

Case Study: Flight Reservation System (Monolithic Application)

Flight.java

```
package com.example.flightreservation.entity;
import java.util.Date;
import jakarta.persistence.*;

@Entity
@Table(name="flight")
public class Flight {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long id;

    private String flightNumber;
    private String origin;
    private String destination;
    private Date departureTime;
    private int seatsAvailable;

    public Flight() {}
    public Flight(Long id, String flightNumber, String origin, String destination, Date
departureTime,
                    int seatsAvailable) {
        this.id = id;
        this.flightNumber = flightNumber;
        this.origin = origin;
        this.destination = destination;
        this.departureTime = departureTime;
        this.seatsAvailable = seatsAvailable;
    }
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = 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 Date getDepartureTime() {
        return departureTime;
    }
    public void setDepartureTime(Date departureTime) {
        this.departureTime = departureTime;
    }
    public int getSeatsAvailable() {
        return seatsAvailable;
    }
    public void setSeatsAvailable(int seatsAvailable) {
        this.seatsAvailable = seatsAvailable;
    }
}

```

Reservation.java

```

package com.example.flightreservation.entity;
import java.util.Date;
import jakarta.persistence.*;
@Entity
@Table(name="reservation")
public class Reservation {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long id;
    private String passengerName;
    private String passengerEmail;
    private int seatsBooked;
    private Date reservedAt;

    @ManyToOne
    @JoinColumn(name="flight_id")
    private Flight flight;
    public Reservation() {}
    public Reservation(Long id, String passengerName, String passengerEmail, int
seatsBooked, Date reservedAt, Flight flight) {
        this.id = id;
        this.passengerName = passengerName;

```

```

        this.passengerEmail = passengerEmail;
        this.seatsBooked = seatsBooked;
        this.reservedAt = reservedAt;
        this.flight = flight;
    }
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = 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 Date getReservedAt() {
        return reservedAt;
    }
    public void setReservedAt(Date reservedAt) {
        this.reservedAt = reservedAt;
    }
    public Flight getFlight() {
        return flight;
    }
    public void setFlight(Flight flight) {
        this.flight = flight;
    }
}

```

FlightRepository.java

```

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

```

```
import com.example.flightreservation.entity.Flight;
public interface FlightRepository extends JpaRepository<Flight, Long>{
}
```

ReservationRepository.java

```
package com.example.flightreservation.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import com.example.flightreservation.entity.Reservation;
public interface ReservationRepository extends JpaRepository<Reservation, Long>{
}
```

FlightService.java

```
package com.example.flightreservation.service;
import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.example.flightreservation.entity.Flight;
import com.example.flightreservation.repository.FlightRepository;
@Service
public class FlightService {
    @Autowired
    private FlightRepository flightRepo;

    public Flight addFlight(Flight flight) {
        return flightRepo.save(flight);
    }

    public List<Flight> getAllFlights(){
        return flightRepo.findAll();
    }

    public Optional<Flight> getFlightById(Long id) {
        return flightRepo.findById(id);
    }

    public Flight updateFlight(Long id, Flight flight) {
        return flightRepo.findById(id).map(f->{
            f.setFlightNumber(flight.getFlightNumber());
            f.setOrigin(flight.getOrigin());
            f.setDestination(flight.getDestination());
            f.setDepartureTime(flight.getDepartureTime());
            f.setSeatsAvailable(flight.getSeatsAvailable());
            return flightRepo.save(f);
        });
    }
}
```

```

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

    public void deleteFlight(Long id) {
        flightRepo.deleteById(id);
    }
}

```

ReservationService.java

```

package com.example.flightreservation.service;
import java.util.Date;
import java.util.List;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.example.flightreservation.entity.Flight;
import com.example.flightreservation.entity.Reservation;
import com.example.flightreservation.repository.FlightRepository;
import com.example.flightreservation.repository.ReservationRepository;
@Service
public class ReservationService {
    @Autowired
    private ReservationRepository reservationRepo;
    @Autowired
    private FlightRepository flightRepo;
    public Reservation makeReservation(Reservation reservation) {
        Long flightId = reservation.getFlight().getId();
        Flight flight = flightRepo.findById(flightId)
            .orElseThrow(() -> new RuntimeException("Flight not found with ID: " + flightId));
        if (flight.getSeatsAvailable() < reservation.getSeatsBooked()) {
            throw new RuntimeException("Not enough seats available for flight: " + flightId);
        }
        flight.setSeatsAvailable(flight.getSeatsAvailable() - reservation.getSeatsBooked());
        flightRepo.save(flight);
        reservation.setFlight(flight);
        reservation.setReservedAt(new Date());
        return reservationRepo.save(reservation);
    }
    public List<Reservation> getAllReservations() {
        return reservationRepo.findAll();
    }
    public Optional<Reservation> getReservationsByFlight(Long id) {
        return reservationRepo.findById(id);
    }
    public String cancelReservation(Long id) {
        Reservation reservation = reservationRepo.findById(id)

```

```

        .orElseThrow(() -> new RuntimeException("Reservation not found with ID: " + id));
    Flight flight = reservation.getFlight();
    flight.setSeatsAvailable(flight.getSeatsAvailable() + reservation.getSeatsBooked());
    flightRepo.save(flight);
    reservationRepo.deleteById(id);
    return "Reservation canceled successfully.";
}
}

```

FlightController.java

```

package com.example.flightreservation.controller;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import com.example.flightreservation.entity.Flight;
import com.example.flightreservation.service.FlightService;
@RestController
@RequestMapping("/api/flights")
public class FlightController {
    @Autowired
    private FlightService flightService;

    @PostMapping
    public ResponseEntity<Flight> add(@RequestBody Flight flight) {
        return ResponseEntity.ok(flightService.addFlight(flight));
    }

    @GetMapping
    public List<Flight> getAllFlights(){
        return flightService.getAllFlights();
    }

    @GetMapping("/{id}")
    public ResponseEntity<Flight> getById(@PathVariable Long id){
        return flightService.getFlightById(id)
            .map(ResponseEntity::ok)
            .orElse(ResponseEntity.notFound().build());
    }

    @PutMapping("/{id}")
    public ResponseEntity<Flight> update(@PathVariable Long id, @RequestBody Flight
flight){
        return ResponseEntity.ok(flightService.updateFlight(id, flight));
    }
}

```

```

        @DeleteMapping("/{id}")
        public ResponseEntity<String> delete(@PathVariable Long id){
            flightService.deleteFlight(id);
            return ResponseEntity.ok("Flight deleted successfully");
        }
    }
}

```

ReservationController.java

```

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;
import java.util.Optional;

@RestController
@RequestMapping("/api/reservations")
public class ReservationController {

    @Autowired
    private ReservationService reservationService;

    @PostMapping
    public Reservation makeReservation(@RequestBody Reservation reservation) {
        return reservationService.makeReservation(reservation);
    }

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

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

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

```