

This is a hospital management system with validation on doctor ,patient ,nurse ,admin and pharmacy model .

Use :spring web ,Lombok and validation dependency .

Create: Doctor model ,patient model ,Nurse model, Admin model and pharmacy model.

Doctor model has : id, name, age , sex ,email , phoneNumber and position and patient.

Validation:

id : Cannot be null and length must be more than 2 digit and contain only numbers.

```
@NotNull
@Length(min = 3,message = "The ID should be more than 3 digits ")
@Pattern (regexp = "[0-9]+$", message = "id should contain only numbers")
private String id;
```

Name: Cannot be null ,length must be 3 and more character r,must contain only letter

```
@NotNull
@Length(min = 3, message = "Name should be 3 characters or more ")
@Pattern(regexp="[a-zA-Z]+$", message = "Name contain letters only")
private String name;
```

Age: Cannot be null , min 25 ,max70

```
@NotNull
@Min(value = 25 ,message = "Age of employee must be 25 or more")
@Max(value = 70 ,message = "Age of employee must be less than or equal 70")
private int age;
```

Sex: cannot be null, must be "male" or "female"

```
@NotNull
@Pattern (regexp = "(male|female)$", message = "Sex should be one of them( male OR female )")
private String sex;
```

Email: must be valid email

```
@Email
private String email;
```

PhoneNumber: must start with "05" , contain 10 numbers.

```
@Pattern(regexp = "05\\d{8}$" , message = "Number should start with 05xxxxxxx and contain 10 numbers")
private String phoneNumber;
```

Position: cannot be null ,must be (general doctor, specialist or advisor)

```
@NotNull
@Pattern (regexp = "^(general doctor |specialist |advisor )$" ,
message = "Position must be general doctor OR specialist OR advisor
only ")
private String position;
```

Patient :not empty

```
@NotEmpty
private ArrayList<PetientModel> petient;
```

DoctorController endpoints :

Get all Doctors : retrieve a list of all doctor in the system.

Add new doctor:Add new doctor to the system

Update doctor :update exist doctor information in the system by id

Delete doctor :delete doctor from the system By checking if doctor exist in the system by id

Search doctor by position : Retrieves a list of doctors based on their position (advisor ,general doctor or specialist) Ensure that the position parameter is valid (either "specialist r" or " general doctor " or " advisor")

Get Doctors by Age Range: Retrieves a list of doctors within a specified age range. minAge and maxAge are valid age values

Patien model has : id, name, age , sex ,email , phoneNumber, doctors and Allergy .

Validation:

id : Cannot be null and length must be more than 2 digit and contain only numbers

```
@NotNull
@Length(min = 3,message = "The ID should be more than 3 digits ")
@Pattern (regexp = "[0-9]+$", message = "id should contain only
numbers")
private String id;
```

Name: Cannot be null ,length must be 3 and more character r,must contain only letter

```
@NotNull
@Length(min = 3, message = "Name should be 3 characters or more ")
```

```
@Pattern(regex="^[a-zA-Z]+$", message = "Name contain letters only")
private String name;
```

Age: Cannot be null , min 10 ,max120

```
@NotNull
@Min(value = 10 ,message = "Age of employee must be 10 or more")
@Max(value = 120 ,message = "Age of employee must be less than or
equal 120")
private int age;
```

Sex: cannot be null, must be "male" or "female"

```
@NotNull
@Pattern (regex = "^(male|female)$", message = "Sex should be one of
them( male OR female )")
private String sex;
```

Email: must be valid email

```
@Email
private String email;
```

PhoneNumber: must start with "05" , contain 10 numbers.

```
@Pattern(regex = "^05\\d{8}$" , message = "Number should start with
05xxxxxxx and contain 10 numbers")
private String phoneNumber;
```

Doctor: get doctor information from doctor model and must not be empty

```
@NotEmpty
private ArrayList<DoctorModel> doctor;
```

Allergy : initially set to false.

```
@AssertFalse
private boolean Allergy;
```

PatientController endpoints :

Get all petient : retrieve a list of all patient in the system.

Add new patient : Add new patient to the system

Update patient : update exist patient information in the system by id

Delete patient r : delete patient from the system By checking if patient exist in the system by id

Search patient by doctor : Retrieves a list of patient based on their doctor .

Get patient by Age Range: Retrieves a list of patient within a specified age range. minAge and maxAge are valid age values

Get Patients with allergy : Retrieves a list of all patients who has allergy.

Get Patient by date Appointment: : Retrieves a list of patient who has appointment in specific date.

Set patient allergy: Allows a doctor to set patient allergy true if they meet certain criteria. Verify that the patient with the specified ID exists. ■ Ensure that the requester (user making the request) is a doctor ■ Confirm that the patient is currently in doctor list

- set the patient's Allergy to "true" if they meet the criteria and write type of allergy.

Pharmacy Model:has Doctor id , patient id ,name of medicine

End point :

Get patients by med :retrieve list of patient who has specific medicine

Get Doctor by med :retrieve list of doctor who write medicine for patient .

Admin model :has Doctor ,Patient ,Pharmacy ,Nurse ,name,id

Id :

Cannot be null ,Length more 3 digit and contain only numbers

```
@NotNull
@Length(min = 3,message = "The ID should be more than 3 digits ")
@Pattern (regexp = "[0-9]+$", message = "id should contain only
numbers")
private String id;
```

Name: Cannot be null ,length must be 3 and more character r,must contain only letter

```
@NotNull
@Length(min = 3, message = "Name should be 3 characters or more ")
@Pattern(regexp="[a-zA-Z]+$", message = "Name contain letters only")
private String name;
```

Doctor :cannot be null

```
@NotEmpty
private ArrayList<DoctorModel> doctor;
```

Patient :cannot be null

```
@NotEmpty
private ArrayList<PateintModel> pateint;
```

Pharmacy :cannot be null

```
@NotEmpty
private ArrayList<PharmacyModel> Pharmacy;
```

Endpoint for admin :

Display all doctors,all patients and nurses .

Add

Delete

Update

Change Doctor position: Allows an admin to change doctor/nurse position if they meet certain criteria.1\ Verify that the doctor/nurse with the specified ID exists. ▪ Ensure that the requester (user making the request) is an admin. ▪ Validate that the doctor age is at least 40 years. And nurse 30 years . Change the position if they meet the criteria.

Doctor position(advisor, general doctor ,specialist doctor)

Nurse position(nurse, senior nurse)