Case Study: Library Management System

🗒️ **Objective:** Design a Library Management System where:

• Readers can borrow books

• Books belong to categories

• Authors can write multiple books

🧱 **Entities:**

1. Reader

• Each reader has a name and email.

• One reader can borrow many books.

2. Book

• Each book has a title and publish date.

• One book can be borrowed by one reader at a time.

• One book belongs to one category.

• One book is written by one author.

3. Category

• Each category has a name (e.g., Fiction, Technology).

• One category can have many books.

4. Author

• Each author has a name.

• One author can write multiple books

application.properties:

```
spring.application.name=LibraryManagement

spring.datasource.url=jdbc:mysql://localhost:3306/library_db

spring.datasource.username=root

spring.datasource.password=root
```

```
spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true
```

Entity class:

Author.java:

```
package com.example.library.entity;


import jakarta.persistence.*;

import lombok.*;


import java.util.List;


@Entity

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Author {

  @Id

  @GeneratedValue(strategy = GenerationType.IDENTITY)

  private Long id;


  private String name;


  @OneToMany(mappedBy = "author", cascade = CascadeType.ALL)

  private List<Book> books;


    public void setId(Long id2) {

        // TODO Auto-generated method stub


    }
```

```
    public Long getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public List<Book> getBooks() {
        return books;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setBooks(List<Book> books) {
        this.books = books;
    }
}
```

Book.java:

```
package com.example.library.entity;

import jakarta.persistence.*;
import lombok.*;

import java.time.LocalDate;

@Entity
@Data
```

```
@NoArgsConstructor

@AllArgsConstructor

public class Book {


  @Id

  @GeneratedValue(strategy = GenerationType.IDENTITY)

  private Long id;


  private String title;

  private LocalDate publishDate;



  public Long getId() {

        return id;

  }


    public String getTitle() {

        return title;

    }


    public LocalDate getPublishDate() {

        return publishDate;

    }


    public Reader getReader() {

        return reader;

    }


    public Category getCategory() {

        return category;
```

```
}

public Author getAuthor() {
    return author;
}

public void setId(Long id) {
    this.id = id;
}

public void setTitle(String title) {
    this.title = title;
}

public void setPublishDate(LocalDate publishDate) {
    this.publishDate = publishDate;
}

public void setReader(Reader reader) {
    this.reader = reader;
}

public void setCategory(Category category) {
    this.category = category;
}

public void setAuthor(Author author) {
    this.author = author;
}

@ManyToOne
```

```
    @JoinColumn(name = "reader_id")
    private Reader reader;

    @ManyToOne
    @JoinColumn(name = "category_id")
    private Category category;

    @ManyToOne
    @JoinColumn(name = "author_id")
    private Author author;
}
```

Category.java:

```
package com.example.library.entity;

import jakarta.persistence.*;
import lombok.*;

import java.util.List;

@Entity
@Data
@NoArgsConstructor
@AllArgsConstructor
public class Category {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;
```

```
    @OneToMany(mappedBy = "category", cascade = CascadeType.ALL)
    private List<Book> books;

    public Long getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public List<Book> getBooks() {
        return books;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setBooks(List<Book> books) {
        this.books = books;
    }

}
```

Reader.java:

```
package com.example.library.entity;

import jakarta.persistence.*;

import lombok.*;

import java.util.List;

@Entity

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Reader {

  @Id

  @GeneratedValue(strategy = GenerationType.IDENTITY)

  private Long id;

  private String name;

  private String email;

  @OneToMany(mappedBy = "reader", cascade = CascadeType.ALL)

  private List<Book> books;

    public Long getId() {

        return id;

    }

    public String getName() {

        return name;

    }
```

```
    public String getEmail() {
        return email;
    }

    public List<Book> getBooks() {
        return books;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public void setBooks(List<Book> books) {
        this.books = books;
    }

}
```

Repository:

AuthorRepository:

```
package com.example.library.repository;
```

```
import com.example.library.entity.Author;

import org.springframework.data.jpa.repository.JpaRepository;

public interface AuthorRepository extends JpaRepository<Author, Long> {}
```

BookRepository:

```
package com.example.library.repository;

import com.example.library.entity.Book;

import org.springframework.data.jpa.repository.JpaRepository;

public interface BookRepository extends JpaRepository<Book, Long> {}
```

CategoryRepository:

```
package com.example.library.repository;

import com.example.library.entity.Category;

import org.springframework.data.jpa.repository.JpaRepository;

public interface CategoryRepository extends JpaRepository<Category, Long> {}
```

ReaderRepository:

```
package com.example.library.repository;

import com.example.library.entity.Reader;

import org.springframework.data.jpa.repository.JpaRepository;

public interface ReaderRepository extends JpaRepository<Reader, Long> {}
```

Main Class:

LibraryManagementApplication:

```
package com.example.library;


import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;


@SpringBootApplication

public class LibraryManagementApplication {


    public static void main(String[] args) {

        SpringApplication.run(LibraryManagementApplication.class, args);

    }


}
```

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

🏥 **1. Overview:** The Hospital Management System helps manage patients, doctors, appointments, and medical records. It allows hospital staff to:

• Add/update patient and doctor records

• Schedule appointments

• Track medical history

application.properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/hospitaldb

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format_sql=true
```

Entity class:

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
    @JoinColumn(name = "patient_id")
    private Patient patient;

    @ManyToOne
    @JoinColumn(name = "doctor_id")
    private Doctor doctor;

    public Long getId() {
        return id;
    }

    public LocalDate getDate() {
```

```
        return date;
    }

    public LocalTime getTime() {
        return time;
    }

    public String getNotes() {
        return notes;
    }

    public Patient getPatient() {
        return patient;
    }

    public Doctor getDoctor() {
        return doctor;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public void setDate(LocalDate date) {
        this.date = date;
    }

    public void setTime(LocalTime time) {
        this.time = time;
    }
```

```
    public void setNotes(String notes) {
        this.notes = notes;
    }

    public void setPatient(Patient patient) {
        this.patient = patient;
    }

    public void setDoctor(Doctor doctor) {
        this.doctor = doctor;
    }


}
```

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 Long getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getSpecialization() {
        return specialization;
    }

    public String getEmail() {
        return email;
    }

    public String getPhone() {
        return phone;
    }

    public List<Appointment> getAppointments() {
        return appointments;
    }
```

```
    public void setId(Long id) {
        this.id = id;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setSpecialization(String specialization) {
        this.specialization = specialization;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

    public void setAppointments(List<Appointment> appointments) {
        this.appointments = appointments;
    }


}
```

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
    @JoinColumn(name = "patient_id")
    private Patient patient;

    public Long getId() {
        return id;
    }

    public String getDiagnosis() {
        return diagnosis;
    }

    public String getTreatment() {
        return treatment;
    }

    public LocalDate getDate() {
```

```
        return date;
    }

    public Patient getPatient() {
        return patient;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public void setDiagnosis(String diagnosis) {
        this.diagnosis = diagnosis;
    }

    public void setTreatment(String treatment) {
        this.treatment = treatment;
    }

    public void setDate(LocalDate date) {
        this.date = date;
    }

    public void setPatient(Patient patient) {
        this.patient = patient;
    }


}
```

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 Long getId() {
        return id;
    }

    public String getName() {
        return name;
    }
```

```
    public int getAge() {
        return age;
    }

    public String getGender() {
        return gender;
    }

    public String getAddress() {
        return address;
    }

    public List<Appointment> getAppointments() {
        return appointments;
    }

    public List<MedicalRecord> getRecords() {
        return records;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
```

```
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public void setAppointments(List<Appointment> appointments) {
        this.appointments = appointments;
    }

    public void setRecords(List<MedicalRecord> records) {
        this.records = records;
    }


}
```

Repository:

AppointmentRepository:

```
package com.example.hospital.repository;

import com.example.hospital.entity.Appointment;

import org.springframework.data.jpa.repository.JpaRepository;

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

DoctorRepository:

```
package com.example.hospital.repository;

import com.example.hospital.entity.Doctor;
import org.springframework.data.jpa.repository.JpaRepository;

public interface DoctorRepository extends JpaRepository<Doctor, Long> {}
```

MedicalRecordRepository:

```
package com.example.hospital.repository;

import com.example.hospital.entity.MedicalRecord;
import org.springframework.data.jpa.repository.JpaRepository;

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

PatientRepository:

```
package com.example.hospital.repository;

import com.example.hospital.entity.Patient;
import org.springframework.data.jpa.repository.JpaRepository;

public interface PatientRepository extends JpaRepository<Patient, Long> {}
```

HospitalController:

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

import com.example.hospital.entity.*;
import com.example.hospital.repository.*;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;
```

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

    @Autowired private PatientRepository patientRepo;
    @Autowired private DoctorRepository doctorRepo;
    @Autowired private AppointmentRepository appointmentRepo;
    @Autowired private MedicalRecordRepository recordRepo;

    // Patient
    @PostMapping("/patients")
    public Patient addPatient(@RequestBody Patient p) {
        return patientRepo.save(p);
    }

    @GetMapping("/patients")
    public List<Patient> getPatients() {
        return patientRepo.findAll();
    }

    // Doctor
    @PostMapping("/doctors")
    public Doctor addDoctor(@RequestBody Doctor d) {
        return doctorRepo.save(d);
    }

    @GetMapping("/doctors")
    public List<Doctor> getDoctors() {
        return doctorRepo.findAll();
```

```
    }

    // Appointment
    @PostMapping("/appointments")
    public Appointment addAppointment(@RequestBody Appointment a) {
        return appointmentRepo.save(a);
    }

    @GetMapping("/appointments")
    public List<Appointment> getAppointments() {
        return appointmentRepo.findAll();
    }

    // Medical Record
    @PostMapping("/medical-records")
    public MedicalRecord addRecord(@RequestBody MedicalRecord m) {
        return recordRepo.save(m);
    }

    @GetMapping("/patients/{id}/records")
    public List<MedicalRecord> getPatientRecords(@PathVariable Long id) {
        Patient p = patientRepo.findById(id).orElse(null);
        return (p != null) ? p.getRecords() : null;
    }
}
```

HospitalApplication:

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

```
import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;



@SpringBootApplication

public class HospitalApplication {



	public static void main(String[] args) {

		SpringApplication.run(HospitalApplication.class, args);

	}



}
```