GYM
MEMBERSHIP MANAGEMENT SYSTEM

A simple web system to make gym management faster, clearer, and more organized.

Lecture: Sek Socheat

Presented by
Chang Eurping • Rattanak Anou • Song Visal • Mey Thanuth

Why did we choose this project?

The problem is simple: everyday gym records can become messy very quickly.

Manual records

Paper forms, spreadsheets, and manual payment records take time and are harder to maintain.

Hard to track

Staff may need to check dates and payments manually to know whether a membership is active or expired.

Growing workload

As the number of members increases, duplicate records, incorrect information, and slow check-ins become bigger problems.

Our idea: put the important gym information in one centralized system.

What are we trying to achieve?

Our goal is not to build everything — it is to solve the most useful day-to-day problems.

MAIN GOAL

Build a web-based gym membership system that manages members, memberships, payments, attendance, and basic reports.

KEY OBJECTIVES

- Secure login with four roles: Admin, Staff, Trainer, and Member.
- Use OOP with Model, Repository, Service, and Route layers.
- Manage members and membership plans with CRUD functions.
- Track payments, expiry dates, renewals, attendance, and simple reports.

Scope and Limitation

Scope

- Member registration & profile management
- Membership package management (daily/monthly/quarterly/yearly)
- Membership enrollment & expiry-aware renewal
- Payment recording with admin verification
- Attendance check-in & history
- Role-based access (Admin, Member)
- Admin dashboard: search, filter, reports

Limitation

- Live payment-gateway settlement
- NFC/biometric hardware integration
- Multi-branch/multi-gym support
- Native mobile app
- Trainer-led class scheduling (stretch goal only)

Who will use the system?

Each role sees the features that match their responsibility.

ADMIN

Controls users, plans, payments, reports, and overall system information.

STAFF

Registers members, manages memberships, records payments, and handles attendance.

TRAINER

Views assigned members and manages basic training session information.

MEMBER

Checks their own profile, membership, payments, and attendance history.

Gym Owner → uses reports to monitor memberships, revenue, and overall performance.

What can the system do?

The main features focus on the tasks gym staff do most often.

Authentication

Login, logout, password hashing, sessions, and role-based access.

Member Management

Create, view, update, and delete member profiles.

Memberships

Create plans, enroll members, renew memberships, and track expiry.

Payments

Record payment amount, method, date, and verification status.

Attendance

Record check-ins and allow check-in only for active memberships.

Reports

Show daily revenue, active memberships, and expiring/expired members.

The OOP idea behind the system

We separate responsibilities so the project stays easier to understand and maintain.

MODEL

Represents the main objects User, Membership, Payment...

→

REPOSITORY

Handles database operations with MySQL.

→

SERVICE

Handles business rules and validation.

→

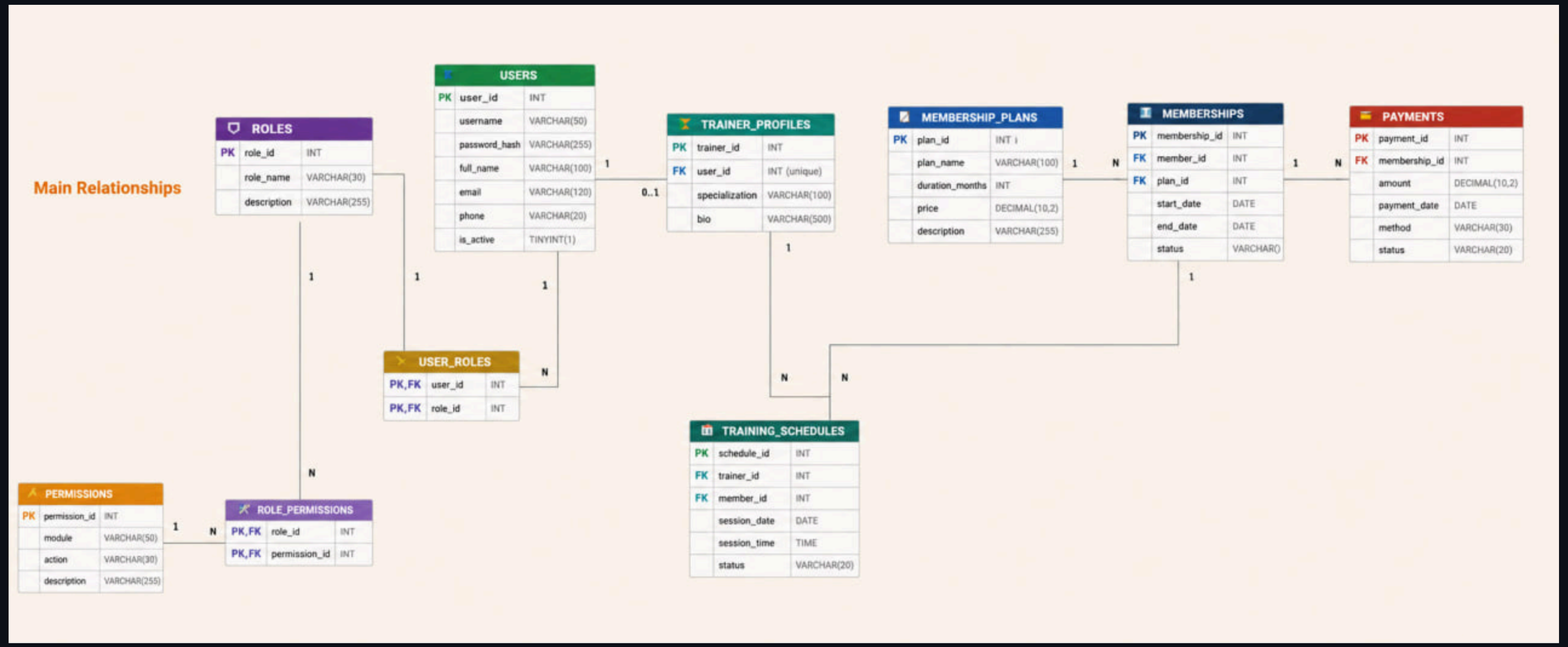
ROUTE

Receives requests and connects the web UI to services.

OOP principles used: Encapsulation • Inheritance • Polymorphism • Abstraction

Example: Admin, Staff, Trainer, and Member share common features through the User class.

Database Plan



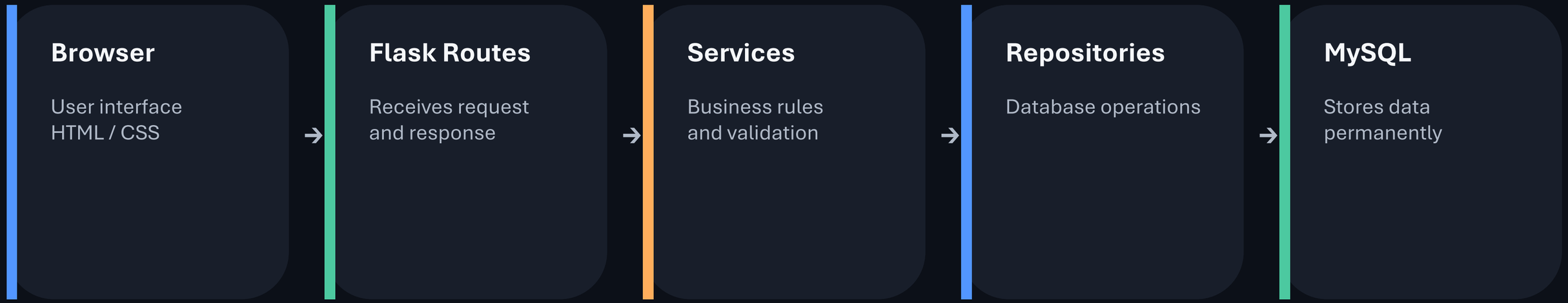
Draft entity-relationship diagram (ERD) for the Gym membership sytem database.

Main Table

Table	Main Columns	Purpose
roles	role_id (PK), role_name	Stores ADMIN, STAFF, TRAINER, and MEMBER.
users	user_id (PK), full_name, email, password_hash, role_id (FK), status	Stores user accounts. Email is unique.
membership_plans	plan_id (PK), plan_name, duration_months, price	Stores gym membership plans.
memberships	membership_id (PK), member_id (FK), plan_id (FK), start_date, end_date, status	Connects a Member to a MembershipPlan.
payments	payment_id (PK), membership_id (FK), amount, payment_date, method, status	Stores membership payments.
attendance	attendance_id (PK), member_id (FK), check_in_time	Stores member check-ins.
training_schedules	schedule_id (PK), trainer_id (FK), member_id (FK), session_date, session_time, status	Stores training sessions.

How does the system work?

A request moves through clear0layers before reaching the database.



Browser → Flask Routes → Services → Repositories → MySQL

This structure keeps the user interface, business logic, and database work from becoming one big block of code.

Database: keeping the records connected

The database is designed around the main things a gym needs to remember.

USERS

Accounts + roles

MEMBERSHIP PLANS

Plan name, duration, price

MEMBERSHIPS

Member enrollment + dates + status

PAYMENTS

Amount, date, method, status

ATTENDANCE

Member check-in history

TRAINING SCHEDULES

Trainer + member + session

Example relationship: one member can have many memberships and many attendance records.

How will we build it?

We will work in short sprints, test as we go, and keep the scope realistic.

WEEK 1 - Requirements, research, OOP design, architecture, ERD, database schema, and project setup.

WEEK 2 - Users, roles, authentication, authorization, and member profiles.

WEEK 3 - Membership plans, memberships, expiry status, and renewal.

WEEK 4 - Payments, attendance, training sessions, search, filtering, and reports.

WEEKS 5–6 - Integration, testing, bug fixing, security checks, documentation, screenshots, and final demo.

Testing happens during development — not only at the end.

What do we expect at the end?

A working prototype that makes common gym tasks easier and more organized.

- A working Gym Membership Management web application using Python, Flask, and MySQL.
- Separate access for Admin, Staff, Trainer, and Member.
- Member and membership-plan CRUD, payment recording, renewal, and attendance.
- Search, filtering, and simple reports for membership status and daily revenue.
- A tested four-layer OOP structure with documentation and demo evidence.

THE BIG PICTURE

Less manual work
+ organized records
+ faster membership checking
+ clearer gym information

UNIVERSITY
of
TECHNOLOGY & ENTREPRENEURSHIP

THANK YOU!