FLIPFIT:FLIPKART JEDI PROGRAM 2025

A NEW ENTERPRISE FITNESS APPLICATION BY FLIPKART

TEAM 03-GL(PAWAN AGRAWAL)

WHO ARE WE?

PAWAN

SANSKAR

KUSH

DIVIJA

AJITESH

TEAM C

AGENDA OVERVIEW

01
PROBLEM STATEMENT

02
FRAMEWORK FOR SIX DAYS

03
ARCHITECTURES

04
JOURNEY OF FLIPFIT

05
TECH STACK

06
CHALLENGE AND LEARNINGS

07
ENGINEERING PRACTICECS

08
DEMO AND CODE REVIEW

PROBLEM STATEMENT

Design a backend system for a new enterprise application that Flipkart is launching, FlipFit.
Prepare a proper backend application for the beta launch.

01

INEFFICIENT SLOT MANAGEMENT

Manual slot management leads to errors, double bookings, and underutilized slots due to a lack of real-time updates.

02

SCALABILITY & SYSTEM RELIABILITY

Expanding FlipFit requires efficient handling of multiple centers, bookings, waitlists, and cancellations.

03

CUSTOMER EXPERIENCE CHALLENGES

Unclear slot availability, booking conflicts, and missing real-time notifications frustrate users. Automated waitlist promotions and nearest-slot suggestions enhance satisfaction.

FRAMEWORK FOR SIX DAYS

1.EVERYDAY DISCUSSION ABOUT JAVA CONCEPTS,TECHNOLOGIES AND DOUBT CLEARANCE

2.PROJECT DISCUSSION AND DIVISION OF PROJECT INTO SMALL PARTS

3.LEARN CONCEPTS RELATED TO GIT,GITHUB,MYSQL ETC

4.INSTALLATION OF GIT,GITHUB,MYSQL,MYSQLWORKBENCH,ETC

5.CREATED UML DIAGRAMS,GIT COLLABORATION,JDBC CONNECTION AND CODE STRUCTURING AND LOOSE COUPLING

6. EXPLORED JAVA 17 FEATURES LIKE STREAM API , DOCUMENTATION.

STAKEHOLDERS

TRAINERS

- MR AMIT BALYAN

SPONSORS

- FLIPKART

COORDINATOR

- RAKSHA DUBEY

TIMELINE

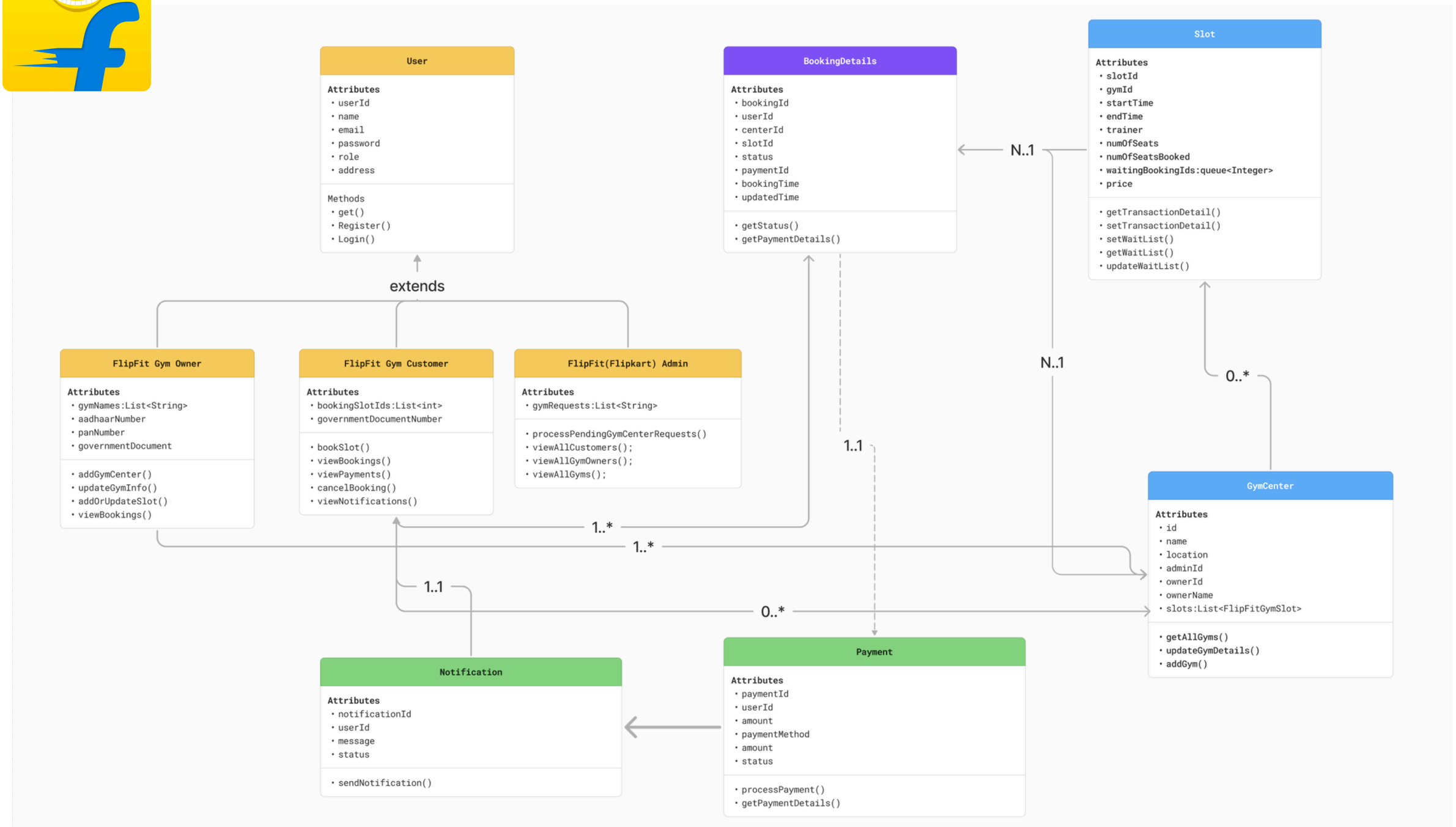
AND... LIMITLESS KNOWLEDGE

FEATURES IMPLEMENTED

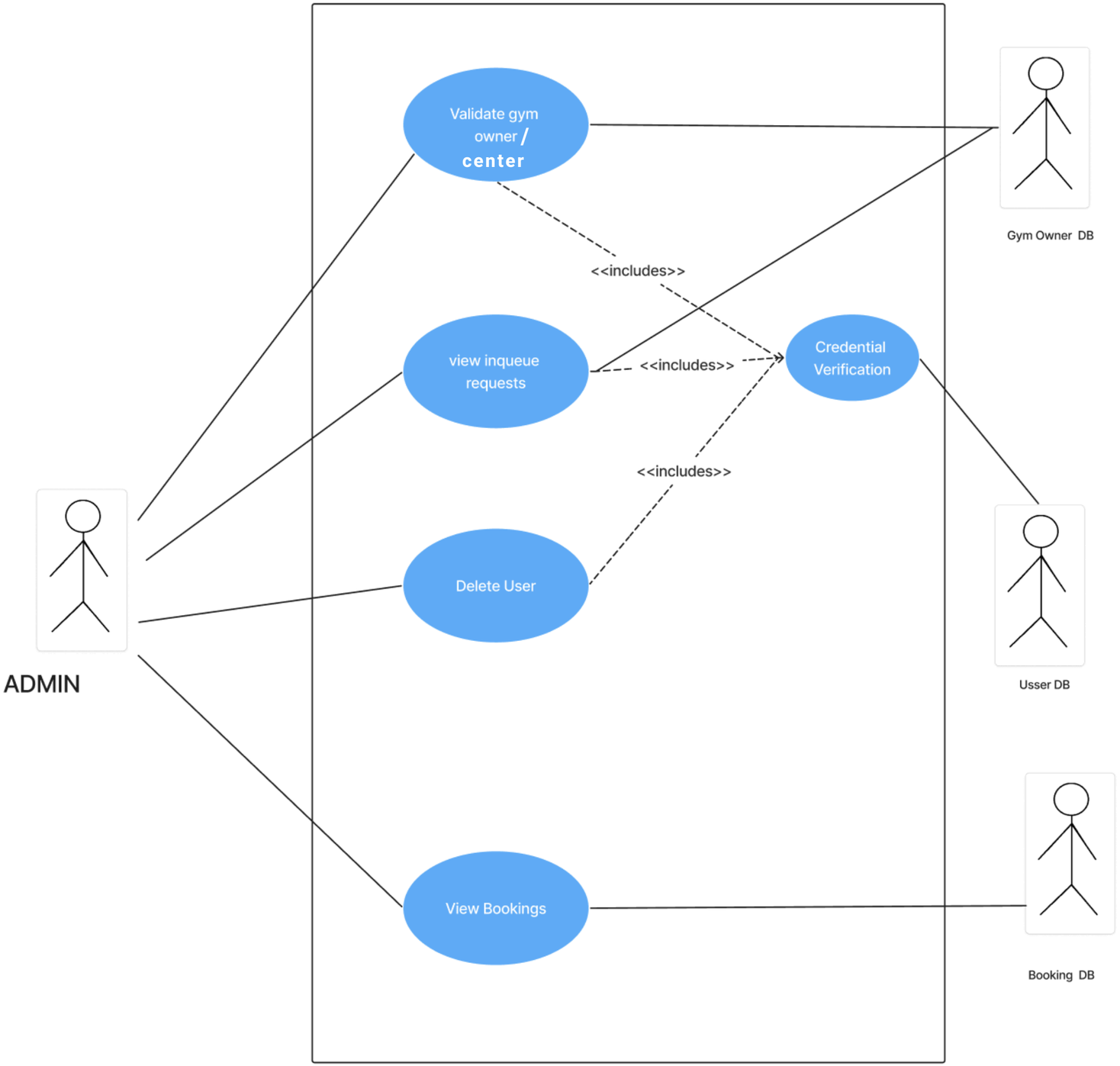
- VIEW CENTERS FOR A CITY
- BOOK A SLOT
- CANCEL PREVIOUS BOOKING
- BOOKING ERROR FOR ALREADY BOOKED SLOT
- PREVENT OVERBOOKING
- VIEW PLAN BY DAY
- GYM CENTER VERIFICATION
- ADD TO WAITLIST
- PAYMENT
- NOTIFICATION
- EDIT/UPDATE SLOT AND GYM INFORMATION

ARCHITECTURES

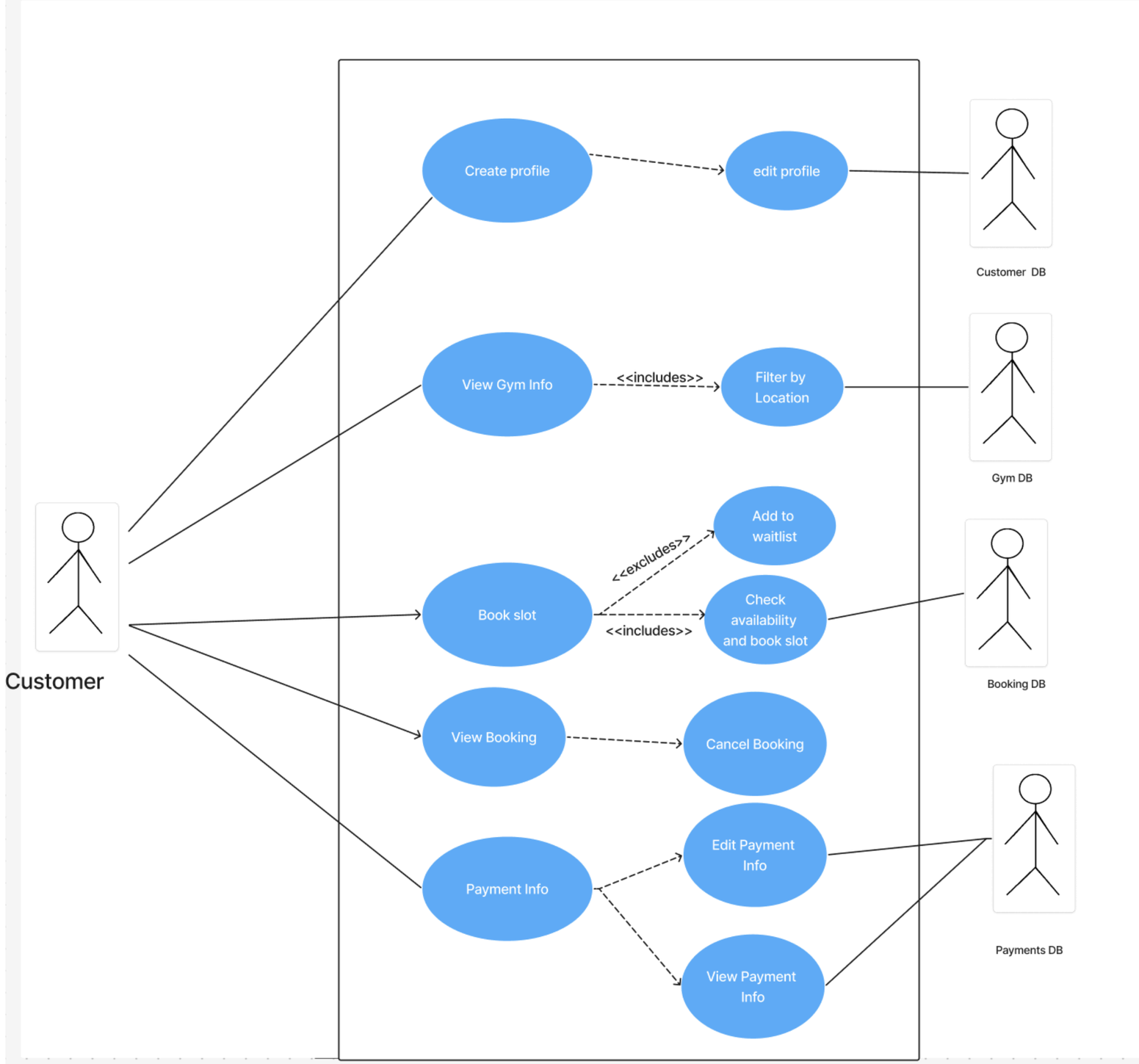
CLASS DIAGRAM



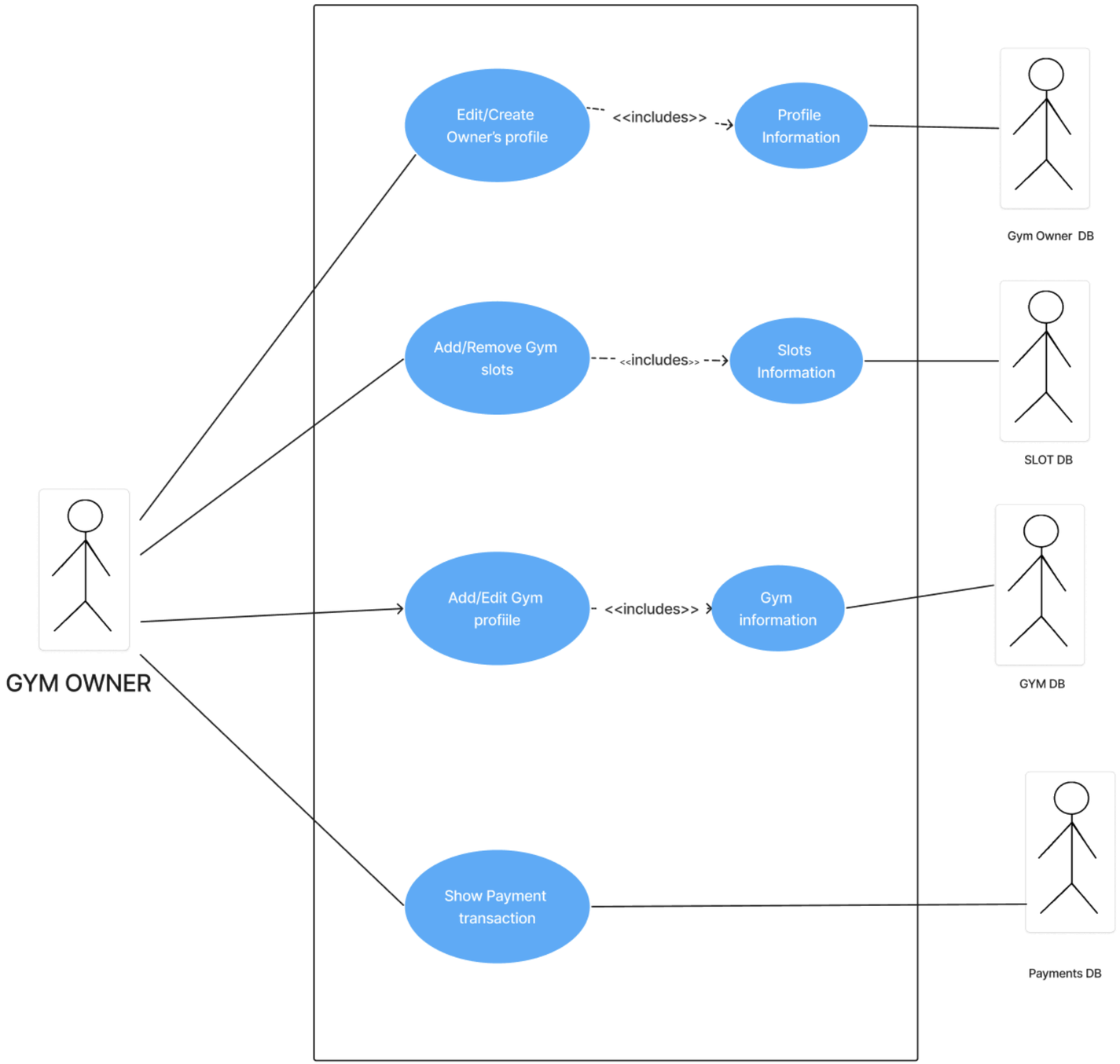
ADMIN USE CASE DIAGRAM



CUSTOMER USE CASE DIAGRAM



GYM OWNER USE CASE DIAGRAM



THE JOURNEY

The FlipFit platform streamlines gym management by enhancing user experience, optimizing booking systems, and improving operational efficiency. With a structured approach to user roles, gym management, and slot reservations, FlipFit ensures seamless interactions between customers, gym owners, and administrators.

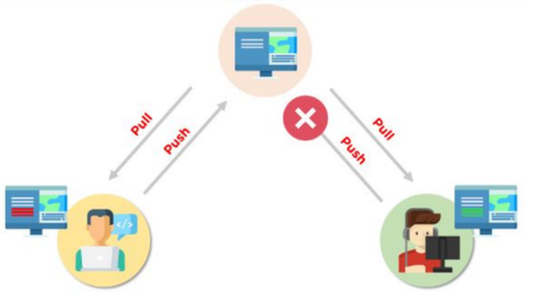
DEVELOPMENT OF FLIPFIT

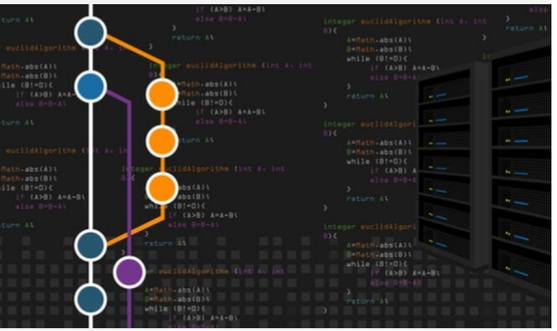
Through in-depth analysis of user needs, business logic, and system architecture, FlipFit overcomes key challenges to deliver an efficient and scalable fitness booking solution.

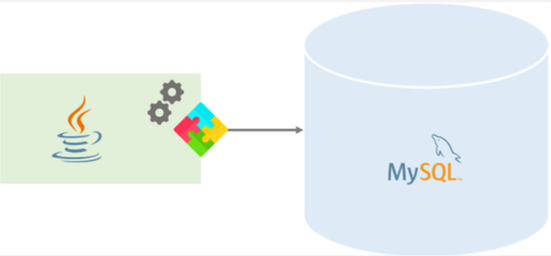
UML Diagrams Created UML diagrams, including class, use case, and sequence diagrams. These helped structure the system, improving clarity and scalability.	**Project Structure** Organized the project into client, business, DAO, and utility packages. This modular approach ensures maintainability and future scalability.
Database & JDBC Designed the FlipFit database schema and connected it via JDBC. Implemented CRUD operations with secure queries and transactions.	**Interfaces & Implementations** Developed interfaces and implementations for role-based functionalities. Added error handling, security checks, and validation for reliability.

TECH STACK

Spring Tools







CHALLENGES AND LEARNINGS

MAINTAINING CODE CONSISTENCY

Changing code across multiple files was difficult, but following a modular design improved maintainability.

COLLABORATION & GIT CONFLICTS

Merging changes in a team of five was intially challenging but during the project we learned about more git commands and it helped us keep a structural git flow.

CONSISTENT FILE AND VARIABLE NAMING

With five people on the team working on the same project, it was somewhat difficult to be consistent with all the variable namings, but following the UML diagrams and proper conventions helped us in being consistent.

DESIGNING PROPER UML DIAGRAMS

Proper UML diagram is a very important part of a functioning application, and learning how to make proper, consistent and concise yet precise UML diagrams were part of our learning

SQL CONNECTIONS AND TABLE STRUCTURE

We learned how to use Java Database Connectivity for connecting our project with the MySQL database.

ENGINEERING PRACTICES

LOOSE COUPLING

- Ensure minimal dependencies between services, enabling independent scaling, flexibility, and easier maintenance.

MICROSERVICES ARCHITECTURE

- Design the system using microservices to manage gym centers, bookings, and user authentication separately, improving modularity and fault tolerance.

TRANSACTION MANAGEMENT

- Use ACID-compliant database transactions to ensure consistency in bookings, preventing overbooking, race conditions, and data inconsistencies.

SQL TABLE MANAGEMENT

- Optimize database design with proper indexing, normalization, and partitioning to handle high read/write operations efficiently and maintain data integrity.

DEMO AND CODE REVIEW

THANK YOU