**Test Specifications Document:**

**Send Message () Method**

**TestCase: sendMessage\_WithAllFields\_UpdatesMainMenuWithFormattedMessage**

* Verifies that a message with all fields (recipient, patient, body) is correctly formatted and sent.
  1. Set recipient, patient, and message text.
  2. Call sendMessage().
* Expected:
  1. MainMenu receivesformattedmessage:  
     "To: doctor@hospital.com | Please review test results. (Regarding Patient: John Smith)"
  2. wasMessageUpdated flag is true.

**TestCase: sendMessage\_WithoutPatientField\_UpdatesMainMenuWithoutPatientInf**

* Ensures messages without a patient field are correctly formatted.
  1. Set recipient and message (no patient).
  2. Call sendMessage().
* Expected:
  1. MainMenu receives message without patient info:  
     "To: nurse@clinic.com | Medication refill needed."

**Test Case: sendMessage\_WithEmptyRecipient\_ShowsErrorDialoge**

* Validates the empty recipient fields trigger an error.
  1. Leave recipient empty, set message.
  2. Simulate "Send" button click.
* Expected:
  1. MainMenu is not updated (wasMessageUpdated = false).
  2. Window remains open (isVisible() = true).

**Test Case: sendMessage\_WithEmptyMessageBody\_ShowsErrorDialoge**

* Ensures that empty message bodies are rejected.
  1. Set recipient but leave message empty.
  2. Call sendMessage().
* Expected:
  1. MainMenu is not updated.
  2. Window stays open.

**Test Case: uiComponents\_AreCorrectlyInitialized**

* Verify all UI components are properly created.
  1. Initialize MessagingSystem.
* Expected:
  1. recipientField, patientField, messageBody, and sendButton are not null.

**Test Case: windowCloses\_AfterSuccessfulMessageSend**

* Ensures that the window closes after a successful send.
  1. Set valid recipient and message.
  2. Call sendMessage().
* Expected:
  1. Window closes (isVisible() = false).

**AssignNewDoctorTest**

**Test Cases:**

**Test Case: constructor\_InitializesUIComponents**

* Verifies that the UI components are properly initialized.
  1. Create an AssignNewDoctor instance.
* **Expected**:
  1. doctorList (JComboBox) is initialized.
  2. doctorIDs list is initialized.
  3. Window title is set to **"Assign New Doctor"**.

**Test Case: loadDoctors\_PopulatesComboBoxFromDatabase**

* Ensures that doctors are correctly loaded from the database into the dropdown.
  1. Initialize AssignNewDoctor (automatically loads doctors).
* **Expected**:
  1. doctorList contains **"Dr. Smith"** and **"Dr. Johnson"**.
  2. doctorIDs contains corresponding IDs (**101, 102**).

**Test Case: assignDoctor\_WithValidSelection\_UpdatesBookingRecord**

* Verifies that selecting a doctor updates the booking record in the database.
  1. Select **"Dr. Smith"** (ID 101) from the dropdown.
  2. Call assignDoctor().
* **Expected**:
  1. Database updates Bookings table with doctorID = 101 for bookingNo = 123.
  2. Success message: **"Doctor successfully assigned."**

**Test Case: assignDoctor\_WithNoSelection\_ShowsErrorDialog**

* Validates error handling when no doctor is selected.
  1. Leave the dropdown unselected (selectedIndex = -1).
  2. Call assignDoctor ().
* **Expected**:
  1. Error message: **"Please select a doctor."**

**Test Case: windowCloses\_AfterSuccessfulAssignment**

* Ensures that the window closes after a successful assignment.
* **Steps**:
  1. Select a doctor and call assignDoctor ().
* **Expected**:
  1. Window closes (isVisible () = false).

**EditVisitDetailsTest**

**Test Case: loadExistingDetails\_LoadsDataCorrectly**  
• Tests loading of visit/prescription data

1. Initialize with test booking ID

Expected

* notesField contains "Initial checkup notes"
* medicationField contains "Ibuprofen"
* All prescription fields populated correctly

**Test Case: loadExistingDetails\_HandlesMissingVisit**  
• Tests error handling for invalid bookings

1. Initialize with non-existent booking ID

Expected

* Error message: "No visit found for the given Booking ID"
* Window closes automatically

**Test Case: updateDetails\_UpdatesDatabaseSuccessfully**  
• Verifies successful data updates

1. Edit fields and submit

Expected

* Visits and Prescriptions tables update in database
* Success message appears
* Window closes

**Test Case: updateDetails\_ValidatesEmptyFields**  
• Tests required field validation

1. Submit with empty fields

Expected

* Error message: "All fields must be filled!"
* No database updates occur
* Window remains open

**Test Case: sendConfirmationMessage\_ShowsDialog**  
• Tests confirmation messaging

1. Complete successful update

Expected

* Confirmation message: "Confirmation sent to doctor and patient"