



ASSESSMENT INSTRUCTIONS

PROJECT INSTRUCTIONS

PROBLEM STATEMENT

## Infosys Certified Spring Associate Hands-On Assessment

### Problem Statement:

In this assessment, you are going to build an **Hospital Management** application which can be used to admit a patient and view all the patients of a given diagnosis. You will have to implement the following functionalities in the application,

1. Admit a new patient
2. Get all patients from a diagnosis

### Project Structure provided:

File Name	Status
application.properties	
com.infy.hospitalmanagement.HospitalManagementApplication.java	
com.infy.hospitalmanagement.exception.PatientAdmissionException.java	
com.infy.hospitalmanagement.service.PatientService.java	

Go to top

17:47

In this assessment, you are going to build an **Hospital Management** application which can be used to admit a patient and view all the patients of a given diagnosis. You will have to implement the following functionalities in the application,

1. Admit a new patient
2. Get all patients from a diagnosis

#### Project Structure provided:

File Name	Status
application.properties	
com.infy.hospitalmanagement.HospitalManagementApplication.java	
com.infy.hospitalmanagement.exception.PatientAdmissionException.java	
com.infy.hospitalmanagement.service.PatientService.java	
com.infy.hospitalmanagement.utility.ErrorInfo.java	
log4j2.properties	
HospitalManagement_TableScripts.sql	
ValidationMessages.properties	Implemented

[Go to top](#)

HospitalManagement_TableScripts.sql	
ValidationMessages.properties	
com.infy.hospitalmanagement.dto.PatientDTO.java	To be modified
com.infy.hospitalmanagement.entity.Patient.java	To be modified
com.infy.hospitalmanagement.utility.ExceptionControllerAdvice.java	
com.infy.hospitalmanagement.api.PatientAPI.java	
com.infy.hospitalmanagement.repository.PatientRepository.java	
com.infy.hospitalmanagement.service.PatientServiceImpl.java	To be implemented
com.infy.hospitalmanagement.validator.PatientValidator.java	

## 1. Entity Classes

### com.infy.hospitalmanagement.entity.Patient.java

- This class has been implemented for you. You have to map this class to the **patient** table with **patientId** as the **primary key**
- Generate the **patientId** using the **IDENTITY** strategy

com.infy.hospitalmanagement.entity.Patient
<input checked="" type="checkbox"/> patientId: Integer <input checked="" type="checkbox"/> patientName: String <input checked="" type="checkbox"/> gender: String

Go to top

## ASSESSMENT INSTRUCTIONS

## PROJECT INSTRUCTIONS

## PROBLEM STATEMENT

### **com.infy.hospitalmanagement.entity.Patient.java**

- This class has been implemented for you. You have to map this class to the patient table with **patientId** as the **primary key**
- Generate the **patientId** using the **IDENTITY** strategy

com.infy.hospitalmanagement.entity.Patient	
▫	patientId: Integer
▫	patientName: String
▫	gender: String
▫	dateOfBirth: LocalDate
▫	admissionDate: LocalDate
▫	diagnosis: String
⌚	Patient()
⌚	Patient(patientId: Integer, patientName: String, gender: String, dateOfBirth: LocalDate, admissionDate: LocalDate, diagnosis: String)
⌚	getPatientId(): Integer
⌚	setPatientId(patientId: Integer): void
⌚	getPatientName(): String
⌚	setPatientName(patientName: String): void
⌚	getGender(): String
⌚	setGender(gender: String): void
⌚	getDateOfBirth(): LocalDate
⌚	setDateOfBirth(dateOfBirth: LocalDate): void
⌚	getAdmissionDate(): LocalDate
⌚	setAdmissionDate(admissionDate: LocalDate): void
⌚	getDiagnosis(): String
⌚	setDiagnosis(diagnosis: String): void

## 2. DTO Class

Go to ↑

● setDiagnosis(diagnosis: String): void

## 2. DTO Class

### **com.infy.hospitalmanagement.dto.PatientDTO.java**

This is a DTO (Data Transfer Object) class that holds patient details such as patient id, patient name, gender, date of birth, admission date and diagnosis.

- Use appropriate annotations from the Java Bean Validation API for validating patient details attributes of this class. (Refer below admitPatient() method of PatientAPI class for validation rules).

## 3. Validator Class

### **com.infy.hospitalmanagement.validator.PatientValidator.java**

Implement this class according to the class diagram and the description given below:

com.infy.hospitalmanagement.validator.PatientValidator
validatePatient(patientDTO: PatientDTO): void
isValidDateOfBirth(dateOfBirth: LocalDate): Boolean

Method Description:

### **1. void validatePatient(PatientDTO patientDTO)**

This method receives an object of PatientDTO and calls the respective methods to validate the values and throws an object of PatientAdmissionException with the message, based on the violation, as given below,

[Go to top](#)

The screenshot shows a web browser window with the URL `file:///home/user/Desktop/user_repo/QP/index.html`. The page content includes navigation links for 'ASSESSMENT INSTRUCTIONS', 'PROJECT INSTRUCTIONS', and 'PROBLEM STATEMENT'. A text block states: 'and throws an object of PatientAdmissionException with the message, based on the violation, as given below,' followed by a table mapping violations to messages.

Violation for	Message
dateOfBirth	PatientValidator.INVALID_DOB

## 2. Boolean isValidDateOfBirth(LocalDate dateOfBirth)

- This method validates the **dateOfBirth** received
- **dateOfBirth** should not be a future date.
- **dateOfBirth** should not be a date which is lesser than 100 years from today's date.
- If the above conditions fail then return false, else return true

## 4. Repository

### **com.infy.hospitalmanagement.repository.PatientRepository.java**

- This interface should extend the appropriate repository interface
- Add the following method to the interface
  - This method should accept the name of the diagnosis and return the list of patients having same value of **diagnosis**.

## 5. Service Class

### **com.infy.hospitalmanagement.service.PatientServiceImpl.java**

Implement this class according to the class diagram and the description given below:

The screenshot shows a desktop environment with a taskbar at the bottom. It displays several open browser tabs, including Mozilla Firefox and Postman, and a workspace window titled 'workspace - lap...'. The system tray shows the date and time as 17:44.

ASSESSMENT INSTRUCTIONS

PROJECT INSTRUCTIONS

PROBLEM STATEMENT

**com.infy.hospitalmanagement.service.PatientServiceImpl.java**

Implement this class according to the class diagram and the description given below:

com.infy.hospitalmanagement.service.PatientServiceImpl
patientRepository: PatientRepository
getListOfPatients(diagnosis: String): List<PatientDTO>
admitPatient(patientDTO: PatientDTO): PatientDTO

- Inject the **PatientRepository** objects using the proper annotation
- Mark it with the appropriate stereotype annotation with the value **patientService**
- Mark this class with necessary annotation to manage transactions

Method Description:

**1. List<PatientDTO> getListOfPatients(String diagnosis)**

- This method is used to get all the patients having similar diagnosis
- Invoke the appropriate method of the PatientRepository interface which will return all the patients on the basis of their diagnosis
- If no patients details are available, throw an object of the **PatientAdmissionException** with the message **PatientService.PATIENT\_UNAVAILABLE**
- Else, populate a list, **List<PatientDTO>** with all the patient details in the descending order of the admission date and return the sorted list

Hint: You can use the **prepareDTO()** method of the PatientDTO class to convert the respective objects to PatientDTO

**2. PatientDTO admitPatient(PatientDTO patientDTO)**

- This method is used to admit a new patient details to the database

Go to top

## PatientDTO

### 2. PatientDTO admitPatient(PatientDTO patientDTO)

- This method is used to admit a new patient details to the database
- Invoke the validatePatient() method of the PatientValidator class to validate the object of the PatientDTO class received as parameter
- Create and populate a new object of **Patient** with the appropriate values from the **PatientDTO** object  
(Hint: You can use the prepareEntity() method of the PatientDTO class for this)
- Then invoke the appropriate method of the PatientRepository to add the **Patient** object to the database which will return the same object with the auto-generated **patientId**
- Set the **patientId** of the **Patient** object obtained in the previous step to the existing **PatientDTO** object
- Return the modified **PatientDTO** object

## 6. API Class

### com.infy.hospitalmanagement.api.PatientAPI.java

Implement this class according to the class diagram and description given below:

	com.infy.hospitalmanagement.api.PatientAPI
	patientService: PatientService
	getPatientByDiagnosis(diagnosis: String): ResponseEntity<List<PatientDTO>>
	admitPatient(patientDTO: PatientDTO): ResponseEntity<PatientDTO>

- Annotate this class with appropriate annotation to declare it as REST controller class
- Annotate this class with appropriate annotation so that all its methods are mapped with **api** as base URL
- Annotate this class with appropriate annotation to enable the bean validation support
- Inject instance of PatientService class using appropriate annotation

[Go to top](#)

- Annotate this class with appropriate annotation so that all its methods are mapped with **api** as base URL
- Annotate this class with appropriate annotation to enable the bean validation support
- Inject instance of PatientService class using appropriate annotation

**Method Description:****1. ResponseEntity<List<PatientDTO>> getPatientsByDiagnosis (String diagnosis)**

- This method is to retrieve all the patients having the given diagnosis
- It accepts **GET** request with URI **patients/{diagnosis}**
- The path variable diagnosis should be validated as below (use **Bean Validation API** for validation),
  - **diagnosis** can have words containing alphabets and numbers, and each word should be separated by a single space. If invalid, set **Diagnosis should contain only numbers, alphabets and spaces** as the validation message

**Note:** The validation message should not be hard-coded. It should be read from the **ValidationMessages.properties** file

- Once validated, invoke the appropriate service class method by passing the diagnosis to get the **List<PatientDTO>**
- Return an object of **ResponseEntity** with the **List<PatientDTO>** object obtained in the previous step as a response and the HTTP response status as **OK**.

**2. public ResponseEntity<PatientDTO> admitPatient (PatientDTO patientDTO)**

- This method admits a patient in the HospitalManagement application
- It accepts **POST** request with URI **patients**
- It receives patient details in the form of JSON. The JSON data should be populated in **PatientDTO** parameter of this method after successful validation.

[Go to top](#)

## **2. public ResponseEntity<PatientDTO> admitPatient (PatientDTO patientDTO)**

- This method admits a patient in the HospitalManagement application
- It accepts **POST** request with URI **patients**
- It receives patient details in the form of JSON. The JSON data should be populated in **PatientDTO** parameter of this method after successful validation.
- The patient details will consist of the patient name, gender, date of birth, admission date and diagnosis. Using **Bean Validation API** validate the details according to the description given below (**Note:** The validation message should not be hard-coded. It should be read from the **ValidationMessages.properties** file),

- **patientName:**

If patientName is not provided then set **Patient name is mandatory** as the validation message  
If it is provided, it should contain only words of English alphabets and each word should be separated by a space. If invalid, set **Patient name should have only words and each word should be separated by a space** as the validation message

- **gender:**

If gender is not provided then set **Gender is mandatory for admission** as the validation message  
If it is provided, it should be Male, Female or Others. If invalid, set **Gender should be Male, Female or Others** as the validation message

- **admissionDate:**

If admissionDate is not provided then set **Date of admission is mandatory** as the validation message  
If it is provided, it should be today's date only. If invalid, set **Date of admission should be today's date** as the validation message

- **diagnosis:**

[Go to top](#)

- **diagnosis:**

If diagnosis is not provided then set **Diagnosis is mandatory** as the validation message

If it is provided, it should contain only numbers, alphabets and spaces as the validation message

- Then invoke the appropriate service class method to admit the patient by passing the **PatientDTO** object which will return the same object with the **patientId** updated.
- Return an object of **ResponseEntity** with the PatientDTO object obtained in the previous step as the response and the HTTP response status as **CREATED**.

## 7. Utility Classes

### **com.infy.hospitalmanagement.utility.ExceptionControllerAdvice.java**

Annotate the following ExceptionControllerAdvice to log and handle all the exceptions of the API class,

com.infy.hospitalmanagement.utility.ExceptionControllerAdvice
LOGGER: Log
environment: Environment
patientAdmissionExceptionHandler(exception: PatientAdmissionException): ResponseEntity<ErrorInfo>
generalExceptionHandler(exception: Exception): ResponseEntity<ErrorInfo>
validatorExceptionHandler(exception: Exception): ResponseEntity<ErrorInfo>

- Inject Environment objects using appropriate annotation
- The body of the methods have been implemented for you

Method Description:

#### 1. **ResponseEntity<ErrorInfo> patientAdmissionExceptionHandler(PatientAdmissionException exception)**

This method is for handling Patient Admission Exception

[Go to top](#)

## ASSESSMENT INSTRUCTIONS

## PROJECT INSTRUCTIONS

## PROBLEM STATEMENT

The body of the methods have been implemented for you

### Method Description:

1. **ResponseType<ErrorInfo> patientAdmissionExceptionHandler(PatientAdmissionException exception)**
  - This method is for handling **PatientAdmissionException** exception
  - Add appropriate annotation to make this method as exception handler for the PatientAdmissionException
2. **ResponseType<ErrorInfo> generalExceptionHandler(Exception exception)**
  - This method is for handling general exceptions
  - Add appropriate annotation to make this method as exception handler for any type of exceptions
3. **ResponseType<ErrorInfo> validatorExceptionHandler(Exception exception)**
  - This method is for handling **MethodArgumentNotValidException** and **ConstraintViolationException** exceptions
  - Add appropriate annotation to make this method as exception handler method for **MethodArgumentNotValidException** and **ConstraintViolationException** exceptions

Please find in the table below sample input and output for REST end points.

**Note:** The date values have been used considering the present date (today) to be 14th June, 2021. Please change the dates appropriately to test your implementation.

[Go to top](#)

file:///home/user/Desktop/user\_repo/QP/index.html

**ASSESSMENT INSTRUCTIONS**    **PROJECT INSTRUCTIONS**    **PROBLEM STATEMENT**

Please find in the table below sample input and output for REST end points.

**Note:** The date values have been used considering the present date (today) to be 14th June, 2021. Please change the dates appropriately to test your implementation.

<b>REST end point</b>	<b>Request Type(valid/invalid)</b>	<b>End point URI with sample input</b>	<b>Sample Response</b>
Get all the patients for the diagnosis	Valid test data	api/patients/Typhoid	[ { "patientId": 1006, "patientName": "Mark", "gender": "Male", "dateOfBirth": "2015-09-12", "admissionDate": "2021-06-14", "diagnosis": "Typhoid" }, { "patientId": 1001, "patientName": "Matthew", "gender": "Male", "dateOfBirth": "1998-08-21", "admissionDate": "2021-01-02", "diagnosis": "Typhoid" }]

[Go to top](#)

ASSESSMENT INSTRUCTIONS		PROJECT INSTRUCTIONS	PROBLEM STATEMENT
			<pre> dateOfBirth: "1998-08-22", "admissionDate": "2021-01-02", "diagnosis": "Typhoid" }, { "patientId": 1003, "patientName": "John", "gender": "Male", "dateOfBirth": "2015-09-12", "admissionDate": "2020-09-20", "diagnosis": "Typhoid" } ] </pre>
Invalid test data 1		api/patients/Typhoids	<pre>{   "errorMessage": "Diagnosis should contain only numbers, alphabets and spaces",   "errorCode": 400 }</pre>
Invalid test data 2		api/patients/Cholera	<pre>{   "errorMessage": "No patient available with given diagnosis",   "errorCode": 400 }</pre>
Admit a	Valid test data	api/patients	<a href="#">Go to top</a>

			}
	Invalid test data 2	api/patients/Cholera	{ "errorMessage": "No patient available with given diagnosis", "errorCode": 400 }
Admit a patient	Valid test data	api/patients  Sample JSON: { "patientName": "Arne Johansson", "gender": "Female", "dateOfBirth": "1995-06-10", "admissionDate": "**", "diagnosis": "Cancer" }  Replace ** with today's date	{ "participantId": 1007, "patientName": "Arne Johansson", "gender": "Female", "dateOfBirth": "1995-06-10", "admissionDate": **, "diagnosis": "Cancer", }  ** will be replaced by today's date
	Invalid test data 1	api/patients  Sample JSON: { "dateOfBirth": "1995-06-10", "admissionDate": "**", "diagnosis": "Cancer"	{ "errorMessage": "Gender is mandatory for admission, Patient name is mandatory", "errorCode": 400 }

Go to top

ASSESSMENT INSTRUCTIONS	PROJECT INSTRUCTIONS	PROBLEM STATEMENT
Invalid test data 1	<p>api/patients</p> <p>Sample JSON:</p> <pre>{     "dateOfBirth": "1995-06-10",     "admissionDate": "**",     "diagnosis": "Cancer" }</pre> <p>Replace ** with today's date</p>	<pre>{     "errorMessage": "Gender is mandatory for admission, Patient name is mandatory",     "errorCode": 400 }</pre>
Invalid test data 2	<p>api/patients</p> <p>Sample JSON:</p> <pre>{     "patientName": "Arne2 Johansson",     "gender": "Female",     "dateOfBirth": "1995-06-10",     "admissionDate": "**",     "diagnosis": "Cancer" }</pre> <p>Replace ** with today's date</p>	<pre>{     "errorMessage": "Patient name should have only words and each word should be separated by a space",     "errorCode": 400 }</pre>
Invalid test data 3	<p>api/patients</p> <p>Sample JSON:</p> <pre>{     "patientName": "Arne Johansson" }</pre>	<pre>{     "errorMessage": "Gender should be Male, Female or Other",     "errorCode": 400 }</pre>

[Go to top](#)

file:///home/user/Desktop/user\_repo/QP/index.html

ASSESSMENT INSTRUCTIONS	PROJECT INSTRUCTIONS	PROBLEM STATEMENT
Invalid test data 3	<p>Replace ** with today's date</p> <p>api/patients</p> <p>Sample JSON:</p> <pre>{   "patientName": "Arne Johansson",   "gender": "Neutral",   "dateOfBirth": "1995-06-10",   "admissionDate": **,   "diagnosis": "Cancer" }</pre> <p>Replace ** with today's date</p>	<pre>{   "errorMessage": "Gender should be Male, Female or Others",   "errorCode": 400 }</pre>
Invalid test data 4	<p>api/patients</p> <p>Sample JSON:</p> <pre>{   "patientName": "Arne Johansson",   "gender": "Female",   "dateOfBirth": "1880-06-10",   "admissionDate": **,   "diagnosis": "Cancer" }</pre> <p>Replace ** with today's date</p>	<pre>{   "errorMessage": "Date of birth should not be a future date and the age should not be more than 100",   "errorCode": 400 }</pre>

Go to top

Mozilla Firefox

/home/user/Desktop/user\_re X + file:///home/user/Desktop/user\_repo/QP/index.html ... 🌐 ⚡ ☆

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

	Replace *** with today's date	
Invalid test data 5	<p>api/patients</p> <p>Sample JSON:</p> <pre>{   "patientName": "Arne Johansson",   "gender": "Female",   "dateOfBirth": "1995-06-10",   "admissionDate": "2022-06-14",   "diagnosis": "Cancer", }</pre>	<pre>{   "errorMessage": "Date of admission should be today's date",   "errorCode": 400 }</pre>
Invalid test data 6	<p>api/patients</p> <p>Sample JSON:</p> <pre>{   "patientName": "Arne Johansson",   "gender": "Female",   "dateOfBirth": "1995-06-10",   "admissionDate": "**", }</pre> <p>Replace ** with today's date</p>	<pre>{   "errorMessage": "Diagnosis is mandatory",   "errorCode": 400 }</pre>
Invalid test data 7	<p>api/patients</p> <p>Sample JSON:</p>	<pre>{   "errorMessage": "Diagnos should contain only numbers."}</pre> <p>Go to top</p>

Mozilla Firefox Postman workspace - lap... 17:46

Mozilla Firefox

/home/user/Desktop/user\_re X + file:///home/user/Desktop/user\_repo/QP/index.html

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

	<p>Sample JSON:</p> <pre>{   "patientName": "Arne Johansson",   "gender": "Female",   "dateOfBirth": "1995-06-10",   "admissionDate": "**", }</pre> <p>Replace ** with today's date</p>	<p>mandatory , "errorCode": 400</p> <p>}</p>
Invalid test data 7	<p>api/patients</p> <p>Sample JSON:</p> <pre>{   "patientName": "Arne Johansson",   "gender": "Female",   "dateOfBirth": "1995-06-10",   "admissionDate": "**",   "diagnosis": "Cancer,", }</pre> <p>Replace ** with today's date</p>	<p>{" "errorMessage": "Diagnosis should contain only numbers, alphabets and spaces", "errorCode": 400 }</p>

PREV

Go to top