

# **Business Analysis Documentation**

## **Project:**

HealthCare System – Symptom2Disease

## **Prepared by:**

Mohammed Alherz

## **Version:**

1.0

**Date:** July 22, 2025

**Code:** Java

;)

## **Project Description:**

**Symptom2Disease** is a convenient web application designed to help patients get medical guidance without the immediate need to visit a hospital.

---

## **Here's how it works in simple terms:**

Easy Medical Consultation at Your Fingertips Instead of waiting for a hospital appointment, patients can access this website from anywhere and describe their symptoms in plain language. The application features an intelligent AI assistant that analyzes the symptoms you provide and offers potential diagnoses or health insights.

---

## **How the Process Works:**

1. **Share Your Symptoms** – Simply type what you're experiencing (headache, fever, stomach pain, etc.) into the website
  2. **AI Analysis** – The built-in AI reviews your symptoms and provides possible explanations or conditions that might match what you're feeling
  3. **Professional Verification** – A qualified doctor reviews the AI's assessment to ensure accuracy and reliability
  4. **Get Results Quickly** – Receive guidance much faster than traditional hospital visits
-

### **Additional Services:**

- **Schedule Appointments** – If you need further consultation, you can easily book an appointment with a doctor directly through the platform
- **Direct Doctor Communication** – Chat directly with a real doctor through the platform for personalized advice
- **Submit Feedback and Ratings** – After your consultation, you can provide feedback and rate the doctor based on your experience to help improve service quality
- **Time-Saving** – No need to travel or wait in long hospital queues for initial consultations
- **Accessible Healthcare** – Get medical insights from the comfort of your home
- **Professional Oversight** – All AI recommendations are verified by licensed medical professionals

---

This application essentially brings preliminary medical consultation to your device, making healthcare more accessible while maintaining professional medical standards through doctor verification. Whether you need quick symptom assessment, want to schedule a proper appointment, or give feedback on your experience — everything can be managed through one convenient platform.

## Validation Checklist for Symptom2Disease System

---

### Entity: User (Patient)

#### 1. **userId**

- Validation:
  - Cannot be null.
  - The length must be greater than 1 digit.
- Annotation:

@NotNull(message = "User ID cannot be null")

@Min(value = 10, message = "User ID must be at least two digits")

#### 2. **fullName**

- Validation:
  - Cannot be empty.
  - The length must be greater than 3 characters.
- Annotation:

@NotEmpty(message = "Full name is required")

@Size(min = 4, message = "Full name must be more than 3 characters")

### 3. email

- Validation:
  - Cannot be null.
  - Must be a valid email format.
- Annotation:

`@NotEmpty(message = "Email is required")`

`@Email(message = "Email format is invalid")`

### 4. password

- Validation:
  - Cannot be null.
  - The length must be at least 8 characters long.
- Annotation:

`@NotEmpty(message = "Password is required")`

`@Pattern(regex = "^(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z])(?=.*[!@#&()–{};:,?/*~$^+=<>]).{8,20}$", message = "Password must contain at least one digit, one lowercase, one uppercase, one special character and be 8-20 characters long")`

### 5. gender

- Validation:
  - Cannot be null.
  - Must be one of: Male, Female.
- Annotation:

@NotEmpty(message = "Gender is required")

@Pattern(regexp = "^(male|female)\$",message = "gender must be either male or female")

## **6. dateOfBirth**

- Validation:
  - Cannot be null.
  - Must be a past date.
- Annotation:

@Past(message = "birthdate must be in the past")

@JsonFormat(pattern = "yyyy-MM-dd")

---

## **Entity: Doctor**

### **1. doctorId**

- Validation:
  - Cannot be null.
  - The length Must be greater than 1 digit.
- Annotation:

@NotNull(message = "Doctor ID cannot be null")

@Min(value = 10, message = "Doctor ID must be at least two digits")

### **2. fullName**

- Validation:

- Cannot be null.
- The length Must be greater than 3 characters.
- Annotation:

@NotEmpty(message = "Doctor name is required")

@Size(min = 4, message = "Doctor name must be more than 3 characters")

### **3. email**

- Validation:
  - Must be a valid email.
- Annotation:

@NotEmpty(message = "Doctor email is required")

@Email(message = "Doctor email format is invalid")

### **4. specialization**

- Validation:
  - Cannot be null.
  - The length must be more than 3 characters
- Annotation:

@NotEmpty(message = "Specialization is required")

@Size(min = 4, message = "Doctor name must be more than 3 characters")

### **5. licenseNumber**

- Validation:

- Cannot be null.
- Must be alphanumeric and 6+ characters.
- Annotation:

`@NotBlank(message = "License number is required")`

`@Pattern(regexp = "[a-zA-Z0-9]{6,}$", message = "License number must be at least 6 alphanumeric characters")`

---

## **Entity: Appointment**

### **1. appointmentId**

- Validation:
  - Cannot be null.
  - Length must be more than 2 characters
- Annotation:

`@NotNull(message = "Appointment ID is required")`

`@Min(value = 10, message = "Appointment ID must be at least two digits")`

### **2. appointmentDate**

- Validation:
  - Cannot be null.
  - Must be a future date.
- Annotation:



@NotEmpty(message = "Appointment date is required")

@Future(message = "Appointment date must be in the future")

@JsonFormat(pattern = "yyyy-MM-dd HH:mm")

▪ **status**

- Validation:
  - Must be one of: Pending, Confirmed, Cancelled.
- Annotation:

@NotEmpty(message = "Status is required")

@Pattern(regexp = "^(canceled|confirmed|pending)\$", message = "status must be either canceled, confirmed or pending")

---

## **Entity: Chat Message**

### **1. messageId**

- Validation:
  - Cannot be null.
  - Length must be more than 2 characters
- Annotation:

@NotNull(message = "Message ID is required")

@Min(value = 10, message = "Doctor ID must be at least two digits")

## **2. messageText**

- Validation:
  - Cannot be null.
  - The length must be greater than 2 characters.
- Annotation:

@NotEmpty(message = "Message text cannot be empty")

@Size(min = 3, message = "Message must be at least 3 characters")

## **3. senderId and receiverId**

- Validation:
  - Cannot be null.
  - Should be valid senderId and receiverId
- Annotation:

@NotNull(message = "Sender ID is required")

@Min(value = 10, message = "Sender ID must be at least two digits")

@NotNull(message = "Receiver ID is required")

@Min(value = 10, message = "Receiver ID must be at least two digits")

---

## **Entity: Feedback**

### **▪ feedbackId**

- Validation:
  - Cannot be null.
  - The length must be more than 1 digit

- Annotation:

@NotNull(message = "Feedback ID is required")

@Min(value = 10, message = "Feedback Id must be at least two digits")

#### ▪ **rating**

- Validation:
  - Must be between 1 and 5.

- Annotation:

@Min(value = 1, message = "Rating must be at least 1")

@Max(value = 5, message = "Rating cannot be more than 5")

#### ▪ **comment**

- Validation:
  - Optional, but if present must be at least 5 characters.

- Annotation:

@Size(min = 5, message = "Comment must be at least 5 characters", groups =  
Optional.class)