

Case Study: Flight Reservation System (Monolithic Application)

Code:

Entity class:

//Flight.java:

```
package com.example.flightreservation.entity;

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

import java.util.ArrayList; import java.util.List;

@Entity
public class Flight {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    private Long id;

    @Column(unique = true, nullable = false)

    private String flightNumber;

    private String origin;

    private String destination;

    private LocalDateTime departureTime;

    private int seatsAvailable;

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

    private List<Reservation> reservations = new ArrayList<>();

    // Constructors

    public Flight() {}

    // Getters and Setters

    public Long getId() {

    return id;

    }
```

```
    public String getFlightNumber() {  
  
        return flightNumber;  
  
    }  
  
    public void setFlightNumber(String flightNumber) {  
  
        this.flightNumber = flightNumber;  
  
    }  
  
    public String getOrigin() {  
  
        return origin;  
  
    }  
  
    public void setOrigin(String origin) {  
  
        this.origin = origin;  
  
    }  
  
    public String getDestination() {  
  
        return destination;  
  
    }  
  
    public void setDestination(String destination) {  
  
        this.destination = destination;  
  
    }  
  
    public LocalDateTime getDepartureTime() {  
  
        return departureTime;  
  
    }
```

```
    public void setDepartureTime(LocalDateTime  
departureTime) {    this.departureTime = departureTime;  
  
    }
```

```
    public int getSeatsAvailable() {  
  
return seatsAvailable;  
  
    }
```

```
    public void setSeatsAvailable(int seatsAvailable) {  
  
this.seatsAvailable = seatsAvailable;  
  
    }
```

```
    public List<Reservation> getReservations() {  
  
return reservations;  
  
    }
```

```
    public void setReservations(List<Reservation> reservations)  
  
{    this.reservations = reservations;  
  
    }  
  
}
```

Reservation.java:

```
package com.example.flightreservation.entity;
```

```
import jakarta.persistence.*;
```

```
import java.time.LocalDateTime;
```

@Entity **public class**

Reservation {

 @Id

 @GeneratedValue(strategy = GenerationType.*IDENTITY*)

private Long id;

private String passengerName;

private String passengerEmail;

private int seatsBooked;

private LocalDateTime reservedAt;

 @ManyToOne

 @JoinColumn(name = "flight_id", nullable =

false) **private** Flight flight; // Constructors

public Reservation() {}

 // Getters and Setters

public Long getId() {

return id;

}

public String getPassengerName() {

return passengerName;

}

```
public void setPassengerName(String passengerName) {  
  
this.passengerName = passengerName;  
  
}
```

```
public String getPassengerEmail() {  
  
return passengerEmail;  
  
}
```

```
public void setPassengerEmail(String passengerEmail) {  
  
this.passengerEmail = passengerEmail;  
  
}
```

```
public int getSeatsBooked() {  
  
return seatsBooked;  
  
}
```

```
public void setSeatsBooked(int seatsBooked) {  
  
this.seatsBooked = seatsBooked;  
  
}
```

```
public LocalDateTime getReservedAt() {  
  
return reservedAt;  
  
}
```

```
public void setReservedAt(LocalDateTime reservedAt) {  
  
this.reservedAt = reservedAt;  
  
}
```

```
    public Flight getFlight() {  
  
        return flight;  
  
    }  
  
    public void setFlight(Flight flight) {  
  
        this.flight = flight;  
  
    }  
}
```

Repository:

```
//FlightRepository:  
  
package com.example.flightreservation.repository;  
  
import com.example.flightreservation.entity.Flight; import  
org.springframework.data.jpa.repository.JpaRepository;  
  
import java.util.Optional;  
  
public interface FlightRepository extends JpaRepository<Flight, Long> {  
  
    Optional<Flight> findByFlightNumber(String flightNumber);  
  
}
```

ReservationRepository:

```
package com.example.flightreservation.repository;  
  
import com.example.flightreservation.entity.Reservation;  
  
import org.springframework.data.jpa.repository.JpaRepository;
```

```
import java.util.List;
```

```
public interface ReservationRepository extends JpaRepository<Reservation, Long> {
```

```
    List<Reservation> findByFlightId(Long  
flightId); }
```

Service:

FlightService:

```
package com.example.flightreservation.service;
```

```
import com.example.flightreservation.entity.Flight; import
```

```
com.example.flightreservation.exception.FlightNotFoundException;
```

```
import com.example.flightreservation.repository.FlightRepository;
```

```
import org.springframework.beans.factory.annotation.Autowired; import
```

```
org.springframework.stereotype.Service;
```

```
import java.util.List;
```

```
@Service
```

```
public class FlightService {
```

```
    @Autowired
```

```
    private FlightRepository flightRepository;
```

```
    public Flight addFlight(Flight flight) {
```

```
return flightRepository.save(flight);
```

```
}
```

```
public List<Flight> getAllFlights() {
```

```
return flightRepository.findAll();
```

```
}
```

```
public Flight getFlightById(Long id) {
```

```
return flightRepository.findById(id)
```

```
.orElseThrow(() -> new FlightNotFoundException("Flight not found with ID: " + id));
```

```
}
```

```
public Flight updateFlight(Long id, Flight updatedFlight) {
```

```
Flight flight = getFlightById(id);
```

```
flight.setFlightNumber(updatedFlight.getFlightNumber());
```

```
flight.setOrigin(updatedFlight.getOrigin());
```

```
flight.setDestination(updatedFlight.getDestination());
```

```
flight.setDepartureTime(updatedFlight.getDepartureTime());
```

```
flight.setSeatsAvailable(updatedFlight.getSeatsAvailable());
```

```
return flightRepository.save(flight);
```

```
}
```

```
public void deleteFlight(Long id) {
```

```
flightRepository.deleteById(id);
```

```
}
```

```
}
```

ReservationService:


```
package com.example.flightreservation.service;
```

```
import com.example.flightreservation.entity.Flight; import
```

```
com.example.flightreservation.entity.Reservation; import
```

```
com.example.flightreservation.exception.FlightNotFoundException;
```

```
import
```

```
com.example.flightreservation.exception.NotEnoughSeatsException;
```

```
import com.example.flightreservation.repository.FlightRepository; import
```

```
com.example.flightreservation.repository.ReservationRepository; import
```

```
org.springframework.beans.factory.annotation.Autowired; import
```

```
org.springframework.stereotype.Service;
```

```
import java.time.LocalDateTime;
```

```
import java.util.List;
```

```
@Service public class
```

```
ReservationService {
```

```
    @Autowired    private ReservationRepository
```

```
reservationRepository;
```

```
    @Autowired    private
```

```
FlightRepository flightRepository;
```

```
public Reservation makeReservation(Long flightId, Reservation reservation) {
```

```
    Flight flight = flightRepository.findById(flightId)
```

```
        .orElseThrow(() -> new FlightNotFoundException("Flight not found with ID: " + flightId));
```

```

    if (reservation.getSeatsBooked() > flight.getSeatsAvailable()) {

        throw new NotEnoughSeatsException("Not enough seats available");

    }

    // Deduct seats    flight.setSeatsAvailable(flight.getSeatsAvailable() -
reservation.getSeatsBooked());    reservation.setFlight(flight);

reservation.setReservedAt(LocalDateTime.now());

    // Save both    flightRepository.save(flight);

return reservationRepository.save(reservation);

}

public List<Reservation> getAllReservations() {

return reservationRepository.findAll();

}

public List<Reservation> getReservationsByFlightId(Long flightId) {

return reservationRepository.findByFlightId(flightId);

}

public void cancelReservation(Long reservationId) {

    Reservation reservation = reservationRepository.findById(reservationId)

        .orElseThrow(() -> new RuntimeException("Reservation not found"));

    Flight flight = reservation.getFlight();

```

```

        // Restore seats        flight.setSeatsAvailable(flight.getSeatsAvailable() +
reservation.getSeatsBooked());

        // Save flight & delete reservation
        flightRepository.save(flight);

reservationRepository.deleteById(reservationId);

    }

}

```

Controller:

FlightController:

```

package com.example.flightreservation.controller;

import com.example.flightreservation.entity.Flight; import
com.example.flightreservation.repository.FlightRepository;

import org.springframework.beans.factory.annotation.Autowired;

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

import java.util.List;

import java.util.Optional;

@RestController

@RequestMapping("/api/flights")

public class FlightController {

    @Autowired    private

    FlightRepository flightRepository;

```

```

    @PostMapping    public Flight

addFlight(@RequestBody Flight flight) {    return

flightRepository.save(flight);

}

@GetMapping

    public List<Flight> getAllFlights() {

        return
flightRepository.findAll();

}

@GetMapping("/{id}")    public Optional<Flight>

getFlightById(@PathVariable Long id) {    return

flightRepository.findById(id);

}

@PutMapping("/{id}")    public Flight updateFlight(@PathVariable Long id,

@RequestBody Flight updatedFlight) {    return

flightRepository.findById(id).map(flight -> {

    flight.setFlightNumber(updatedFlight.getFlightNumber());

    flight.setOrigin(updatedFlight.getOrigin());

    flight.setDestination(updatedFlight.getDestination());

    flight.setDepartureTime(updatedFlight.getDepartureTime());

    flight.setSeatsAvailable(updatedFlight.getSeatsAvailable());    return

flightRepository.save(flight);

}).orElseThrow(() -> new RuntimeException("Flight not found with ID: " + id));

```

```
}
```

```
@DeleteMapping("/{id}") public void
```

```
deleteFlight(@PathVariable Long id) {
```

```
flightRepository.deleteById(id);
```

```
}
```

```
}
```

ReservationController:

```
package com.example.flightreservation.controller;
```

```
import com.example.flightreservation.entity.Reservation; import
```

```
com.example.flightreservation.service.ReservationService;
```

```
import
```

```
org.springframework.beans.factory.annotation.Autowired;
```

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

```
import java.util.List;
```

```
@RestController
```

```
@RequestMapping("/api/reservations")
```

```
public class ReservationController {
```

```
@Autowired private ReservationService
```

```
reservationService;
```

```
    @PostMapping("/flight/{flightId}")    public Reservation makeReservation(@PathVariable Long flightId,  
    @RequestBody Reservation reservation) {        return reservationService.makeReservation(flightId,  
reservation);  
  
    }
```

```
    @GetMapping    public List<Reservation>  
getAllReservations() {        return  
reservationService.getAllReservations();  
  
    }
```

```
    @GetMapping("/flight/{flightId}")    public List<Reservation>  
getReservationsByFlightId(@PathVariable Long flightId) {        return  
reservationService.getReservationsByFlightId(flightId);  
  
    }
```

```
    @DeleteMapping("/{reservationId}")  
    public void cancelReservation(@PathVariable Long reservationId) {  
reservationService.cancelReservation(reservationId);  
  
    }  
  
}
```