.",
        "Muscle Building": "Advanced weightlifting routines with progressive overload.",
        "Stamina": "Long-distance running or intense cycling sessions.",
        "General Composition": "High-intensity functional training and core workouts.",
    },
    "Expert": {
        "Weight Loss": "Extreme cardio and circuit workouts.",
        "Muscle Building": "Advanced bodybuilding routines and powerlifting.",
        "Stamina": "Marathon training and advanced endurance exercises.",
        "General Composition": "Advanced functional training and agility workouts.",
    },
    "Professional": {
        "Weight Loss": "Customized high-intensity routines with a focus on calorie burn.",
        "Muscle Building": "Professional bodybuilding and strength training programs.",
        "Stamina": "Professional endurance training and triathlon preparation.",
        "General Composition": "Customized workouts designed by fitness professionals.",
    },
}

intense_recommendations = {
    "Beginner": {
        "Weight Loss": "Start daily cardio with activities like brisk walking or cycling for 30 minutes.",
        "Muscle Building": "Begin with bodyweight exercises like push-ups, squats, and lunges.",
        "Stamina": "Begin stamina-building with light jogging, then increase your distance gradually.",
        "General Composition": "Work on core strength and flexibility through yoga and Pilates."
    },
    "Intermediate": {
        "Weight Loss": "Try interval training, combining cardio and strength exercises.",
        "Muscle Building": "Lift weights and do compound exercises for muscle growth.",
        "Stamina": "Do HIIT workouts and longer runs or cycling sessions.",
        "General Composition": "Focus on high-intensity interval workouts and functional training."
    },
    "Advanced": {
        "Weight Loss": "Do advanced HIIT workouts and circuit training to burn calories.",
        "Muscle Building": "Follow advanced weightlifting routines with progressive overload.",
        "Stamina": "Challenge yourself with long-distance runs and intense cycling sessions.",
        "General Composition": "Engage in high-intensity functional training and core workouts."
    },
    "Expert": {
        "Weight Loss": "Try extreme cardio and circuit workouts for fat loss.",
        "Muscle Building": "Become a true strength titan with advanced bodybuilding routines and powerlifting."
    },
    "Professional": {
        "Weight Loss": "Try extreme cardio and circuit workouts for fat loss.",
        "Muscle Building": "Become a true strength titan with advanced bodybuilding routines and powerlifting.",
        "Stamina": "Train for marathons and advanced endurance exercises.",
        "General Composition": "Master advanced functional training and agility workouts for a sculpted physique."
    },
}

}

fitness_level = user.fitness_level
fitness_goal = user.fitness_goal

if fitness_level in regular_recommendations and fitness_goal in regular_recommendations[fitness_level]:
    recommendation = regular_recommendations[fitness_level][fitness_goal]
    print("Options:")
    print("1. Regular Recommendation")
    print("2. Intense Recommendation")
    option = input("Enter the number corresponding to your choice (1-2): ")
    if option == "1":
        print(recommendation)
    elif option == "2":
        intense_recommendation = intense_recommendations[fitness_level][fitness_goal]
        print(f"Intense Version of Workout Recommendation:\n{intense_recommendation}")
    else:
        print("Invalid option. Please choose 1 or 2.")
else:
    print("Sorry, we don't have specific recommendations for your fitness level and goal.")

def provide_guidance(self, user):
    print(f"Guidance for {user.name} ({user.fitness_goal}):")

    # Check if the user's fitness goal is related to weight loss or muscle building
    if user.fitness_goal in ("Weight Loss", "Muscle Building"):
        # Provide generic guidance for weight loss or muscle building
        if user.fitness_goal == "Weight Loss":
            print("To achieve your weight loss goal:")
            print("- Maintain a balanced diet with a calorie deficit.")
            print("- Incorporate cardio workouts and strength training.")
            print("- Stay consistent with your exercise routine.")
        elif user.fitness_goal == "Muscle Building":

```



```

        print("Incorporate cardio workouts and strength training.")
        print("- Stay consistent with your exercise routine.")
    elif user.fitness_goal == "Muscle Building":
        print("To build muscle effectively:")
        print("- Focus on progressive resistance training.")
        print("- Consume enough protein to support muscle growth.")
        print("- Allow for proper recovery between workouts.")
    elif user.fitness_goal in ("Stamina", "General Composition"):
        # Provide guidance for stamina or general composition
        if user.fitness_goal == "Stamina":
            print("To improve your stamina:")
            print("- Incorporate regular cardiovascular exercises.")
            print("- Gradually increase the duration and intensity of your workouts.")
            print("- Include interval training to boost endurance.")
        elif user.fitness_goal == "General Composition":
            print("For general composition improvement:")
            print("- Combine strength and flexibility exercises.")
            print("- Include yoga or Pilates for core strength and flexibility.")
            print("- Maintain a balanced diet to support overall health.")
    else:
        print("Guidance for this fitness goal is not available.")

def get_gender_input():
    while True:
        gender = input("Enter Gender (Male/Female): ").strip().capitalize() # Convert input to title case
        if gender in ("Male", "Female"):
            return gender
        else:
            print("Invalid gender. Please enter 'Male' or 'Female'.")

def get_age_input():
    while True:
        age_str = input("Enter Age: ").strip()
        if age_str.isdigit():
            age = int(age_str)
            if 10 <= age <= 100:
                return age
            else:
                print("Age must be between 10 and 100.")
        else:
            print("Invalid age. Please enter a valid whole number.")

def get_fitness_goal_input():
    while True:
        print("Select a fitness goal:")
        print("1. Weight Loss")
        print("2. Muscle Building")
        print("3. Stamina")
        print("4. General Composition")
        choice = input("Enter the number corresponding to your fitness goal (1-4): ")
        if choice in ("1", "2", "3", "4"):
            return {
                "1": "Weight Loss",
                "2": "Muscle Building",
                "3": "Stamina",
                "4": "General Composition"
            }[choice]
        else:
            print("Invalid choice. Please enter a number between 1 and 4.")

def get_duration_input():
    while True:
        duration_str = input("Enter Duration (minutes): ").strip()
        if duration_str.isdigit():
            duration = int(duration_str)
            if duration > 0:
                return duration
            else:
                print("Duration must be greater than 0 minutes.")
        else:
            print("Invalid duration. Please enter a valid number of minutes.")

def get_intensity_input():
    while True:
        print("Intensity Level:")
        print("1. High")
        print("2. Medium")
        print("3. Low")
        choice = input("Enter the number corresponding to your intensity level (1-3): ")
        if choice in ("1", "2", "3"):
            return {
                "1": "High",
                "2": "Medium",
                "3": "Low"
            }[choice]
        else:
            print("Invalid intensity level. Please enter a number between 1 and 3.")

```

```

def get_fitness_level_input():
    while True:
        print("Fitness Level:")
        print("1. Beginner")
        print("2. Intermediate")
        print("3. Advanced")
        print("4. Expert")
        print("5. Professional")
        choice = input("Enter the number corresponding to your fitness level (1-5): ")
        if choice in ("1", "2", "3", "4", "5"):
            return {
                "1": "Beginner",
                "2": "Intermediate",
                "3": "Advanced",
                "4": "Expert",
                "5": "Professional"
            }[choice]
        else:
            print("Invalid choice. Please enter a number between 1 and 5.")

def get_float_input(prompt):
    while True:
        value_str = input(prompt).strip()
        if is_valid_float(value_str):
            return float(value_str)
        else:
            print("Invalid input. Please enter a valid number.")

def is_valid_float(value_str):
    try:
        float(value_str)
        return True
    except ValueError:
        return False

def main():
    fitness_system = FitnessSystem()

    user_data = [
        {"user_id": 101, "name": "Alex Johnson", "age": 28, "gender": "Male", "fitness_goal": "Weight Loss", "sports_workout": "Cardio"},
        {"user_id": 102, "name": "Sara Smith", "age": 32, "gender": "Female", "fitness_goal": "Muscle Building", "sports_workout": "Strength Training"},
        {"user_id": 103, "name": "Michael Brown", "age": 22, "gender": "Male", "fitness_goal": "Stamina", "sports_workout": "Endurance"},
        {"user_id": 104, "name": "Taylor Emanuel", "age": 35, "gender": "Female", "fitness_goal": "General Composition", "sports_workout": "Mixed"},
        {"user_id": 105, "name": "Bruce Lee", "age": 40, "gender": "Male", "fitness_goal": "Muscle Building", "sports_workout": "Strength Training"},
        {"user_id": 106, "name": "Jessica Lea", "age": 25, "gender": "Female", "fitness_goal": "Weight Loss", "sports_workout": "Cardio"},
        {"user_id": 107, "name": "Peggy Carter", "age": 30, "gender": "Male", "fitness_goal": "Stamina", "sports_workout": "Endurance"},
        {"user_id": 108, "name": "Lara Milson", "age": 27, "gender": "Female", "fitness_goal": "General Composition", "sports_workout": "Mixed"},
        {"user_id": 109, "name": "Robert Loh", "age": 35, "gender": "Male", "fitness_goal": "Muscle Building", "sports_workout": "Strength Training"},
        {"user_id": 110, "name": "Elizabeth Hugh", "age": 29, "gender": "Female", "fitness_goal": "Weight Loss", "sports_workout": "Cardio"},
        {"user_id": 111, "name": "James Jamerson", "age": 24, "gender": "Male", "fitness_goal": "Stamina", "sports_workout": "Endurance"},
        {"user_id": 112, "name": "Angela Rawa", "age": 31, "gender": "Female", "fitness_goal": "General Composition", "sports_workout": "Mixed"}
    ]

    intensity": "Moderate", "fitness_level": "Intermediate", "current_progress": "weight: 85 kg", "next_target": "80 kg"},
    ur", "intensity": "High", "fitness_level": "Advanced", "current_progress": "muscle mass: 24%", "next_target": "26%"},
    tensity": "Low", "fitness_level": "Beginner", "current_progress": "running distance: 5 km", "next_target": "7 km"},
    hour", "intensity": "Moderate", "fitness_level": "Intermediate", "current_progress": "bmi: 23", "next_target": "22"},
    hours", "intensity": "High", "fitness_level": "Expert", "current_progress": "muscle mass: 27%", "next_target": "29%"},
    intensity": "Moderate", "fitness_level": "Beginner", "current_progress": "weight: 75 kg", "next_target": "next target: 70 kg"},
    "intensity": "High", "fitness_level": "Advanced", "current_progress": "running distance: 15 km", "next_target": "18 km"},
    mins", "intensity": "Moderate", "fitness_level": "Intermediate", "current_progress": "bmi: 24", "next_target": "23"},
    r", "intensity": "High", "fitness_level": "Professional", "current_progress": "muscle mass: 30%", "next_target": "32%"},
    ng", "duration": "40 mins", "intensity": "High", "fitness_level": "Intermediate", "current_progress": "weight: 68 kg", "next_target": "65 kg"},
    5 hours", "intensity": "High", "fitness_level": "Advanced", "current_progress": "running distance: 20 km", "next_target": "25 km"},
    r", "intensity": "Moderate", "fitness_level": "Intermediate", "current_progress": "bmi: 22", "next_target": "21"}

```

```

for user_data_entry in user_data:
    new_user = UserProfile(**user_data_entry)
    fitness_system.add_user(new_user)

while True:
    print("\nFitness System Menu:")
    print("1. View user profile")
    print("2. Add a new user")
    print("3. Exit")
    option = input("Please choose 1-3: ")
    if option == "1":
        fitness_system.list_all_users()
    elif option == "2":
        name = input("Enter Name: ")
        age = get_age_input()
        gender = get_gender_input()
        fitness_goal = get_fitness_goal_input()
        sports_workout = input("Enter Sports/Workout: ")
        duration = get_duration_input()
        intensity = get_intensity_input()
        fitness_level = get_fitness_level_input()

        current_progress = None
        next_target = None
        if fitness_goal in ("Weight Loss", "Muscle Building", "Stamina", "General Composition"):
            if fitness_goal == "Weight Loss":
                current_progress = f"current weight: {get_float_input('Enter current weight (kg): ')} kg"
                next_target = f"next target in kg: {get_float_input('Enter next target weight (kg): ')} kg"
            elif fitness_goal == "Muscle Building":
                current_progress = f"current muscle mass: {get_float_input('Enter current muscle mass (%): ')} %"
                next_target = f"next target in %: {get_float_input('Enter next target muscle mass (%): ')} %"
            elif fitness_goal == "Stamina":
                current_progress = f"current running distance: {get_float_input('Enter current running distance (km): ')} km"
                next_target = f"next target in km: {get_float_input('Enter next target running distance (km): ')} km"
            elif fitness_goal == "General Composition":
                current_progress = f"current BMI: {get_float_input('Enter current BMI: ')}"
                next_target = f"next target BMI: {get_float_input('Enter next target BMI: ')}"

```

```

        new_user = UserProfile(name=name, age=age, gender=gender, fitness_goal=fitness_goal,
                               sports_workout=sports_workout, duration=duration, intensity=intensity,
                               fitness_level=fitness_level, current_progress=current_progress, next_target=next_target)
        fitness_system.add_user(new_user)
        print(f"New user profile has been added with User ID: {new_user.user_id}")
    elif option == "3":
        print("Thank you, have a good day!")
        break
    else:
        print("Option unavailable, please try 1-3 again.")

if __name__ == "__main__":
    main()

```