Case Study: Library Management System Day 11

Objective: Design a Library Management System where:

Reader.java

```
package com.example.library.entity;
import jakarta.persistence.*;
import java.util.List;
@Entity
public class Reader {
   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   private Long id;
   private String name;
   private String email;
   @OneToMany(mappedBy = "reader")
   private List<Book> books;
   public Reader() {}
   public Reader(Long id, String name, String email, List<Book> books) {
      this.id = id;
      this.name  =  name;
      this.email  =  email;
      this.books = books;
   }
   public Long getId() { return id; }
   public void setId(Long id) { this.id = id; }
   public String getName() { return name; }
   public void setName(String name) { this.name = name; }
```

```
    public String getEmail() { return email; }
    public void setEmail(String email) { this.email = email; }
    public List<Book> getBooks() { return books; }
    public void setBooks(List<Book> books) { this.books = books; }
}
```

Author.java

```
package com.example.library.entity;
import jakarta.persistence.*;
import java.util.List;
@Entity
public class Author {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    @OneToMany(mappedBy = "author")
    private List<Book> books;
    public Author() {}
    public Author(Long id, String name, List<Book> books) {
        this.id = id;
        this.name = name;
        this.books = books;
    }
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public List<Book> getBooks() { return books; }
    public void setBooks(List<Book> books) { this.books = books; }
```

```
}
```

Category.java

```
package com.example.library.entity;
import jakarta.persistence.*;
import java.util.List;
@Entity
public class Category {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    @OneToMany(mappedBy = "category")
    private List<Book> books;
    public Category() {}
    public Category(Long id, String name, List<Book> books) {
        this.id = id;
        this.name = name;
        this.books = books;
    }
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public List<Book> getBooks() { return books; }
    public void setBooks(List<Book> books) { this.books = books; }
}
```

Book.java

```
package com.example.library.entity;
import jakarta.persistence.*;
```

```
import java.time.LocalDate;

@Entity
public class Book {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String title;
    private LocalDate publishDate;
    @ManyToOne
    @JoinColumn(name = "reader_id")
    private Reader reader;
    @ManyToOne
    @JoinColumn(name = "author_id")
    private Author author;
    @ManyToOne
    @JoinColumn(name = "category_id")
    private Category category;
    public Book() {}
    public Book(Long id, String title, LocalDate publishDate, Reader reader, Author author, Category
category) {
        this.id = id;
        this.title = title;
        this.publishDate = publishDate;
        this.reader = reader;
        this.author = author;
        this.category = category;
    }
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }


    public String getTitle() { return title; }
```

```
    public void setTitle(String title) { this.title = title; }
    public LocalDate getPublishDate() { return publishDate; }
    public void setPublishDate(LocalDate publishDate) { this.publishDate = publishDate; }
    public Reader getReader() { return reader; }
    public void setReader(Reader reader) { this.reader = reader; }
    public Author getAuthor() { return author; }
    public void setAuthor(Author author) { this.author = author; }
    public Category getCategory() { return category; }
    public void setCategory(Category category) { this.category = category; }
}
```

Repository Interfaces — com.example.library.repository

```
package com.example.library.repository;
import com.example.library.entity.*;
import org.springframework.data.jpa.repository.JpaRepository;
public interface ReaderRepository extends JpaRepository<Reader, Long> {}
public interface AuthorRepository extends JpaRepository<Author, Long> {}
public interface CategoryRepository extends JpaRepository<Category, Long> {}
public interface BookRepository extends JpaRepository<Book, Long> {}
```

LibraryController.java

```
package com.example.library.controller;
import com.example.library.entity.*;
import com.example.library.repository.*;
import org.springframework.web.bind.annotation.*;
import java.util.List;
@RestController
@RequestMapping("/api")
public class LibraryController {
    private final ReaderRepository readerRepo;
    private final BookRepository bookRepo;
```

```
    private final AuthorRepository authorRepo;
    private final CategoryRepository categoryRepo;
    public LibraryController(ReaderRepository readerRepo, BookRepository bookRepo,
                             AuthorRepository authorRepo, CategoryRepository categoryRepo) {
        this.readerRepo = readerRepo;
        this.bookRepo = bookRepo;
        this.authorRepo = authorRepo;
        this.categoryRepo = categoryRepo;
    }
    @PostMapping("/readers")
    public Reader addReader(@RequestBody Reader reader) {
        return readerRepo.save(reader);
    }
    @PostMapping("/authors")
    public Author addAuthor(@RequestBody Author author) {
        return authorRepo.save(author);
    }
    @PostMapping("/categories")
    public Category addCategory(@RequestBody Category category) {
        return categoryRepo.save(category);
    }
    @PostMapping("/books")
    public Book addBook(@RequestBody Book book) {
        return bookRepo.save(book);
    }
    @GetMapping("/books")
    public List<Book> getAllBooks() {
        return bookRepo.findAll();
    }
}
```

Case Study Title: Hospital Management System using Spring Boot and Spring Data JPA

Patient.java

```
package com.example.hospital.entity;
import jakarta.persistence.*;
import java.util.List;
@Entity
public class Patient {
   @Id
   @GeneratedValue(strategy = GenerationType.IDENTITY)
   private Long id;
   private String name;
   private int age;
   private String gender;
   private String address;
   @OneToMany(mappedBy = "patient", cascade = CascadeType.ALL)
   private List<Appointment> appointments;
   @OneToMany(mappedBy = "patient", cascade = CascadeType.ALL)
   private List<MedicalRecord> records;
   public Patient() {}
   public Patient(Long id, String name, int age, String gender, String address,
            List<Appointment> appointments, List<MedicalRecord> records) {
      this.id = id;
      this.name = name;
      this.age = age;
      this.gender = gender;
      this.address = address;
      this.appointments = appointments;
      this.records = records;
   }
   public Long getId() { return id; }
   public void setId(Long id) { this.id = id; }
   public String getName() { return name; }
```

```
    public void setName(String name) { this.name = name; }
    public int getAge() { return age; }
    public void setAge(int age) { this.age = age; }
    public String getGender() { return gender; }
    public void setGender(String gender) { this.gender = gender; }
    public String getAddress() { return address; }
    public void setAddress(String address) { this.address = address; }
    public List<Appointment> getAppointments() { return appointments; }
    public void setAppointments(List<Appointment> appointments) { this.appointments =
appointments; }
    public List<MedicalRecord> getRecords() { return records; }
    public void setRecords(List<MedicalRecord> records) { this.records = records; }
}
```

Doctor.java

```
package com.example.hospital.entity;
import jakarta.persistence.*;
import java.util.List;
@Entity
public class Doctor {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String name;
    private String specialization;
    private String email;
    private String phone;
    @OneToMany(mappedBy = "doctor", cascade = CascadeType.ALL)
    private List<Appointment> appointments;
    public Doctor() {}
    public Doctor(Long id, String name, String specialization, String email, String phone,
List<Appointment> appointments) {
        this.id = id;
        this.name = name;
```

```
        this.specialization = specialization;
        this.email = email;
        this.phone = phone;
        this.appointments = appointments;
    }
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }
    public String getSpecialization() { return specialization; }
    public void setSpecialization(String specialization) { this.specialization = specialization; }
    public String getEmail() { return email; }
    public void setEmail(String email) { this.email = email; }
    public String getPhone() { return phone; }
    public void setPhone(String phone) { this.phone = phone; }
    public List<Appointment> getAppointments() { return appointments; }
    public void setAppointments(List<Appointment> appointments) { this.appointments =
appointments; }
}
```

Appointment.java

```
package  com.example.hospital.entity;
import jakarta.persistence.*;
import java.time.LocalDate;
import  java.time.LocalTime;
@Entity
public class Appointment {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private LocalDate date;
    private LocalTime time;
    private String notes;
    @ManyToOne
```

```
    private Patient patient;
    @ManyToOne
    private Doctor doctor;
    public Appointment() {}
    public Appointment(Long id, LocalDate date, LocalTime time, String notes, Patient patient, Doctor
doctor) {
        this.id = id;
        this.date = date;
        this.time = time;
        this.notes = notes;
        this.patient = patient;
        this.doctor = doctor;
    }
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }
    public LocalDate getDate() { return date; }
    public void setDate(LocalDate date) { this.date = date; }
    public LocalTime getTime() { return time; }
    public void setTime(LocalTime time) { this.time = time; }
    public String getNotes() { return notes; }
    public void setNotes(String notes) { this.notes = notes; }
    public Patient getPatient() { return patient; }
    public void setPatient(Patient patient) { this.patient = patient; }
    public Doctor getDoctor() { return doctor; }
    public void setDoctor(Doctor doctor) { this.doctor = doctor; }
}
```

MedicalRecord.java

```
package com.example.hospital.entity;
import jakarta.persistence.*;
import java.time.LocalDate;
@Entity
public class MedicalRecord {
    @Id
```

```
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String diagnosis;
    private String treatment;
    private LocalDate date;
    @ManyToOne
    private Patient patient;
    public MedicalRecord() {}
    public MedicalRecord(Long id, String diagnosis, String treatment, LocalDate date, Patient patient) {
        this.id = id;
        this.diagnosis = diagnosis;
        this.treatment = treatment;
        this.date = date;
    this.patient = patient;

    }
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }
    public String getDiagnosis() { return diagnosis; }
    public void setDiagnosis(String diagnosis) { this.diagnosis = diagnosis; }
    public String getTreatment() { return treatment; }
    public void setTreatment(String treatment) { this.treatment = treatment; }
    public LocalDate getDate() { return date; }
    public void setDate(LocalDate date) { this.date = date; }
    public Patient getPatient() { return patient; }
    public void setPatient(Patient patient) { this.patient = patient; }
}
```

Repository Interfaces (com.example.hospital.repository)

```
package com.example.hospital.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import com.example.hospital.entity.*;
public interface PatientRepository extends JpaRepository<Patient, Long> {}
public interface DoctorRepository extends JpaRepository<Doctor, Long> {}
```

```
public interface AppointmentRepository extends JpaRepository<Appointment, Long> {}

public interface MedicalRecordRepository extends JpaRepository<MedicalRecord, Long> {}
```

HospitalController.java

```
package com.example.hospital.controller;

import org.springframework.web.bind.annotation.*;

import com.example.hospital.entity.*;

import com.example.hospital.repository.*;

import java.util.List;

@RestController

@RequestMapping("/api")
  public class HospitalController {

  private final PatientRepository patientRepo;

  private final DoctorRepository doctorRepo;

  private final AppointmentRepository appointmentRepo;

  private final MedicalRecordRepository medicalRecordRepo;

  public HospitalController(PatientRepository patientRepo, DoctorRepository doctorRepo,

                  AppointmentRepository appointmentRepo, MedicalRecordRepository
medicalRecordRepo) {

    this.patientRepo = patientRepo;

    this.doctorRepo = doctorRepo;

    this.appointmentRepo = appointmentRepo;

    this.medicalRecordRepo = medicalRecordRepo;

  }

  @PostMapping("/patients")

  public Patient addPatient(@RequestBody Patient patient) {

    return patientRepo.save(patient);

  }

  @GetMapping("/patients")

  public List<Patient> getAllPatients() {

    return patientRepo.findAll();

  }

  @PostMapping("/doctors")
```

```
    public Doctor addDoctor(@RequestBody Doctor doctor) {
        return doctorRepo.save(doctor);
    }
    @PostMapping("/appointments")
    public Appointment bookAppointment(@RequestBody Appointment appointment) {
        return appointmentRepo.save(appointment);
    }
    @GetMapping("/appointments")
    public List<Appointment> getAppointments() {
    return appointmentRepo.findAll();
    }
    @PostMapping("/medical-records")
    public MedicalRecord addRecord(@RequestBody MedicalRecord record) {
        return medicalRecordRepo.save(record);
    }
    @GetMapping("/patients/{id}/records")
    public List<MedicalRecord> getPatientRecords(@PathVariable Long id) {
        Patient patient = patientRepo.findById(id).orElseThrow();
        return patient.getRecords();
    }
}
```