

DOMAIN NAME

HEALTHCARE

TEST NUMBER

3



TEST CASE ID	Model data	Model data	validation checklist	Potential consequences of failing	Java Implementation
TC_User_01	Patient	Full name	User must enter his full name mainly first and last. Name cannot be empty	Incorrect patient name can lead to misdiagnosis or errors in medical records.	@NotEmpty(message = "Full name cannot be empty") @Pattern(regexp = "^[A-Za-z]+ [A-Za-z]+\$", message = "first and last name is required. No numbers allow") private String fullName;
TC_User_02		Patient ID	ID must be longer than 10 characters and cannot be null.	Incorrect ID, which can lead to patient mix-ups or incorrect treatments and medications.	@NotEmpty(message = "Enter Patient ID") @Size(min = 10, message = "Patient ID must be longer than 10 characters") private String patientID;
TC_User_03		Date of Birth	Date format must be YYYY-MM-DD. The date should be in the past (age 0-100).	Incorrect date of birth, which can result in mixing up patients or administering the wrong treatments or medications	@NotNull(message = "Date of birth cannot be null") @Past(message = "Date of birth must be in the past") @DateTimeFormat(pattern = "yyyy-MM-dd") private LocalDate dateOfBirth;
TC_User_04		phone number	Must follow the format 0512345678; no letters. cannot be empty.	Missed communications, like appointment reminders or emergency alerts.	@NotEmpty(message = "Phone number cannot be empty") @Size(min = 10, max = 10, message = "phone number must be 10 digit ") @Pattern(regexp = "^(05){1}[0-9]{7}\$", message = "Phone number must be from Saudi Arabia") private String phoneNumber;
TC_User_05		Emergency phone number	Must follow the format 0512345678; no letters. cannot be empty.	Failure to contact a family member during a medical emergency.	@NotEmpty(message = "Emergency phone number cannot be empty") @Size(min = 10, max = 10, message = "Emergency phone number must be 10 digit ") @Pattern(regexp = "^(05){1}[0-9]{7}\$", message = "Emergency phone number must be from Saudi Arabia") private String emergencyPhoneNumber;
TC_User_06		Email Address	Must be in the format example@domain.com. cannot be empty.	Unsuccessful delivery of electronic medical records or appointment confirmations.	@NotEmpty(message = "Enter your email") @Email(message = "Email is not valid", regexp = "^[a-zA-Z0-9_!#\$%&'*/=?`{}~^.-]+@[a-zA-Z0-9.-]+.\$") private String email;
TC_Appointment_01	Appointment	Appointment Date	Must be in the future	Failure to attend appointments	@NotNull(message = "Appointment date of birth cannot be null") @Future(message = "Appointment date must be in the future") private LocalDateTime appointmentDate;
TC_Appointment_02		Doctor ID	ID must be longer than 10 characters and cannot be null.	Patients being assigned to non-existent or incorrect doctors, which leads delayed treatment	@NotEmpty(message = "Enter doctor ID") @Size(min = 10, message = "Doctor ID must be longer than 10 characters") private String doctorID;
TC_Appointment_03		Patient ID	ID must be longer than 10 characters and cannot be null.	Mismatches between patient records and appointment schedules can lead to treatment delays.	@NotEmpty(message = "Enter Patient ID") @Size(min = 10, message = "Patient ID must be longer than 10 characters") private String patientID;
TC_Appointment_04		Appointment Type	Must specify if the appointment is in-person or online. cannot be empty.	Patients being scheduled incorrectly lead to treatment delays.	@NotEmpty(message = "Appointment type cannot be empty") @Pattern(regexp = "(in-person online)\$", message = "Appointment type must be either in-person or online only") private String appointmentType;
TC_Staff_01		Staff Name	User must enter his full name mainly first and last. Name cannot be empty	Incorrect staff name, resulting in scheduling mistakes, or wrong treatment assignments	@NotEmpty(message = "staff name cannot be empty") @Pattern(regexp = "^[A-Za-z]+ [A-Za-z]+\$", message = "first and last name is required. No numbers allow") private String staffName;
TC_Staff_02		Role	Must be one of the predefined roles Doctor, Nurse, Technician and Receptionist. cannot be empty.	Incorrect role leads to scheduling mistakes, or treatment delays.	@NotEmpty(message = "Role cannot be empty") @Pattern(regexp = "(Doctor Nurse Technician Receptionist)\$") private String role;
TC_Staff_03		Staff ID	ID must be longer than 10 characters and cannot be null.	Problems in tracking staff performance and payroll, as well as the inability to associate staff with the correct records	@NotEmpty(message = "Enter Staff ID") @Size(min = 10, message = "Staff ID must be longer than 2 characters") private String staffID;
TC_Staff_04		Email Address	Must be in the format example@domain.com. cannot be empty.	Missed communications, such as important updates, shift schedules, or emergency contact information	@NotEmpty(message = "Enter your email") @Email(message = "Email is not valid", regexp = "^[a-zA-Z0-9_!#\$%&'*/=?`{}~^.-]+@[a-zA-Z0-9.-]+.\$") private String staffEmail;

TC_Staff_05	Staff	phone number	Must follow the format 0512345678; no letters. cannot be empty.	delayed communications, such as important updates, shift schedules, or emergency contact information	@NotEmpty(message = "Phone number cannot be empty") @Size(min = 10 , max = 10 , message = "phone number must be 10 digit ") @Pattern(regexp = "^{(05)}{([0-9]{7})}\$", message = "Phone number must be from Saudi Arabia") private String phoneNumber;
TC_Staff_06		Department	Must be one of the predefined roles Cardiology, Pediatrics and Emergency. cannot be empty.	Incorrect staffing of departments, resulting in efficiency drop and potential misdiagnosis or delayed care."	@NotEmpty(message = "Department cannot be empty") @Pattern(regexp = "^(Cardiology Pediatrics Emergency)\$") private String department;
TC_Staff_07		Date of Joining	Must be a past or present date	mistakes in tracking staff seniority, along with potential discrepancies in payroll or benefits calculations	@PastOrPresent(message = "Date of joining must be in the past or present") private LocalDate dateOfJoining
TC_Staff_08		Salary	Must be a positive number greater than 6000. cannot be empty.	wrong payments	@NotNull(message = "salary cannot be empty") @Positive(message = "Salary must be positive") @Min(value = 6000) private double salary;

