

Wasfi's Clinic

Description: A clinic has doctors from different specialties, and the patient can book an appointment with a specific doctor. When the examination of the patient's condition is completed, the doctor decides whether to prescribe medication for the patient or not.

System contains specified validations and endpoints.

Use: Spring web Lombok and Validation Dependencies.

Create: Doctor Model, Patient Model, Appointment Model, Medicine Model and patientMedicine Model.

Doctor Model has: doctorID, doctorName, specialtyName, phoneNumber.

Patient Model has: patientID, patientName, gender, address, age, phoneNumber, doctorID, medicine.

Appointment Model has: appointmentNumber, doctorID, patientID, appointmentDate, appointmentTime, bloodPressure, weight.

Medicine Model has: medicineName, medicineCode

patientMedicine Model has: appointmentNumber, medicineCode.

Doctor Model has: doctorID, doctorName, specialtyName, phoneNumber.

Validation:

doctorID:

Cannot be empty

Contains only 5 digits

doctorName:

Cannot be empty

Length of name must be more than 2

Contains only character

specialtyName:

Cannot be empty

Contains only character

phoneNumber:

Cannot be empty

Contains only numbers

Length of number equals 13 and start with +966

```
@NotEmpty(message="Doctor ID should not be empty")
@Pattern(regexp = "\\d{5}$")
private String doctorID;
@NotEmpty(message = "Doctor name should not be empty")
@Size(min=3,message="Name length must be more than 3 characters")
@Pattern(regexp="[a-zA-Z]*$")
private String doctorName;
@NotEmpty(message = "Specialty name should not be empty")
@Pattern(regexp="[a-zA-Z]*$")
private String specialtyName;
@NotEmpty(message = "Phone number should not be empty")
@Pattern(regexp = "(\\+966)\\d{13}$")
private String phoneNumber;
```

Patient Model has: patientID, patientName, gender, address, age, phoneNumber, doctorID, medicine.

Validation:

patientID:

Cannot be empty

Contains only 10 digits

patientName:

Cannot be empty

Length of name must be more than 2

Contains only character

gender:

Cannot be empty

Must be either Male or Female only

address:

Cannot be empty

Must contain only alphanumeric characters, spaces, and hyphens

age:

Cannot be empty

Must contain only numbers

phoneNumber:

Cannot be empty

Contains only numbers

Length of number equals 13 and start with +966

doctorID:

Cannot be empty

Contains only 5 digits

medicine:

Cannot be empty

Must be initially set to false.

```

public class Patient {
    @NotEmpty(message="Patient ID should not be empty")
    @Pattern(regexp = "\\d{10}$")
    private String patientID;
    @NotEmpty(message = "Patient name should not be empty")
    @Size(min=3,message="Name length must be more than 3 characters")
    @Pattern(regexp="[a-zA-Z]*$")
    private String patientName;
    @NotEmpty(message = "Gender should not be empty")
    @Pattern(regexp = "(Male|Female)$",message="Only two options(Male or Female)")
    private Boolean gender;
    @NotEmpty(message="address should not be empty")
    @Pattern(regexp = "[a-zA-Z0-9\\s-]*$", message = "Street must contain only alphanumeric characters, spaces, and hyphens")
    private String address;
    @NotNull(message="Age should not be empty")
    @Positive(message = "Age must be a positive number")
    private int age;
    @NotEmpty(message = "Phone number should not be empty")
    @Pattern(regexp = "(\\+966)\\d{13}$")
    private String phoneNumber;
    @NotEmpty(message="Doctor ID should not be empty")
    @Pattern(regexp = "\\d{5}$")
    private String doctorID;
    @NotEmpty(message="Medicine should not be empty")
    @AssertFalse
    private Boolean medicine;
}

```

Appointment Model has: appointmentNumber, doctorID,patientID, appointmentDate, appointmentTime, bloodType, weight, bloodPressure.

Validation:

appointmentNumber:

Cannot be empty

Must contain only numbers

doctorID:

Cannot be empty

Contains only 5 digits

patientID:

Cannot be empty

Contains only 10 digits

appointmentDate:

Cannot be empty

Must be a date in the present or the past.

Date must be in the format yyyy-MM-dd

appointmentTime:

Cannot be empty

Time must be in the format HH:mm

bloodType:

Cannot be empty

Must be A+, A-, B+, B-, AB+, AB-, O+, or O-

weight:

Cannot be empty

Must be only numbers

bloodPressure:

Cannot be empty

must be in the format ddd/dd and only numbers

```

public class Appointment {
    @NotEmpty(message="Appointment number should not be empty")
    @Positive(message = "Appointment number should be positive number")
    @Min(1)
    private int appointmentNumber;
    @NotEmpty(message="Doctor ID should not be empty")
    @Pattern(regexp = "^\\d{5}$")
    private String doctorID;
    @NotEmpty(message="Patient ID should not be empty")
    @Pattern(regexp = "^\\d{10}$")
    private String patientID;
    @NotEmpty(message="Appointment date should not be empty")
    @PastOrPresent()
    @JsonFormat(pattern="yyyy-MM-dd")
    private LocalDate appointmentDate;
    @NotEmpty(message="Appointment time should not be empty")
    @Pattern(regexp = "^([01]\\d|2[0-3]):([0-5]\\d)$", message = "Appointment time must be in the format HH:mm")
    private LocalTime appointmentTime;
}

```

```

@NotEmpty(message = "Blood type should not be empty")
@Pattern(regexp = "^([A|B|AB|O][+-])$", message = "Blood type must be in the format A+, A-, B+, B-, AB+, AB-, O+, or O-")
private String bloodType;
@NotNull(message= "Weight should not be empty")
@Positive(message= "Weight should not be negative number")
private int weight;
@NotEmpty(message="Blood pressure should not be empty")
@Pattern(regexp = "^(1[0-9][0-9]|[1-9][0-9]|[1-9])/(((1[0-9][0-9]|[1-9]))$)", message = "Blood pressure must be in the format ddd/dd (e.g., 120/80)")
private int bloodPressure;

```

Medicine Model has: medicineName, medicineCode

Validation:

medicineName:

Cannot be empty

Length must be more than 3

Only contains a character

medicineCode:

Cannot be empty

Length must be 4

```
public class Medicine {  
    @NotEmpty(message="Medicine name should not be empty")  
    @Pattern(regexp="^[a-zA-Z]*$")  
    @Size(min=5)  
    private String medicineName;  
    @NotEmpty(message="Medicine code should not be empty")  
    @Pattern(regexp = "^\\d{4}$")  
    private String medicineCode;  
}
```

patientMedicine Model has: appointmentNumber, medicineCode.

Validation:

medicineCode:

Cannot be empty

Length must be 4

appointmentNumber:

Cannot be empty

Must contain only numbers

```
public class patientMedicine {  
  
    @NotEmpty(message="Medicine code should not be empty")  
    @Pattern(regexp = "^\\d{4}$")  
    private String medicineCode;  
  
    @NotEmpty(message="Appointment number should not be empty")  
    @Positive(message = "Appointment number should be positive number")  
    @Min(1)  
    private int appointmentNumber;  
}
```