

```
1 package com.infy.hospitalmanagement.entity;
2
3 import java.time.LocalDate;
4
5 import javax.persistence.Entity;
6 import javax.persistence.GeneratedValue;
7 import javax.persistence.GenerationType;
8 import javax.persistence.Id;
9 import javax.persistence.Table;
10 @Entity
11 @Table(name="patient")
12 public class Patient {
13     @Id
14     @GeneratedValue(strategy=GenerationType.IDENTITY)
15     private Integer patientId;
16     private String patientName;
17     private String gender;
18     private LocalDate dateOfBirth;
19     private LocalDate admissionDate;
20     private String diagnosis;
21
22     public Integer getPatientId() {
23         return patientId;
24     }
25
26     public void setPatientId(Integer patientId) {
27         this.patientId = patientId;
28     }
29
30     public String getPatientName() {
31         return patientName;
32     }
33
34     public void setPatientName(String patientName) {
```

I

```
1 package com.infy.hospitalmanagement.dto;
2
3 import java.time.LocalDate;
4
5 import javax.validation.constraints.NotNull;
6 import javax.validation.constraints.Pattern;
7
8 import org.springframework.format.annotation.DateTimeFormat;
9
10 import com.infy.hospitalmanagement.entity.Patient;
11
12 public class PatientDTO {
13
14     private Integer patientId;
15     @NotNull(message="{patient.name.notpresent}")
16     @Pattern(regexp="[a-zA-z]+([\\s-a-zA-z])*", message="{patient.name.invalid}")
17     private String patientName;
18     @NotNull(message="{patient.gender.notpresent}")
19     @Pattern(regexp="(Male|Female|Others)", message="{patient.gender.invalid}")
20     private String gender;
21     private LocalDate dateOfBirth;
22     @NotNull(message="{patient.admissiondate.notpresent}")
23     @DateTimeFormat(iso=DateTimeFormat.ISO.DATE_TIME)
24     private LocalDate admissionDate;
25     @NotNull(message="patient.diagnosis.notpresent")
26     @Pattern(regexp="[a-zA-Z0-9\\s]+")
27     private String diagnosis;
28
29     public Integer getPatientId() {
30         return patientId;
31     }
32
33     public void setPatientId(Integer patientId) {
34         this.patientId = patientId;
35     }
36 }
```

```

12 import com.infy.hospitalmanagement.entity.Patient;
13 import com.infy.hospitalmanagement.exception.PatientAdmissionException;
14 import com.infy.hospitalmanagement.repository.PatientRepository;
15 import com.infy.hospitalmanagement.validator.PatientValidator;
16 @Service(value="patientService")
17 @Transactional
18 public class PatientServiceImpl implements PatientService{
19     @Autowired
20     private PatientRepository patientRepository;
21
22     @Override
23     public List<PatientDTO> getListOfPatients(String diagnosis) throws PatientAdmissionException {
24         List<Patient> patient=patientRepository.getPatientBydiagnosis(diagnosis);
25         if(patient.isEmpty()|| patient.size()==0)
26         {
27             throw new PatientAdmissionException("PatientService.PATIENT_UNAVAILABLE");
28         }
29         List<PatientDTO> dtos=new ArrayList<>();
30         patient.stream().forEach(x->dtos.add(new PatientDTO().prepareDTO(x)));
31         dtos.sort((x1,x2)->x1.getAdmissionDate().compareTo(x2.getAdmissionDate()));
32         return dtos;
33     }
34
35     @Override
36     public PatientDTO admitPatient(PatientDTO patientDTO) throws PatientAdmissionException {
37         PatientValidator.validatePatient(patientDTO);
38         Patient patient =PatientDTO.prepareEntity(patientDTO);
39         patient=patientRepository.save(patient);
40         patientDTO.setPatientId(patient.getPatientId());
41
42
43         return patientDTO;
44     }
45
46 }
47

```



PatientAdmissio

PatientServiceI

PatientService.

PatientReposito

PatientAPI.java

```
1 package com.infy.hospitalmanagement.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.repository.CrudRepository;
6
7 import com.infy.hospitalmanagement.entity.Patient;
8
9 public interface PatientRepository extends CrudRepository<Patient, Integer>{
10     List<Patient>getPatientBydiagnosis(String diagnosis);
11
12 }
13
```

```

1 package com.infy.hospitalmanagement.utility;
2
3 import java.util.stream.Collectors;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 @RestControllerAdvice
23 public class ExceptionControllerAdvice {
24
25     private static final Log LOGGER = LogFactory.getLog(ExceptionControllerAdvice.class);
26     @Autowired
27     private Environment environment;
28     @ExceptionHandler(PatientAdmissionException.class)
29     public ResponseEntity<ErrorInfo> patientAdmissionExceptionHandler(PatientAdmissionException exception) {
30         LOGGER.error(exception.getMessage(), exception);
31         ErrorInfo errorInfo = new ErrorInfo();
32         errorInfo.setErrorCode(HttpStatus.BAD_REQUEST.value());
33         errorInfo.setErrorMessage(environment.getProperty(exception.getMessage()));
34         return new ResponseEntity<>(errorInfo, HttpStatus.BAD_REQUEST);
35     }
36     @ExceptionHandler(Exception.class)
37     public ResponseEntity<ErrorInfo> generalExceptionHandler(Exception exception) {
38         LOGGER.error(exception.getMessage(), exception);
39         ErrorInfo errorInfo = new ErrorInfo();
40         errorInfo.setErrorCode(HttpStatus.INTERNAL_SERVER_ERROR.value());
41         errorInfo.setErrorMessage(environment.getProperty("General.EXCEPTION_MESSAGE"));
42         return new ResponseEntity<>(errorInfo, HttpStatus.INTERNAL_SERVER_ERROR);
43     }
44     @ExceptionHandler({MethodArgumentNotValidException.class, ConstraintViolationException.class})
45     public ResponseEntity<ErrorInfo> validatorExceptionHandler(Exception exception) {
46         LOGGER.error(exception.getMessage(), exception);
47         String errorMsg;
48         if (exception instanceof MethodArgumentNotValidException) {
49             MethodArgumentNotValidException manvException = (MethodArgumentNotValidException) exception;
50             errorMsg = manvException.getBindingResult().getAllErrors().get(0).getDefaultMessage();
51         }
52     }
53 }

```

```
1 package com.infy.hospitalmanagement.validator;
2
3 import java.time.LocalDate;
4
5 public class PatientValidator {
6
7     public static void validatePatient(PatientDTO patientDTO) throws PatientAdmissionException {
8         if(! isValidDateOfBirth(patientDTO.getDateOfBirth())) {
9             throw new PatientAdmissionException("Patientvalidator.INVALID_DOB");
10        }
11    }
12
13    public static Boolean isValidDateOfBirth(LocalDate dateOfBirth) {
14        if(dateOfBirth.isAfter(LocalDate.now())) {
15            return false;
16        }
17        else if(LocalDate.now().getYear()-dateOfBirth.getYear()>100) {
18            return false;
19        }
20        else {
21            return true;
22        }
23    }
24 }
25
26 }
```



```

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-Z0-9_]{1,100}")
        throws PatientAdmissionException {
        List<PatientDTO> patientDTOs = patientService.getListOfPatients(diagnosis);
        return new ResponseEntity<>(patientDTOs, HttpStatus.OK);
    }

    @PostMapping(value="/patients")
    public ResponseEntity<PatientDTO> admitPatient(PatientDTO patientDTO)
        throws PatientAdmissionException {
        PatientDTO patientDTO1 = patientService.admitPatient(patientDTO);
        return new ResponseEntity<>(patientDTO1, HttpStatus.CREATED);
    }
}

```

```
1 package com.infy.hospitalmanagement.validator;
2
3 import java.time.LocalDate;
4
5
6
7
8 public class PatientValidator {
9
10
11
12     public static void validatePatient(PatientDTO patientDTO) throws PatientAdmissionException {
13         if(! isValidDateOfBirth(patientDTO.getDateOfBirth())) {
14             throw new PatientAdmissionException("Patientvalidator.INVALID_DOB");
15         }
16     }
17
18
19
20     public static Boolean isValidDateOfBirth(LocalDate dateOfBirth) {
21         if(dateOfBirth.isAfter(LocalDate.now())) {
22             return false;
23         }
24         else if(LocalDate.now().getYear()-dateOfBirth.getYear()>100) {
25             return false;
26         }else {
27
28             return true;
29         }
30     }
31 }
```