Case Study: Library Management System

application.properties:

```
spring.application.name=LibraryManagement
spring.datasource.url=jdbc:mysql://localhost:3306/library_db
spring.datasource.username=root spring.
datasource.password=Swetha@57
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

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) {
 }
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;
}
@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> {}

Category.Repository:

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;

Packagecom.example.library;

Importorg.springframework.boot.SpringApplication;

Importorg.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

Public class LibraryManagenmentApplication{

```
Public static void main(String[] args){
  Spring Application run(LibraryManagementApplication.class.args);
}
}
```

CaseStudy2:

Hospital Management System:

HospitalApplicationTests;

```
package com.example.hospital;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class HospitalApplicationTests {

	@Test
	void contextLoads() {
	}

}
```

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 patientRepository;

    @Autowired
    private DoctorRepository doctorRepository;

    @Autowired
    private AppointmentRepository appointmentRepository;

    @Autowired
    private MedicalRecordRepository medicalRecordRepository;

    // ---------------- Patient ----------------

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

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

    @GetMapping("/patients/{id}/records")
    public List<MedicalRecord> getPatientRecords(@PathVariable Long id) {
        Patient patient = patientRepository.findById(id)
                .orElseThrow(() -> new RuntimeException("Patient not found with id: " + id));
```

```
    return patient.getMedicalRecords();
}

// ---------- Doctor ----------

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

@GetMapping("/doctors")
public List<Doctor> getAllDoctors() {
    return doctorRepository.findAll();
}

// ---------- Appointment ----------

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

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

// ---------- Medical Record ----------

@PostMapping("/medical-records")
public MedicalRecord addMedicalRecord(@RequestBody MedicalRecord record) {
```

```
        return medicalRecordRepository.save(record);
    }

    @GetMapping("/medical-records")
    public List<MedicalRecord> getAllMedicalRecords() {
        return medicalRecordRepository.findAll();
    }
}
```

Appontment.java:

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

import jakarta.persistence.*;
import java.time.LocalDateTime;

@Entity
public class Appointment {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private LocalDateTime dateTime;
    private String notes;

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

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

```
    // Getters and Setters
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }

    public LocalDateTime getDateTime() { return dateTime; }
    public void setDateTime(LocalDateTime dateTime) { this.dateTime = dateTime; }

    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; }
}
```

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;

    // Getters and Setters
    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; }
}
```

MedicalReport;

```
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;

    // Getters and Setters
    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; }
```

```
}
```

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> medicalRecords;

    // Getters and Setters
    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> getMedicalRecords() { return medicalRecords; }
    public void setMedicalRecords(List<MedicalRecord> medicalRecords) { this.medicalRecords =
medicalRecords; }
}
```

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> {
}
```

DocyorRepository;

```
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> {
}
```

MedicalRepository:

```
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> {
}
```

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);
	}
```

}

Application.properties:

```
# MySQL DB Connection
spring.datasource.url=jdbc:mysql://localhost:3306/hospitaldb
spring.datasource.username=root
spring.datasource.password=Swetha@57
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
```