JEDI TRAINING PRESENTATION

TEAM DELTA

FlipFit Gym Application

THE PROBLEM STATMENT

The goal of the FlipFit project is to design and develop a robust, enterprise-grade application that facilitates Flipkart's entry into the fitness industry through partnerships with gym centers across Bangalore. The system aims to provide a seamless digital platform where users can register, browse available gym centers, and book specific hour-long workout slots based on real-time availability. Key technical objectives include implementing a scalable microservices architecture, ensuring high concurrency control to prevent overbooking, and providing an intuitive UI for managing workout schedules.

OUR TEAM

1. GRACY GULATI
2. ADITYA S RAY
3. BRIJ BIDHIN DESAI
4. DHRUV BHANDARI
5. PRIYA GUPTA
6. ESHAA ARANGGAN
7. JYOTSNA SHRIVASTAVA
8. MAHIMA N R
9. POTHURI AVANEESH
10. SAMRAVI SIVA

OUR JOURNEY

DAILY PLANS AND DELIVERABLES

DAY 1

Initiated the system design phase by drafting Activity and Class diagrams to map out the booking logic and the relationships between users, gym centers, and time slots.

DAY 2

Established the project skeleton by implementing a modular architecture with dedicated packages for Bean (POJO) entities, Business logic, and Client interfaces to ensure clean separation of concerns.

DAY 3

Developed the interactive user menu and finalized the implementation of all core functions, including gym registration, slot availability viewing, and the logic to prevent overlapping bookings.

DAY 4

Integrated the application with a relational database by creating the Data Access Object (DAO) layer, moving away from in-memory storage to ensure persistent data for centers and user bookings.

DAY 5

Enhanced system stability by implementing robust exception handling (e.g., for overbooking or unauthorized access) and wrapped the microservice into a Dropwizard application for deployment and monitoring.

TECH STACK

1 BACKEND

2 DATA

3 TOOLS

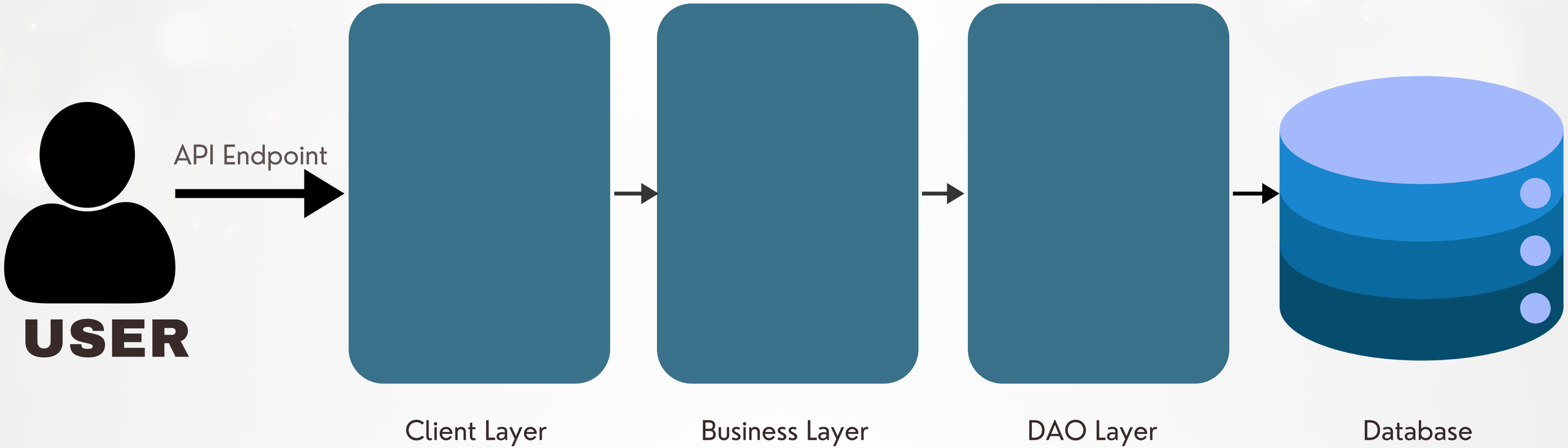
FEATURES

- Admin Oversight: Includes an administrative role responsible for approving gym owners and managing the initial entry of center and slot data.
- Multi-Gym Ownership: Enables registered gym owners to manage and list multiple fitness centers across the city.
- Customer Accessibility: Provides a unified platform where customers can browse and book slots at any approved gym center.
- Automated Waitlist: Once a slot reaches maximum capacity, subsequent users are placed on a waiting list.
- Smart Notifications: Automatically alerts waitlisted candidates via a notification system when they have been promoted to an active booking.

PROJECT STRUCTURE

- Beans Layer: Developed specialized Data Transfer Objects (DTOs) utilizing setters and getters to ensure standardized data encapsulation across the application.
- Client Layer: Created a robust command-line interface that serves as the primary entry point, managing frontend navigation and user interaction flow for gym discovery and booking.
- Business Layer: Engineered the core service logic to manage complex requirements such as seat availability checks, booking conflict resolution, and gym center management.
- DAO: Established a dedicated layer to manage MySQL database interactions, ensuring persistent storage for center details and booking records as required by the project guidelines.
- Exception Handling: Designed and implemented custom exception classes to provide clear feedback for scenarios like invalid credentials, overbooked slots, or missing data.
- Constants & SQL Schema: Defined a structured SQL schema for relational data integrity and maintained a centralized Constants file for SQL queries to promote code reusability and maintainability.

PROJECT STRUCTURE



CHALLENGES FACED

- Navigating how to use new software.
- Collaborating and coordinating with the rest of the team while building the application.
- Managing a shared codebase for a multi-layered application (DAO, Business, Client) required strict coordination to resolve merge conflicts and maintain the Git Commits SDLC standards.
- Coordinating between team members working on the UI/Client layer and those on the Backend/DAO layer to ensure that data structures matched during integration
- Transitioning from standard Java applications to a Dropwizard microservices architecture, which required understanding integrated Jetty servers and managed environments.

LEARNINGS

- Team Collaboration and Effective Communication.
- Tech Stack: GitHub, Springboot, Java, MySQL.
- Learned to use UML diagrams to map out the interactions between gym centers, hour-long time slots, and user bookings before writing a single line of code.
- Gained experience in translating high-level problem statements into a functioning system by designing both a user interface and a backend system for a real-world enterprise application.
- Learned to build a "defensive" application by implementing custom exceptions to handle invalid inputs or unauthorized operations, improving the overall user experience.

DEMO

Q&A

THANK YOU