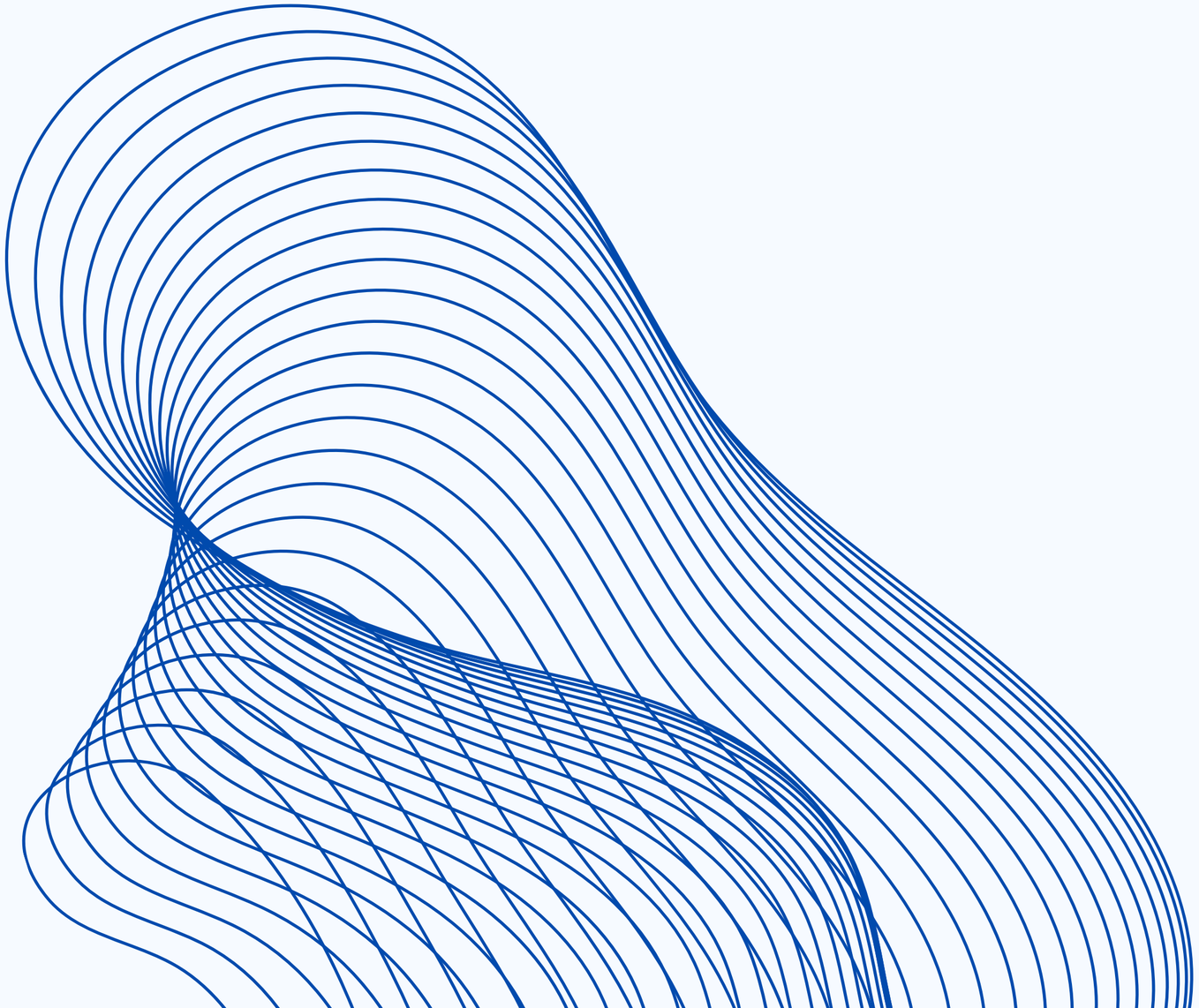


INTERNSHIP PLACEMENT MANAGEMENT SYSTEM

SCED Group-6
ARCHISHA DEB BARMA
CHARLES LEE JUN NGAI
CHU **YI XIN**
HARITH BIN FA'IZAL
MAHALIK **AMISHA** ALAMA
WAI YAN MIN KO KO

01

ESSENTIAL APPLICATION INFORMATION

Student

1. View Internships
2. Apply Internships
3. Accept Internships
4. Withdraw Internships

Company Representative

1. Create Opportunities
2. Manage Opportunities
3. Manage Applications
4. Toggle visibility

Career Center Staff

1. Approve Accounts
2. Approve Opportunities
3. Manage Withdrawals
4. Generate Report

02

TECH STACK

Backend

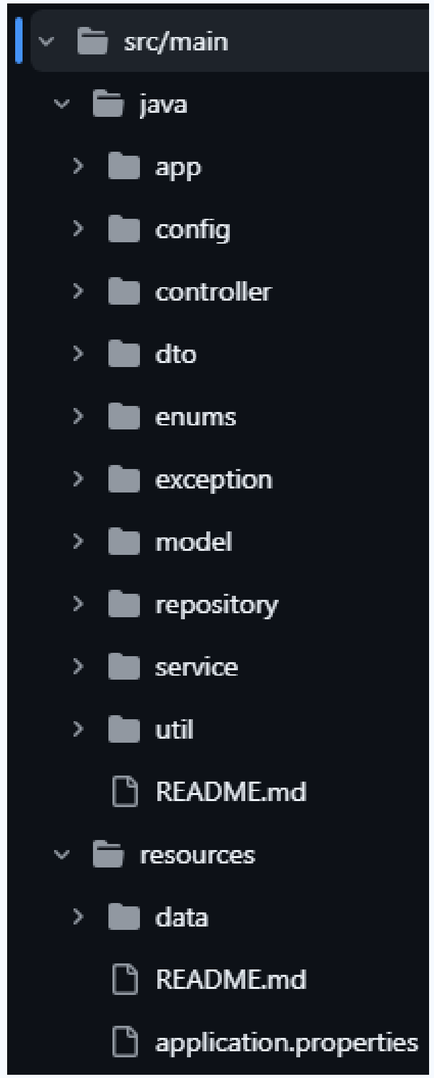
- Java 21 (OpenJDK)
- Spring Boot
- Maven
- OpenCSV

Frontend

- React
- Vite
- Material UI

Architecture

- RESTful API
- Layered Architechture
- CSV-based Data Persistance

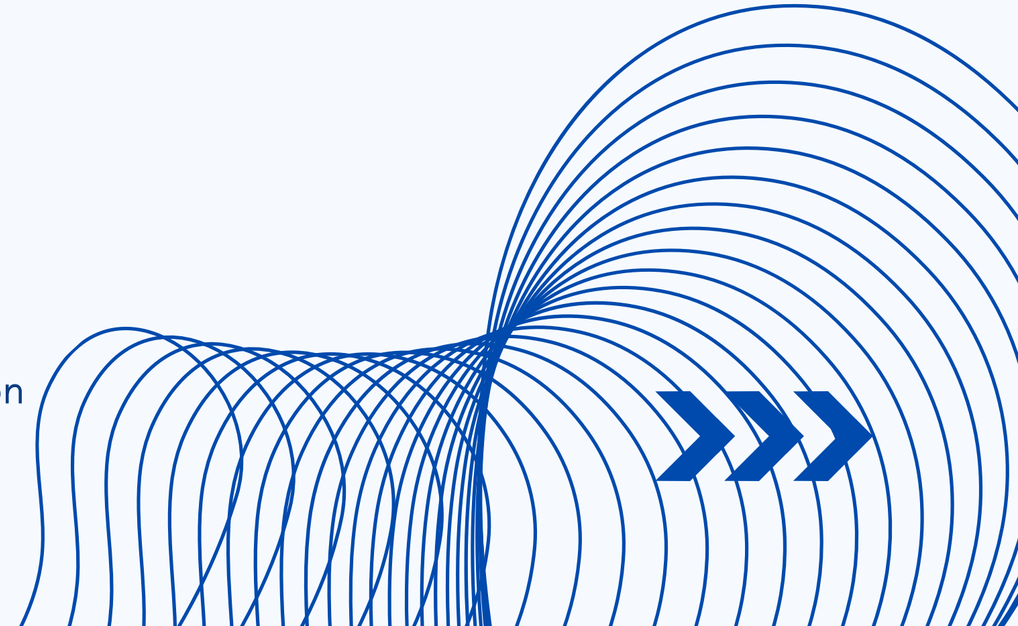


src/main/
java/
- app/InternshipPlacementApplication.java
- config/ Spring configuration
- controller/ REST API controllers
- dto/ Data Transfer Objects
- enums/ Enumerations
- exception/ Custom exceptions
- model/ Domain models
- repository/ Repository interfaces
  - impl/ CSV repository implementations
- service/ Service interfaces
  - impl/ Service implementations
  - indiv_contribution/ Individual contributions
- util/ Utility classes

resources/
- application.properties Spring Boot configuration
- data/ CSV data files

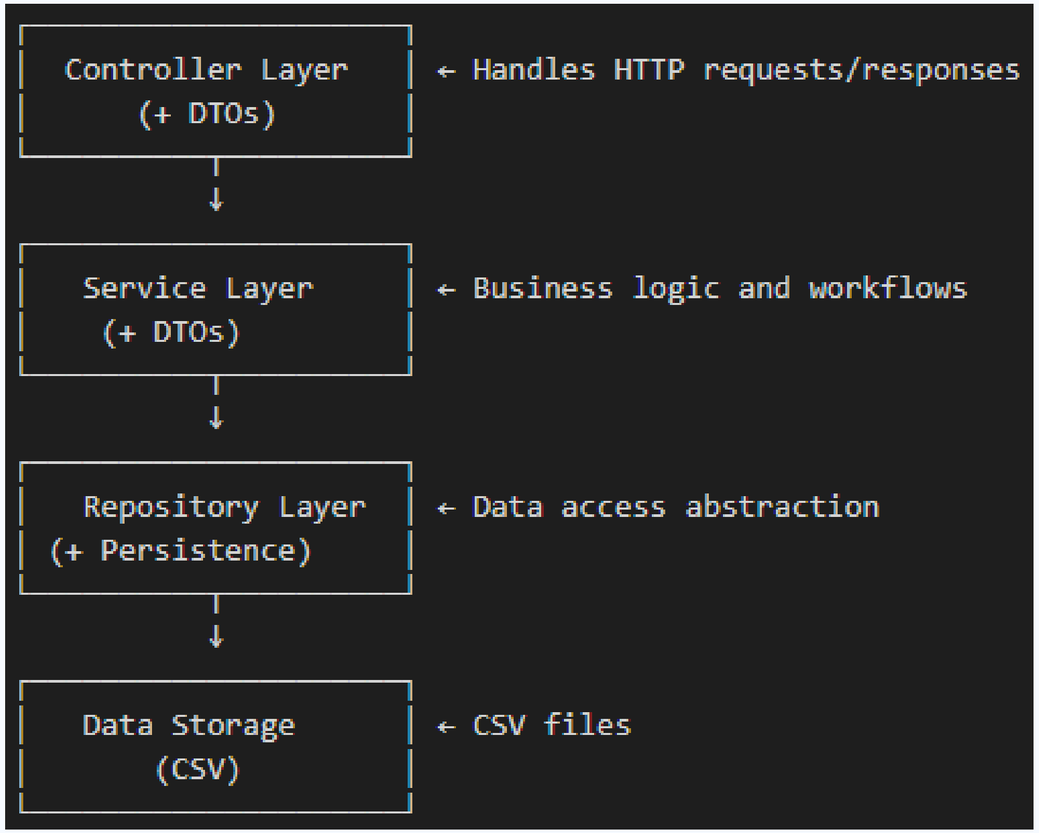
04

JAVA BACKEND FOLDER STRUCTURE



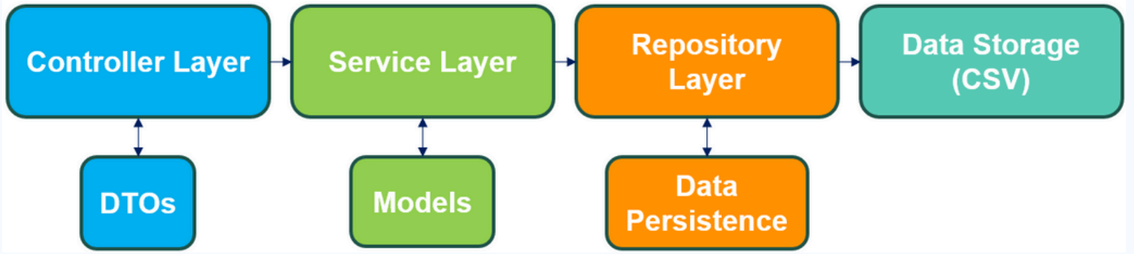
LAYERED ARCHITECTURE

05



1. **Distinct Responsibility**: easier to understand, maintain, and test the codebase

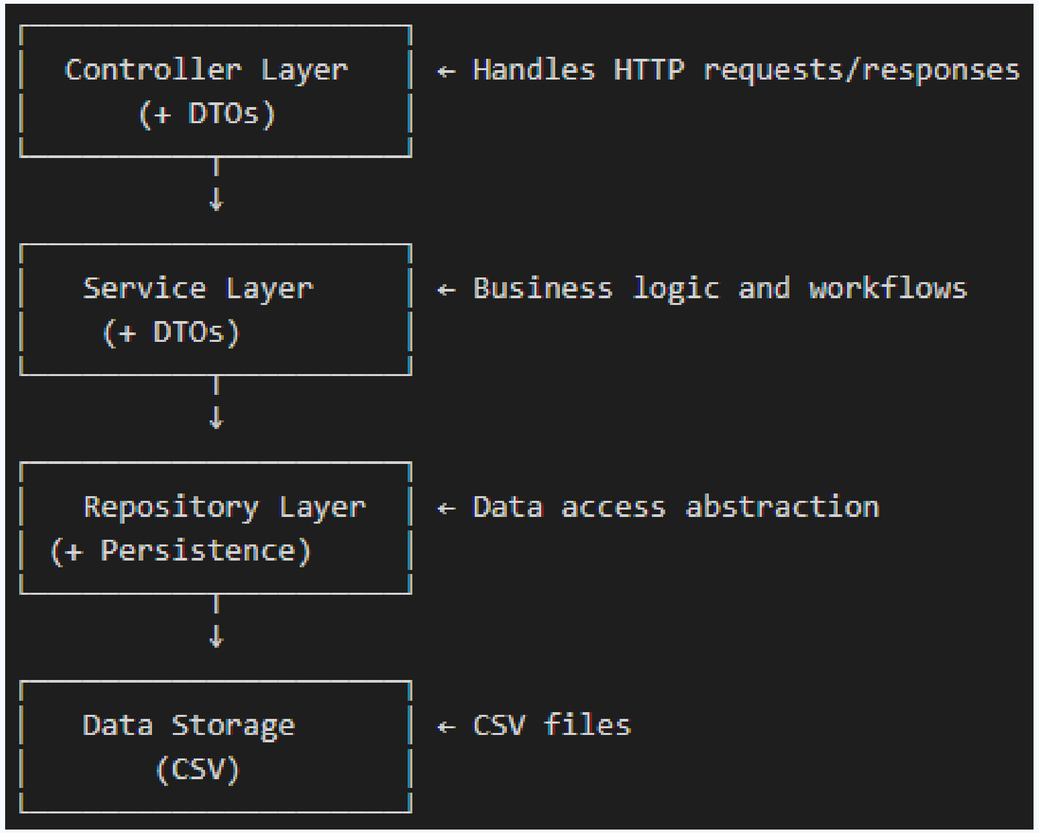
2. Follows a **Layered Architecture** pattern:



3. **Object-Oriented Design Principles**:
   - **Encapsulation**: Business logic contained within appropriate classes and packages
   - **Abstraction**: Interfaces define contracts (Repository, Service)
   - **Inheritance**: User hierarchy
   - **Polymorphism**: Different implementations of repository interfaces

LAYERED ARCHITECTURE

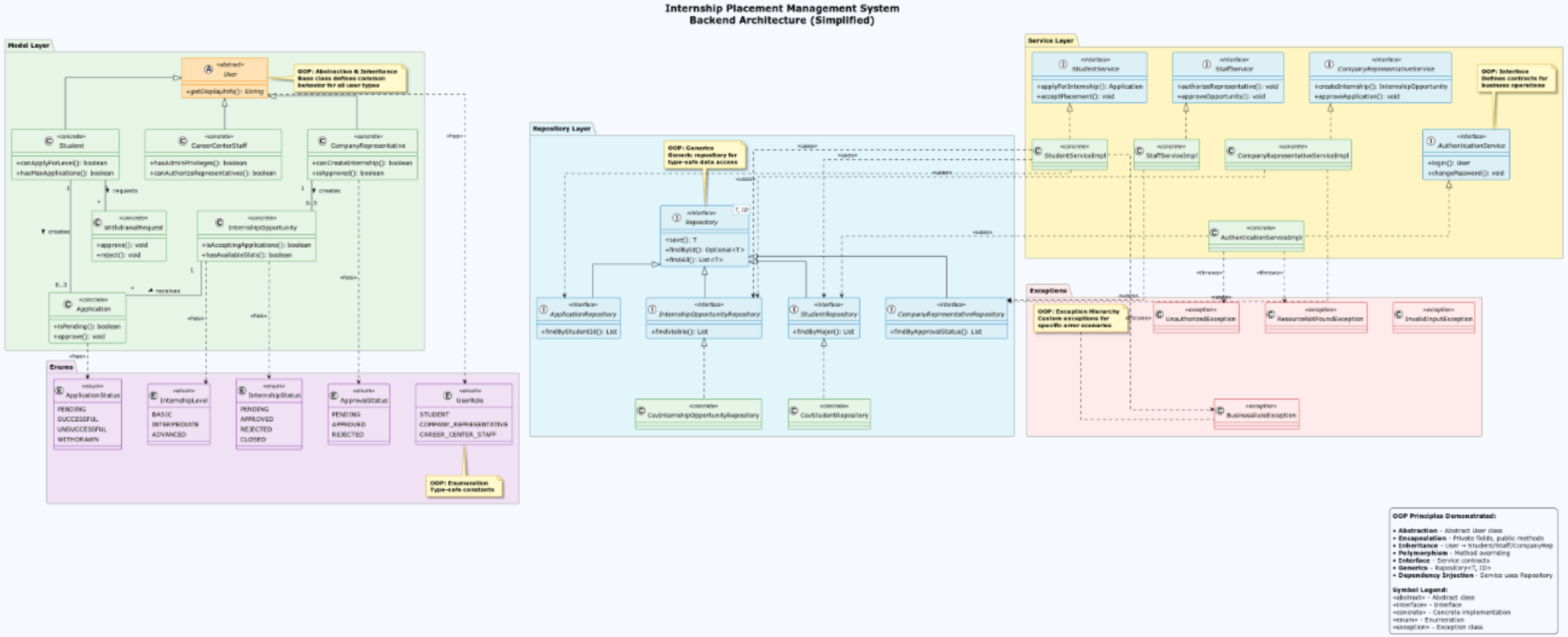
05



4. **Maintainability**: Clear dependencies between packages

5. **Scalability**: Easy to add new features

6. **Industry Standards**: Follows common Java/Spring Boot conventions



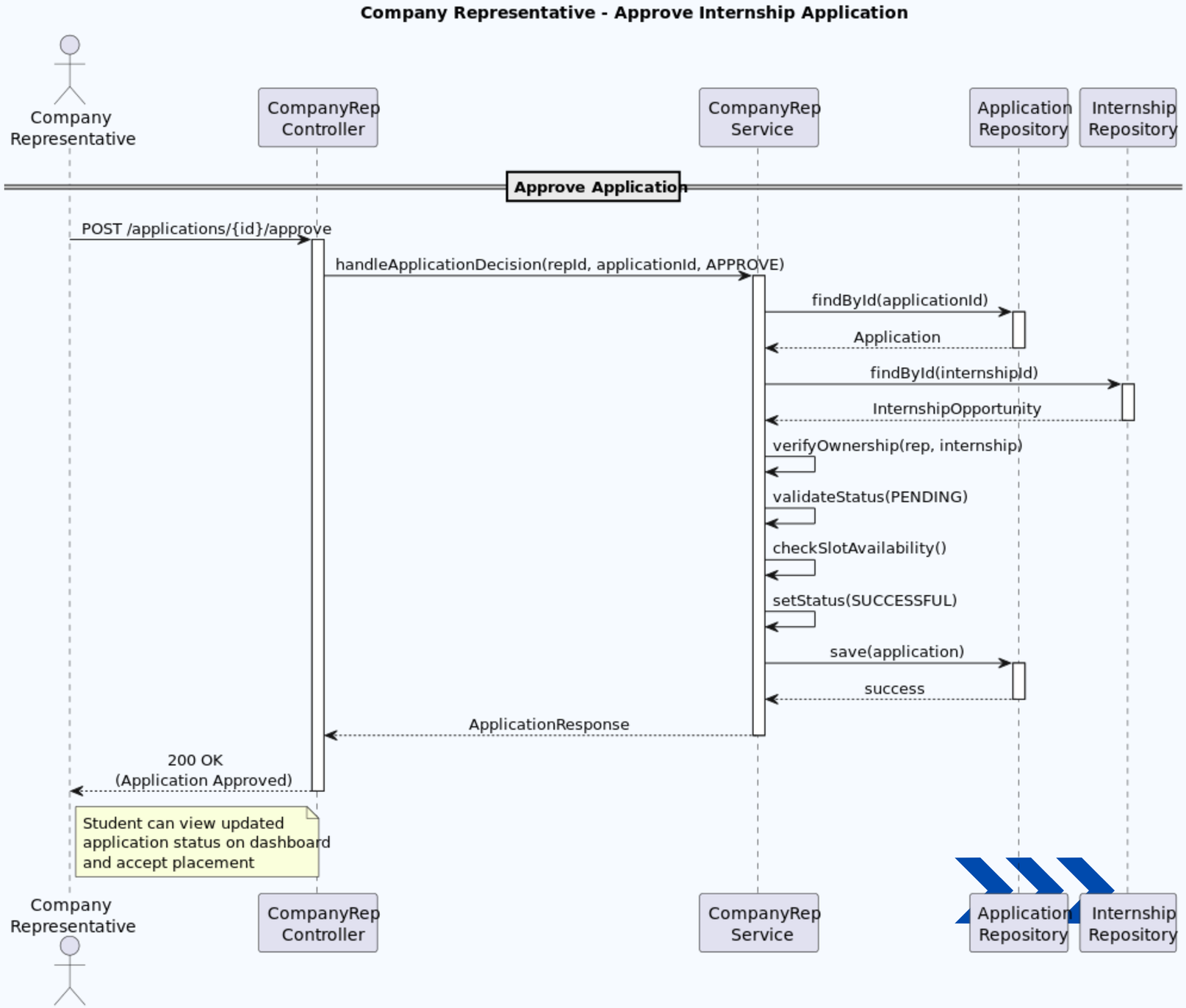
06

UML CLASS DIAGRAM (SIMPLIFIED)

07

UML SEQUENCE DIAGRAM

COMPANY REPRESENTATIVE



08

SAMPLE TEST CASES

Users

1. Student: U2310001A/Tan Wei Ling, Year 2, Computer Science
2. Company Rep: C73422/Alice Tan (Company: SAP), HR Manager
3. Coordinator: tan002/Mr. Tan Boon Kiat (Department: CCDS)

Opportunities

1. INT001 @ SAP — Software Engineering Intern, Level: BASIC, PreferredMajor: Computer Science, Slots: 3, Visible: Yes, Status: APPROVED

08

SAMPLE TEST CASES

Valid Login (Student) - Archisha

Pre: Student U2310001A exists in the system.
Steps: Login as U2310001A with correct password.
Expected: Login success; menu shows Student actions only.

Create new Internship opportunity - Amisha

Pre: Company Rep (Alice Tan) is logged in, Created fewer than 5 opportunities
Steps: Create AI Solutions Architect, Fill in Description, Level, PreferredMajor, Slots, Opening Date, Closing Date, Start Date, End Date
Expected: Opportunity saved with status PENDING, Opportunity is NOT visible to students until approved by coordinator

Apply Internship when eligible - Harith

Pre: Student U2310001A, Year 2, Computer Science, 0 current applications, has not accepted any placement
Software Engineering Intern: INT001, APPROVED, BASIC, Computer Science, Visible: Yes
Steps: Internships → Apply to Software Engineering Intern (INT001)
Expected: Application created with status "PENDING", Application appears in "My Applications" list. Student's active application count increases to 1/3

Generate placement report - Archisha

Pre: Career Support Staff (Tan Boon Kiat) is logged in.
Steps: Reports → Select Filters → Generate Report
Expected: Report produced on the frontend – Downloadable PDF

LIVE DEMO

THANK YOU