Fitness Tracker & Gym Management System

HW1: Service Description

Course: Databases Project 2025
Assignment: 1
Team Members: Aleksandr Zinovev, Siwoo Lee, Arslan Ahmet Berk

Project Overview

FitTrack Pro is a web app for tracking workouts and managing gym stuff. People can log their exercises and see their progress, while gym staff can manage members and equipment. It's basically a fitness tracker that also handles gym operations.

Functionality Description

User Experience Overview

The app has three types of users: regular people who want to track fitness, gym members, and gym staff.

Regular Users can make an account, log their workouts, and track their progress. They can set fitness goals and see charts of their improvement. The app has a database of exercises with instructions so users know how to do them properly.

Gym Members get extra features like booking fitness classes, reserving equipment when the gym is busy, and scheduling sessions with trainers. They can also share workouts with other gym members and join gym challenges.

Gym Staff can manage member accounts, create workout programs, schedule classes, and keep track of equipment maintenance. Trainers can assign specific workouts to clients and monitor how they're doing.

The interface should be easy to use on both computers and phones. The main goal is connecting personal fitness tracking with gym management in one system.

Entity-Relationship Diagram

ER Diagram with ISA Hierarchies

Text Version:

ISA Hierarchy: User Entity

USER (Main table) - user_id (PK), username, email, password_hash - first_name, last_name, date_of_birth, gender - registration_date, profile_picture

ISA Hierarchy (inheritance): - **INDIVIDUAL USER** gets all User fields + fitness_level, primary_goals, workout_types - **GYM MEMBER** gets all User fields + membership_id, membership_type, start_date, end_date, gym_id (FK) - **STAFF** gets all User fields + employee_id, position, hire_date, salary, gym_id (FK)

Staff ISA Hierarchy

STAFF (Inherits from User) - All User fields + employee_id, position, hire_date, salary, gym_id (FK)

ISA Hierarchy (inheritance): - **TRAINER** gets all Staff fields + certification, specialization, hourly_rate - **MANAGER** gets all Staff fields + department, access_level, bonus_eligible - **RECEPTIONIST** gets all Staff fields + shift_schedule, languages_spoken

Exercise ISA Hierarchy

EXERCISE (Main table) - exercise_id (PK), name, category, muscle_groups, difficulty, instructions, equipment_needed

ISA Hierarchy (inheritance): - **CARDIO** gets all Exercise fields + target_heart_rate, intensity_level - **STRENGTH** gets all Exercise fields + weight_range, rep_range, muscle_focus - **FLEXIBILITY** gets all Exercise fields + stretch_duration, flexibility_level

Core Entities and Relationships

WORKOUT - workout_id (PK), user_id (FK), workout_name, date, duration, calories_burned, notes - One user can have many workouts (1:N)

EXERCISE - exercise_id (PK), name, category, muscle_groups, difficulty, instructions, equipment_needed - Many exercises can be in many workouts (M:N)

WORKOUT_EXERCISE (Junction Table) - workout_id (FK), exercise_id (FK), sets, reps, weight, duration, rest_time - Connects workouts and exercises (M:N relationship)

GYM - gym_id (PK), name, address, phone, email, operating_hours, facilities - One gym has many members and staff (1:N)

CLASS - class_id (PK), gym_id (FK), trainer_id (FK), name, description, schedule_time, duration, max_capacity - Gyms offer classes, staff teach them (1:N)

CLASS_BOOKING - booking_id (PK), class_id (FK), member_id (FK), booking_date, status - Members can book classes (M:N)

EQUIPMENT - equipment_id (PK), gym_id (FK), name, type, status, purchase_date, maintenance_schedule - Gyms own equipment (1:N)

PROGRESS_TRACKING - tracking_id (PK), user_id (FK), date, weight, body_fat_percentage, muscle_mass, measurements - Users track their progress over time (1:N)

TRAINER (ISA: Staff) - All Staff fields + certification, specialization, hourly_rate

MANAGER (ISA: Staff) - All Staff fields + department, access_level, bonus_eligible

RECEPTIONIST (ISA: Staff) - All Staff fields + shift_schedule, languages_spoken

CARDIO (ISA: Exercise) - All Exercise fields + target_heart_rate, intensity_level

STRENGTH (ISA: Exercise) - All Exercise fields + weight_range, rep_range, muscle_focus

FLEXIBILITY (ISA: Exercise) - All Exercise fields + stretch_duration, flexibility_level

Main Relationships

1. **User → Workout** (1:N): Users can log multiple workouts
2. **Workout ↔ Exercise** (M:N): Workouts contain exercises, exercises used in multiple workouts
3. **Gym → Member** (1:N): Gyms have multiple members
4. **Gym → Staff** (1:N): Gyms employ staff

5. **Gym → Class** (1:N): Gyms offer fitness classes
6. **Member ↔ Class** (M:N): Members book classes, classes have multiple attendees
7. **Gym → Equipment** (1:N): Gyms own equipment

ISA Hierarchies (3 Required for 3-Person Team)

1. **Users ISA Hierarchy**
   - **User** (superclass) → **Individual User**, **Gym Member**, **Staff**
2. **Staff ISA Hierarchy**
   - **Staff** (superclass) → **Trainer**, **Manager**, **Receptionist**
3. **Exercise ISA Hierarchy**
   - **Exercise** (superclass) → **Cardio**, **Strength**, **Flexibility**

User Interactions

What Users Can Do

1. **Account Stuff**
   - **See:** Registration page, login screen, profile page
   - **Do:** Sign up, log in/out, edit profile, upload photo
   - **What happens:** Get personal dashboard, settings saved
   - **Errors:** Wrong password, email already used, weak password
2. **Workout Logging**
   - **See:** Exercise list, workout forms, progress graphs
   - **Do:** Search exercises, create workouts, log what they did (sets/reps/weight)
   - **What happens:** Workout saved, can see history and progress
   - **Errors:** Missing data, invalid numbers
3. **Progress Tracking**
   - **See:** Charts, reports, goal tracking
   - **Do:** Enter weight/measurements, set goals, check history
   - **What happens:** See progress over time, track goal completion
   - **Errors:** Unrealistic numbers, wrong dates
4. **Social Stuff** (Gym Members Only)
   - **See:** Other members, shared workouts, challenges
   - **Do:** Share workouts, join challenges, follow people
   - **What happens:** Connect with gym community
   - **Errors:** Privacy issues, inappropriate content
5. **Class Booking** (Gym Members Only)

  - **See:** Class schedule, equipment availability
  - **Do:** Book classes, reserve equipment, cancel bookings
  - **What happens:** Spot reserved, get reminders
  - **Errors:** Class full, double booking

6. **Admin Stuff** (Staff Only)
   - **See:** Member list, equipment status, class schedules
   - **Do:** Manage members, schedule classes, track equipment maintenance
   - **What happens:** Gym runs smoothly, generate reports
   - **Errors:** No permission, scheduling conflicts

Input Validation & Security

- **Authentication:** Secure password requirements, session timeout, role-based access control
- **Data Validation:** Numerical ranges for measurements, date validation, exercise form validation
- **Privacy Protection:** Personal data encryption, workout privacy settings, secure data transmission
- **Error Prevention:** Real-time form validation, confirmation dialogs for critical actions, backup/recovery systems

Technical Implementation Notes

Technology Stack: LAMP with Python (FastAPI)
Database: MySQL with normalized schema design
Architecture: Layered architecture focusing on application logic layer
Scalability: Designed for easy expansion with additional team members and features

---