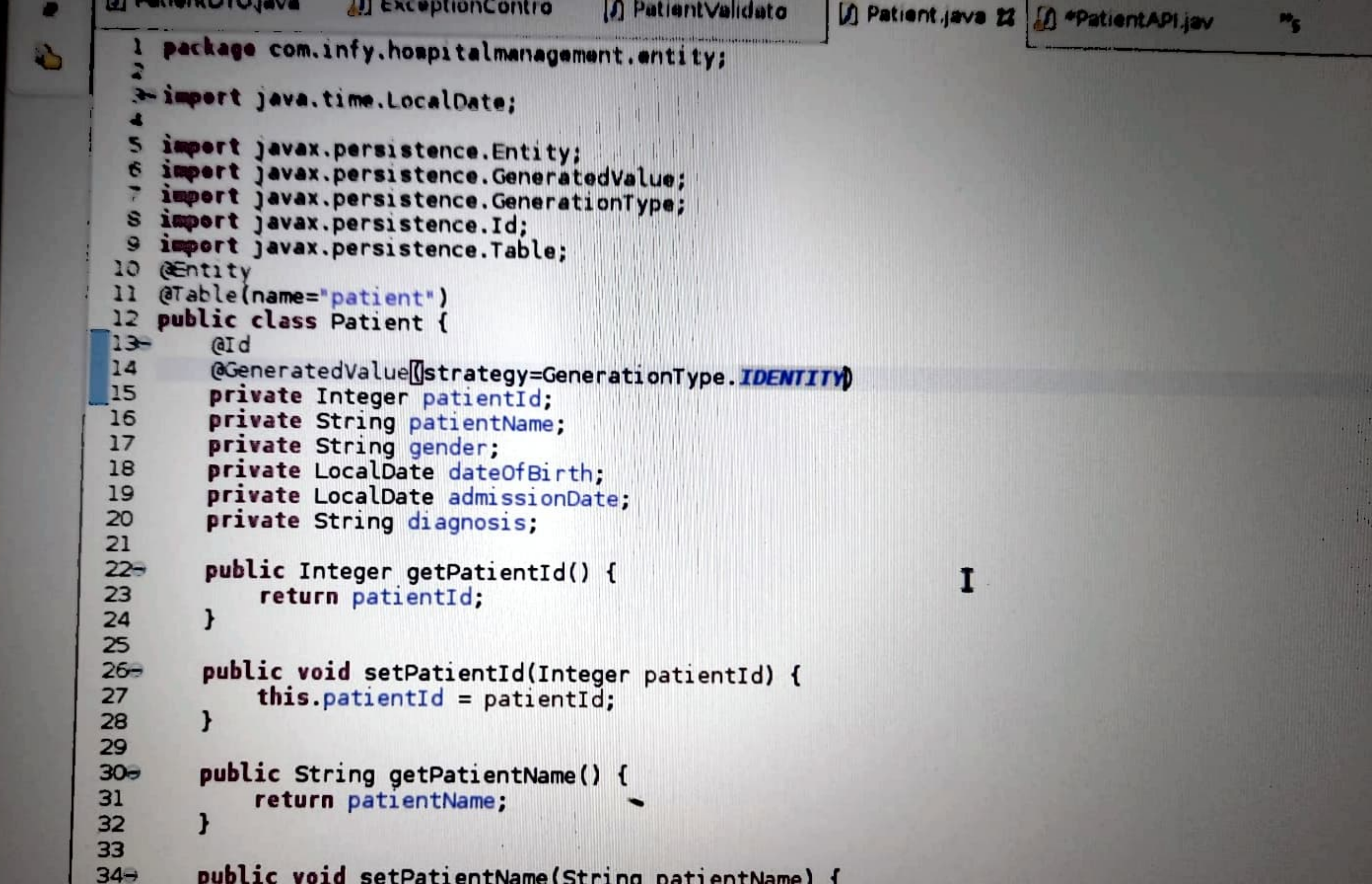
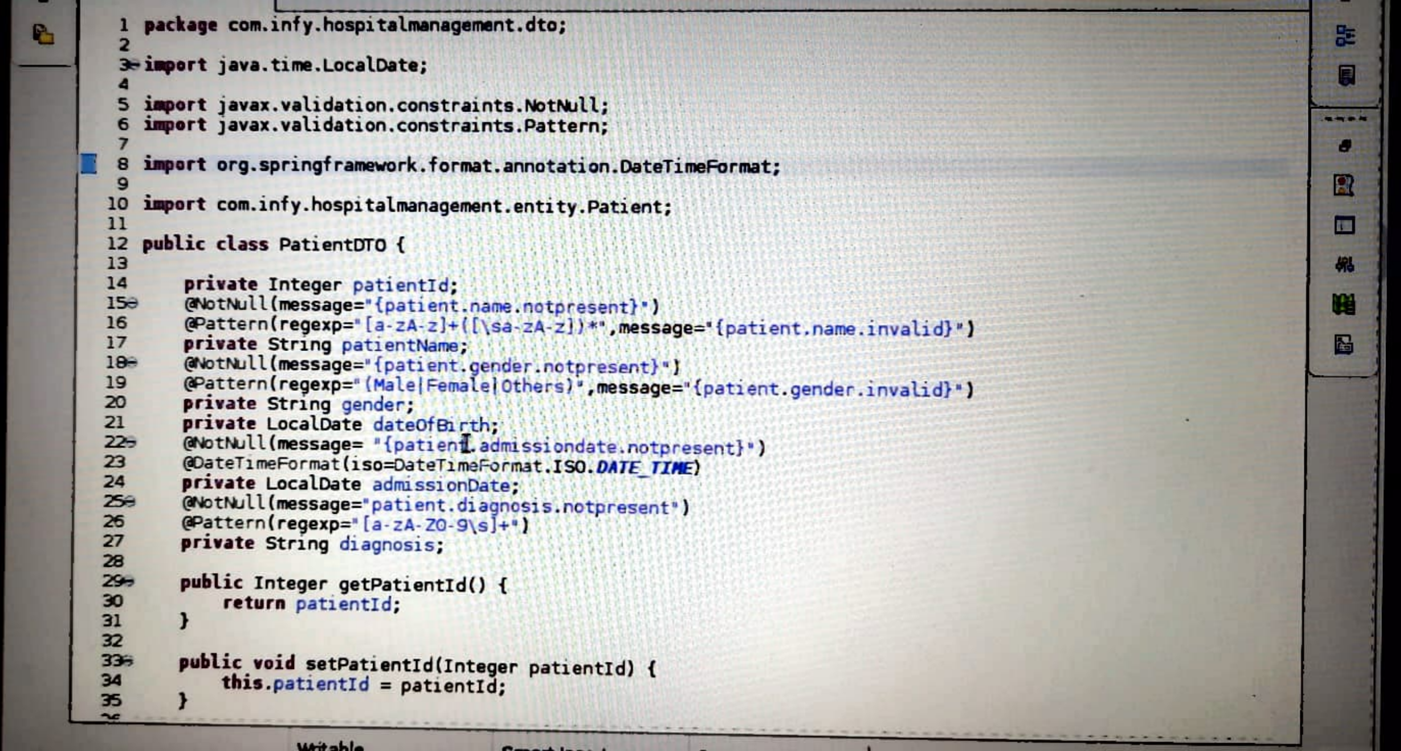
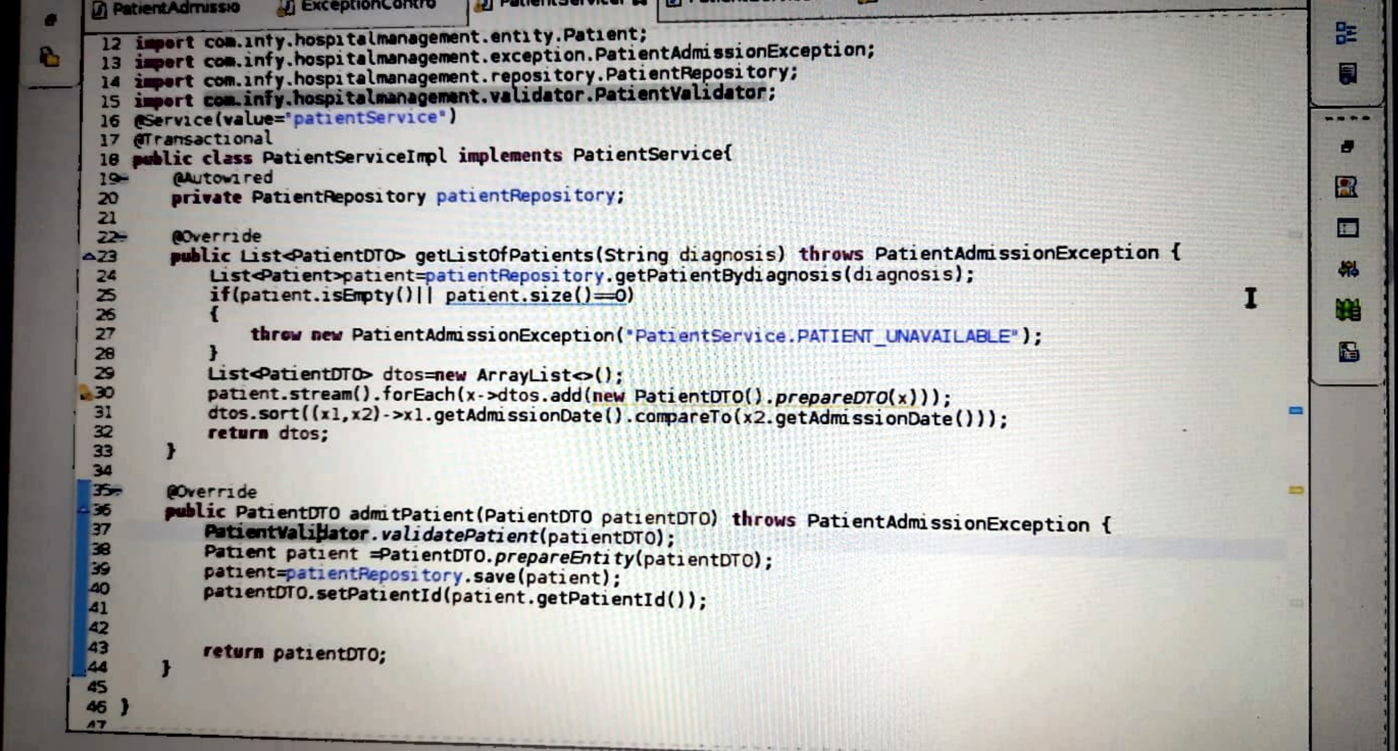


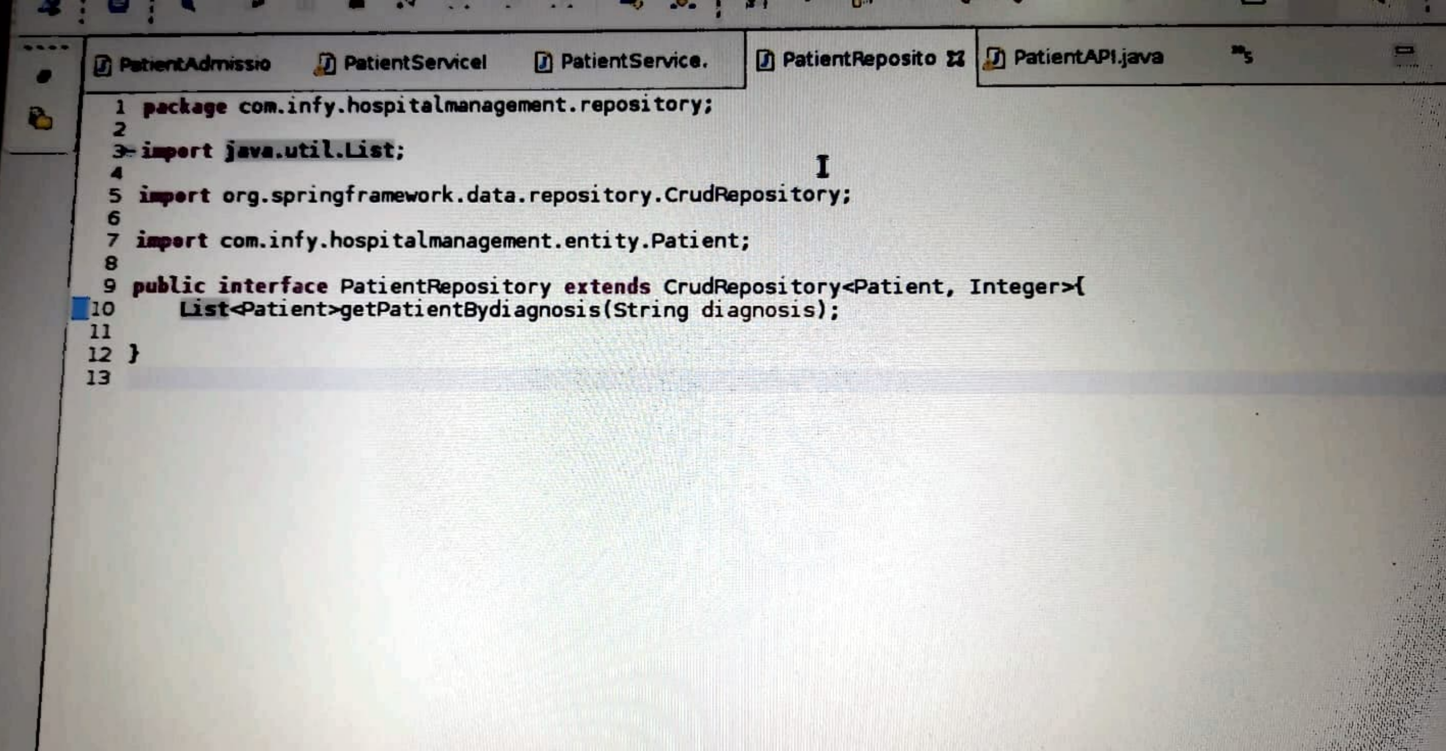
```java
package com.infy.hospitalmanagement.entity;

import java.time.LocalDate;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;
@Entity
@Table(name="patient")
public class Patient {
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer patientId;
    private String patientName;
    private String gender;
    private LocalDate dateOfBirth;
    private LocalDate admissionDate;
    private String diagnosis;

    public Integer getPatientId() {
        return patientId;
    }

    public void setPatientId(Integer patientId) {
        this.patientId = patientId;
    }

    public String getPatientName() {
        return patientName;
    }

    public void setPatientName(String patientName) {
```



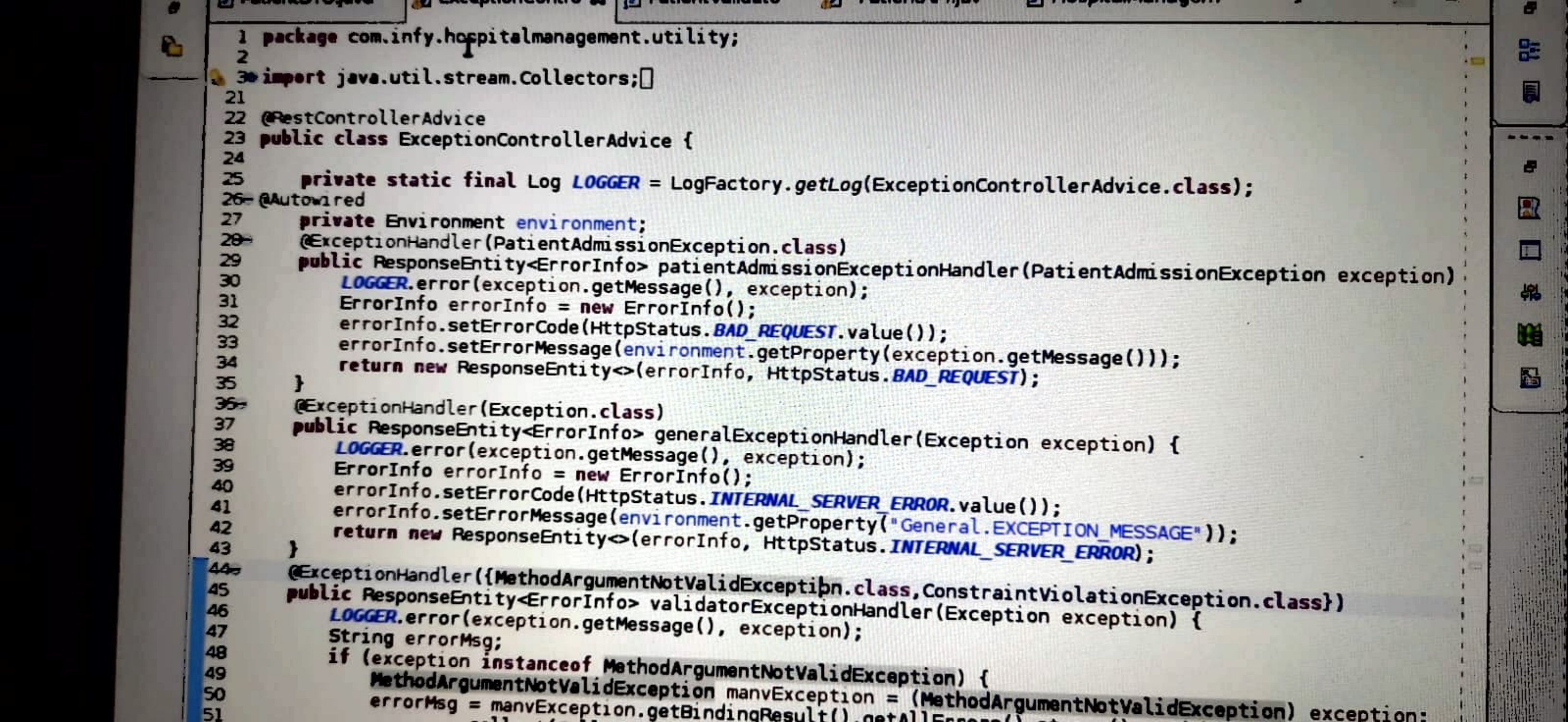
```java
package com.infy.hospitalmanagement.dto;

import java.time.LocalDate;

import javax.validation.constraints.NotNull;
import javax.validation.constraints.Pattern;

import org.springframework.format.annotation.DateTimeFormat;

import com.infy.hospitalmanagement.entity.Patient;

public class PatientDTO {

    private Integer patientId;
    @NotNull(message="{patient.name.notpresent}")
    @Pattern(regexp="[a-zA-z]+([\sa-zA-z])*",message="{patient.name.invalid}")
    private String patientName;
    @NotNull(message="{patient.gender.notpresent}")
    @Pattern(regexp="(Male|Female|Others)",message="{patient.gender.invalid}")
    private String gender;
    private LocalDate dateOfBirth;
    @NotNull(message= "{patient.admissiondate.notpresent}")
    @DateTimeFormat(iso=DateTimeFormat.ISO.DATE_TIME)
    private LocalDate admissionDate;
    @NotNull(message="patient.diagnosis.notpresent")
    @Pattern(regexp="[a-zA-Z0-9\s]+")
    private String diagnosis;

    public Integer getPatientId() {
        return patientId;
    }

    public void setPatientId(Integer patientId) {
        this.patientId = patientId;
    }
```



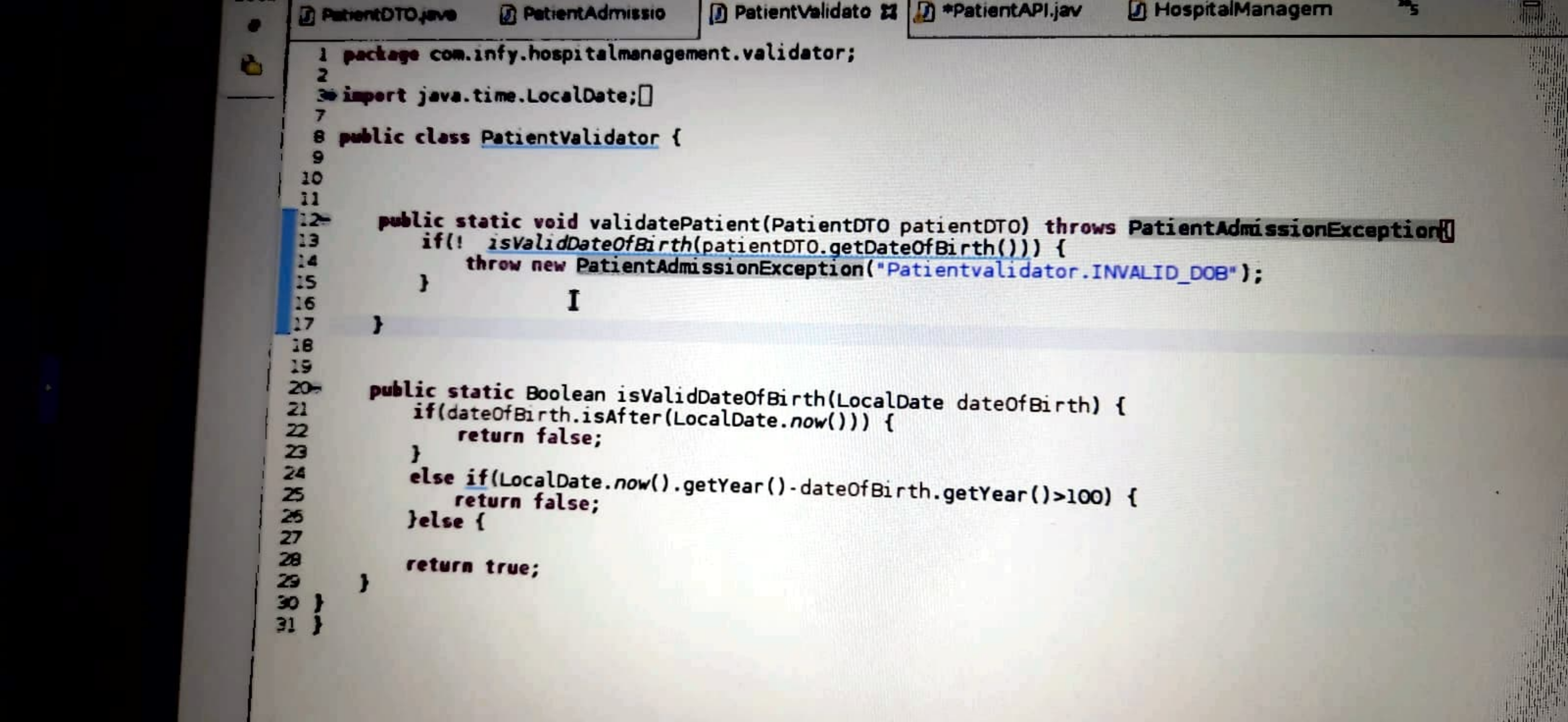
```
import com.infy.hospitalmanagement.entity.Patient;
import com.infy.hospitalmanagement.exception.PatientAdmissionException;
import com.infy.hospitalmanagement.repository.PatientRepository;
import com.infy.hospitalmanagement.validator.PatientValidator;
@Service(value="patientService")
@Transactional
public class PatientServiceImpl implements PatientService{
    @Autowired
    private PatientRepository patientRepository;

    @Override
    public List<PatientDTO> getListOfPatients(String diagnosis) throws PatientAdmissionException {
        List<Patient>patient=patientRepository.getPatientBydiagnosis(diagnosis);
        if(patient.isEmpty()|| patient.size()==0)
        {
            throw new PatientAdmissionException("PatientService.PATIENT_UNAVAILABLE");
        }
        List<PatientDTO> dtos=new ArrayList<>();
        patient.stream().forEach(x->dtos.add(new PatientDTO().prepareDTO(x)));
        dtos.sort((x1,x2)->x1.getAdmissionDate().compareTo(x2.getAdmissionDate()));
        return dtos;
    }

    @Override
    public PatientDTO admitPatient(PatientDTO patientDTO) throws PatientAdmissionException {
        PatientValidator.validatePatient(patientDTO);
        Patient patient =PatientDTO.prepareEntity(patientDTO);
        patient=patientRepository.save(patient);
        patientDTO.setPatientId(patient.getPatientId());


        return patientDTO;
    }

}
```



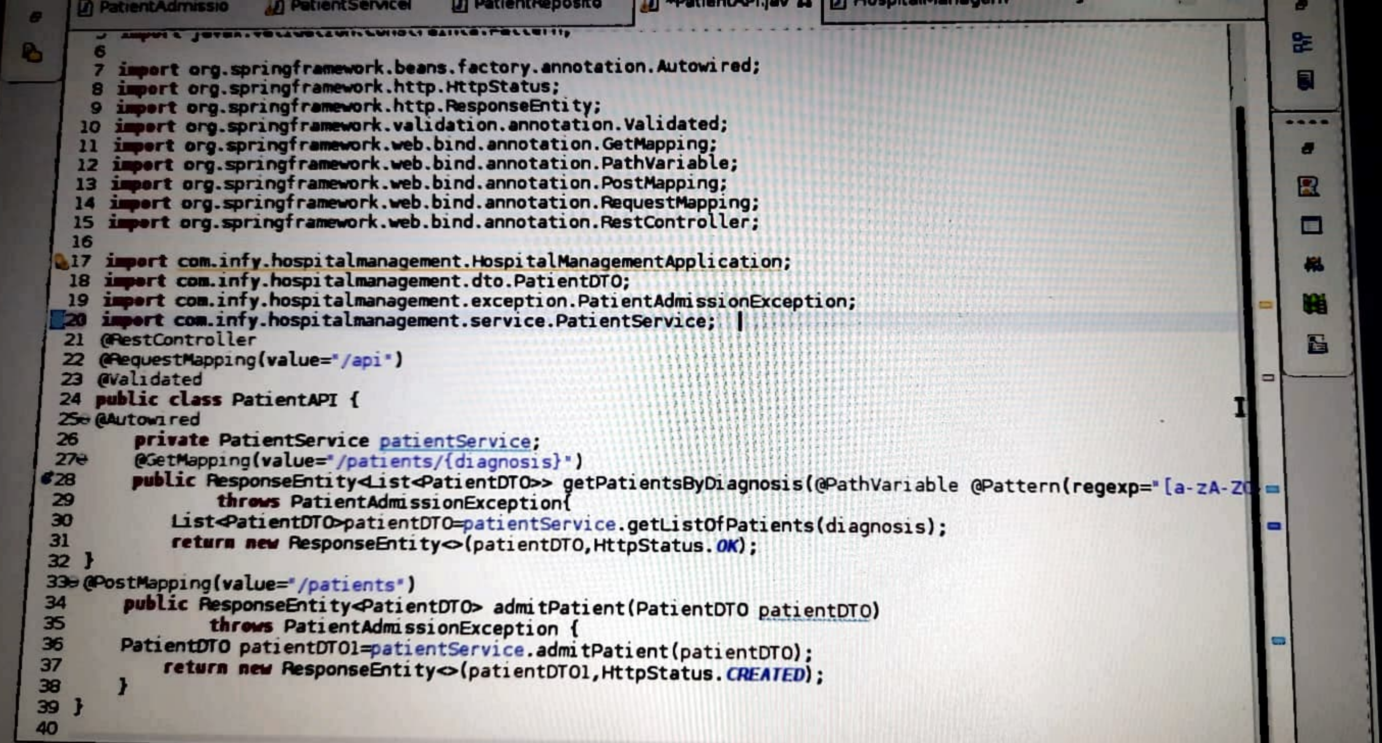
```java
package com.infy.hospitalmanagement.repository;

import java.util.List;

import org.springframework.data.repository.CrudRepository;

import com.infy.hospitalmanagement.entity.Patient;

public interface PatientRepository extends CrudRepository<Patient, Integer>{
    List<Patient>getPatientBydiagnosis(String diagnosis);

}

```



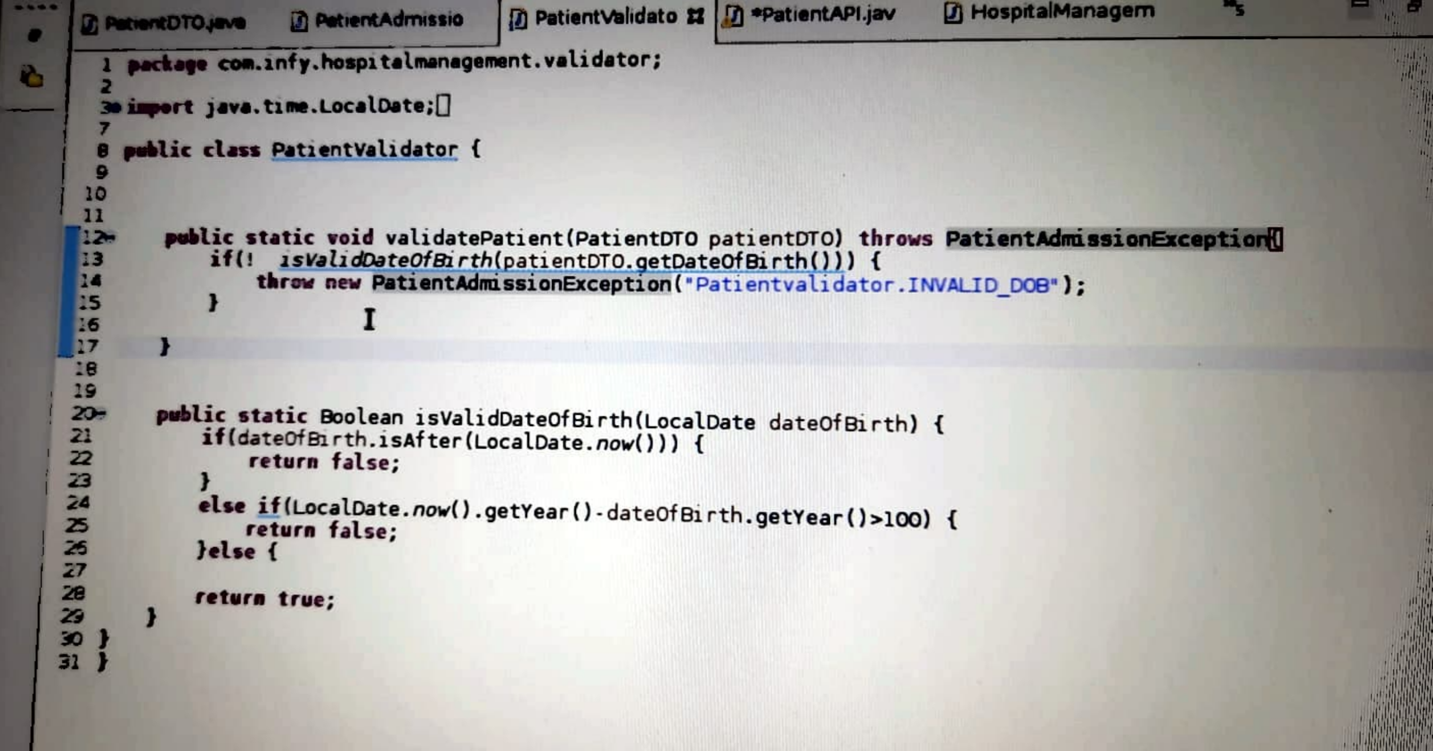
```
package com.infy.hospitalmanagement.utility;

import java.util.stream.Collectors;

@RestControllerAdvice
public class ExceptionControllerAdvice {

    private static final Log LOGGER = LogFactory.getLog(ExceptionControllerAdvice.class);
@Autowired
    private Environment environment;
    @ExceptionHandler(PatientAdmissionException.class)
    public ResponseEntity<ErrorInfo> patientAdmissionExceptionHandler(PatientAdmissionException exception)
        LOGGER.error(exception.getMessage(), exception);
        ErrorInfo errorInfo = new ErrorInfo();
        errorInfo.setErrorCode(HttpStatus.BAD_REQUEST.value());
        errorInfo.setErrorMessage(environment.getProperty(exception.getMessage()));
        return new ResponseEntity<>(errorInfo, HttpStatus.BAD_REQUEST);
    }
    @ExceptionHandler(Exception.class)
    public ResponseEntity<ErrorInfo> generalExceptionHandler(Exception exception) {
        LOGGER.error(exception.getMessage(), exception);
        ErrorInfo errorInfo = new ErrorInfo();
        errorInfo.setErrorCode(HttpStatus.INTERNAL_SERVER_ERROR.value());
        errorInfo.setErrorMessage(environment.getProperty("General.EXCEPTION_MESSAGE"));
        return new ResponseEntity<>(errorInfo, HttpStatus.INTERNAL_SERVER_ERROR);
    }
    @ExceptionHandler({MethodArgumentNotValidException.class,ConstraintViolationException.class})
    public ResponseEntity<ErrorInfo> validatorExceptionHandler(Exception exception) {
        LOGGER.error(exception.getMessage(), exception);
        String errorMsg;
        if (exception instanceof MethodArgumentNotValidException) {
            MethodArgumentNotValidException manvException = (MethodArgumentNotValidException) exception;
            errorMsg = manvException.getBindingResult().getAllErrors()
```



```java
package com.infy.hospitalmanagement.validator;

import java.time.LocalDate;

public class PatientValidator {



    public static void validatePatient(PatientDTO patientDTO) throws PatientAdmissionException{
        if(!  isValidDateOfBirth(patientDTO.getDateOfBirth())) {
            throw new PatientAdmissionException("Patientvalidator.INVALID_DOB");
        }

    }


    public static Boolean isValidDateOfBirth(LocalDate dateOfBirth) {
        if(dateOfBirth.isAfter(LocalDate.now())) {
            return false;
        }
        else if(LocalDate.now().getYear()-dateOfBirth.getYear()>100) {
            return false;
        }else {

        return true;
    }
}
}
```



```
import javax.validation.constraints.Pattern;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.infy.hospitalmanagement.HospitalManagementApplication;
import com.infy.hospitalmanagement.dto.PatientDTO;
import com.infy.hospitalmanagement.exception.PatientAdmissionException;
import com.infy.hospitalmanagement.service.PatientService;
@RestController
@RequestMapping(value="/api")
@Validated
public class PatientAPI {
@Autowired
    private PatientService patientService;
    @GetMapping(value="/patients/{diagnosis}")
    public ResponseEntity<List<PatientDTO>> getPatientsByDiagnosis(@PathVariable @Pattern(regexp="[a-zA-Z
        throws PatientAdmissionException{
      List<PatientDTO>patientDTO=patientService.getListOfPatients(diagnosis);
      return new ResponseEntity<>(patientDTO,HttpStatus.OK);
}
@PostMapping(value="/patients")
    public ResponseEntity<PatientDTO> admitPatient(PatientDTO patientDTO)
            throws PatientAdmissionException {
    PatientDTO patientDTO1=patientService.admitPatient(patientDTO);
        return new ResponseEntity<>(patientDTO1,HttpStatus.CREATED);
    }
}

```



```java
package com.infy.hospitalmanagement.validator;

import java.time.LocalDate;

public class PatientValidator {



    public static void validatePatient(PatientDTO patientDTO) throws PatientAdmissionException{
        if(!  isValidDateOfBirth(patientDTO.getDateOfBirth())) {
            throw new PatientAdmissionException("Patientvalidator.INVALID_DOB");
        }

    }


    public static Boolean isValidDateOfBirth(LocalDate dateOfBirth) {
        if(dateOfBirth.isAfter(LocalDate.now())) {
            return false;
        }
        else if(LocalDate.now().getYear()-dateOfBirth.getYear()>100) {
            return false;
        }else {

        return true;
    }
}
}
```