BANGLADESH UNIVERSITY OF PROFESSIONALS

FACULTY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (CSE)

Test Suite

Submitted By:

Name	Roll
Mohaiminul Raju	2252421020
Shamoyeta Mourin Mouly	2252421036
Latifa Nishat Nishi	2252421062
Tahsina Tabassum Roza	2252421084
Raiyan Sarwar	2252421096

Section : B

Semester : 6th

Course Name : Software Engineering Laboratory

Course Code : CSE-3206

Date of Submission: 30.05.2025

_____

Signature of Teacher

Test Suite

Below is a comprehensive test suite documentation covering all 10 major features of the Hospital Management System. Each feature's test cases are designed to verify that the functionality meets its specifications under normal, edge, and erroneous conditions. In the following sections, you will find at least five (and in some cases six) detailed test cases for each feature, including objectives, preconditions, step-by-step procedures, and expected outcomes.

1. Appointment Booking

Test Case AB-1: Valid Appointment Booking

- **Objective:** Verify that a patient can successfully book an appointment when all required information is provided and the selected time slot is available.
- **Preconditions:**
  - Patient is registered and logged in.
  - The chosen doctor and time slot are available.
- **Test Steps:**
  1. Log in using valid patient credentials.
  2. Navigate to the appointment booking module.
  3. Select an available doctor.
  4. Choose an available date and time slot.
  5. Fill in all required fields (e.g., reason for visit, contact details).
  6. Submit the booking request.
- **Expected Outcome:**
  - A confirmation page with appointment details is displayed.
  - A confirmation email or notification is sent to the patient.
  - The selected time slot is marked as booked in the schedule.

Test Case AB-2: Appointment Booking with Missing Required Fields

- **Objective:** Validate that the system handles cases where mandatory information is omitted.
- **Preconditions:**
  - Patient is logged in and accesses the appointment booking form.
- **Test Steps:**

1. Open the appointment booking form.
2. Leave one or more required fields (e.g., appointment date/time, contact details) blank.
3. Attempt to submit the form.

- **Expected Outcome:**
  - Submission is prevented with error messages indicating which field(s) are missing.
  - No appointment is created.

Test Case AB-3: Booking When the Selected Time Slot Is Unavailable

- **Objective:** Ensure that the system prevents bookings for a time slot that is fully reserved.
- **Preconditions:**
  - The selected time slot is fully booked.
- **Test Steps:**
  1. Log in as a patient.
  2. Navigate to the appointment booking module.
  3. Select a doctor and choose a time slot that is already booked.
  4. Complete the booking form and submit.
- **Expected Outcome:**
  - The system notifies the patient that the time slot is unavailable and prompts for an alternative.

Test Case AB-4: Modification of an Existing Appointment

- **Objective:** Verify that a patient can update appointment details (time, date, or doctor) and that changes are properly processed.
- **Preconditions:**
  - A valid appointment exists in the patient's history.
  - Patient is logged in.
- **Test Steps:**
  1. Log in and access the appointment history.
  2. Select the appointment to modify.
  3. Edit the desired fields (e.g., new time or doctor).

4. Submit the modifications.

- **Expected Outcome:**
    - The updated appointment is validated, applied, and confirmed via notification.

Test Case AB-5: Appointment Cancellation

- **Objective:** Confirm that a patient can cancel an existing appointment with proper system updates.
- **Preconditions:**
    - A cancelable appointment exists.
    - Patient is logged in.
- **Test Steps:**
    1. Log in and navigate to appointment history.
    2. Select the appointment to cancel.
    3. Initiate and confirm cancellation.
- **Expected Outcome:**
    - The appointment is marked as cancelled with the time slot released for rebooking.
    - A cancellation confirmation is sent.

Test Case AB-6: Prevention of Overlapping Appointments

- **Objective:** Ensure that the system prevents scheduling appointments that overlap with an existing one.
- **Preconditions:**
    - The patient already has an appointment.
    - An attempt is made to book a second appointment with overlapping timing.
- **Test Steps:**
    1. Log in as a patient.
    2. Attempt to create a new appointment that overlaps with the existing one.
    3. Submit the booking request.
- **Expected Outcome:**

- A warning indicates the scheduling conflict, and the new booking is not processed until resolved.

2. Patient History

Test Case PH-1: Retrieving Patient History with Date Range Filtering

- **Objective:** Validate that users can filter patient history records by specifying a date range and that the system returns only records within that period.
- **Preconditions:**
  - Multiple history records exist for a patient, spanning various dates.
- **Test Steps:**
  1. Log in with appropriate credentials (patient or authorized staff).
  2. Navigate to the Patient History module.
  3. Set a start and end date in the filter options.
  4. Execute the search/filter.
- **Expected Outcome:**
  - Only the records falling within the specified date range are displayed, correctly ordered by date.

Test Case PH-2: Unauthorized Access Attempt for Patient History Data

- **Objective:** Ensure that the system restricts patient history access to authorized users only.
- **Preconditions:**
  - A patient history exists; a user with insufficient privileges attempts access.
- **Test Steps:**
  1. Log in using credentials that lack permission (e.g., an unverified user or one with a restricted role).
  2. Attempt to access a patient's history.
- **Expected Outcome:**
  - The system denies access, displays a "Permission Denied" or similar error message, and logs the attempt for security auditing.

Test Case PH-3: Exporting Patient History in Multiple Formats

- **Objective:** Confirm that an authorized user can export patient history records in different formats (such as PDF and CSV) and that the output accurately reflects the on-screen data.
- **Preconditions:**
    - The patient has multiple history records.
    - The user has export privileges.
- **Test Steps:**
    1. Log in and navigate to the Patient History section.
    2. Select the export option and choose PDF format. Save or view the output.
    3. Repeat the export process for CSV format.
- **Expected Outcome:**
    - Both exported files are generated successfully and contain all relevant history data in the correct format.

Test Case PH-4: Handling New or Empty Patient History

- **Objective:** Validate that the system provides an appropriate message or prompt when no history records exist for a patient.
- **Preconditions:**
    - A new patient record exists with no historical entries.
- **Test Steps:**
    1. Log in and open the Patient History module for the new patient.
    2. Observe how the system handles the empty state.
- **Expected Outcome:**
    - A friendly message is displayed (e.g., “No records found. Please add a new entry.”), along with suggestions or links to help begin the record-keeping process.

Test Case PH-5: Editing an Existing Patient History Record

- **Objective:** Validate that an authorized user can modify an existing patient history record to correct or update information.
- **Preconditions:**
    - The patient history contains at least one record that is eligible for editing.
    - The user (doctor/authorized staff) is logged in with editing privileges.

- **Test Steps:**
    1. Log in and access the specific patient history record.
    2. Select the record to be edited and click “Edit.”
    3. Modify the necessary fields (e.g., update diagnosis details or treatment information).
    4. Submit the changes.
- **Expected Outcome:**
    - The updated record is saved, and the changes are immediately reflected in the patient history.
    - An audit log entry is created (if applicable) to track the update.

3. Staff Management

Test Case SM-1: Valid Staff Member Addition

- **Objective:** Verify that an admin can add a new staff member with complete and valid details.
- **Preconditions:**
    - An admin is logged in with access to the Staff Management module.
- **Test Steps:**
    1. Navigate to the Staff Management panel.
    2. Click “Add New Staff.”
    3. Enter all required details (name, department, position, contact info).
    4. Submit the form.
- **Expected Outcome:**
    - A new staff member is added to the system, the list updates, and an audit log is recorded.

Test Case SM-2: Staff Role Modification

- **Objective:** Ensure that admins can modify the role or department of an existing staff member.
- **Preconditions:**
    - Staff member exists in the system.
    - Admin is logged in.
- **Test Steps:**

1. In the Staff Management module, select an existing staff member.
2. Change the role or department.
3. Confirm and submit the changes.

- **Expected Outcome:**
  - The staff member's details are updated and a confirmation is logged.

Test Case SM-3: Invalid Staff Data Submission

- **Objective:** Confirm that the system rejects incomplete or invalid staff data.
- **Preconditions:**
  - Admin is on the "Add New Staff" form.
- **Test Steps:**
  1. Leave required fields blank or enter invalid data (e.g., numeric values in the name field).
  2. Attempt submission.
- **Expected Outcome:**
  - Error messages are displayed and no record is created until corrections are made.

Test Case SM-4: Deletion of a Staff Member

- **Objective:** Verify that an admin can remove a staff record from the system as needed.
- **Preconditions:**
  - A staff member exists.
  - Admin is logged in.
- **Test Steps:**
  1. Navigate to the Staff Management module.
  2. Select an existing staff member.
  3. Click "Delete" and confirm the action.
- **Expected Outcome:**
  - The staff member is removed, and the deletion is logged.

Test Case SM-5: Staff Search and Filter Performance

- **Objective:** Validate that the search function within the Staff Management module works efficiently.

- **Preconditions:**
    - Multiple staff records exist.
    - Admin is logged in.
- **Test Steps:**
    1. Use the search functionality to filter staff based on specific criteria (name or department).
- **Expected Outcome:**
    - Matching staff records are displayed promptly and accurately.

4. Room Allocation

Test Case RA-1: Valid Room Allocation

- **Objective:** Confirm that an available room can be correctly assigned to a patient.
- **Preconditions:**
    - A patient requires room allocation and at least one room is available.
    - Administrator or receptionist is logged in.
- **Test Steps:**
    1. Navigate to the Room Allocation module.
    2. Select an available room.
    3. Assign the room to the patient.
    4. Submit the allocation request.
- **Expected Outcome:**
    - The room's status is updated to "occupied," and the patient's record reflects the assignment.

Test Case RA-2: Room Unavailability Handling

- **Objective:** Ensure the system prevents assigning a room that is already occupied.
- **Preconditions:**
    - A specific room is already occupied.
- **Test Steps:**
    1. Attempt to allocate the same room to another patient.

- **Expected Outcome:**
    - An error message indicates the room is unavailable; allocation is halted.

Test Case RA-3: Updating Room Status After Discharge

- **Objective:** Verify that discharging a patient updates the room status to "available."
- **Preconditions:**
    - A patient is occupying a room.
- **Test Steps:**
    1. Mark the patient as discharged.
    2. Update the room allocation status.
- **Expected Outcome:**
    - The room status changes from "occupied" to "available."

Test Case RA-4: Room Cleaning Status Update

- **Objective:** Ensure that housekeeping can update a room's cleanliness status.
- **Preconditions:**
    - A room is marked as occupied or under maintenance.
- **Test Steps:**
    1. A housekeeping user logs into the Room Allocation module.
    2. Update the status to "clean" after servicing the room.
- **Expected Outcome:**
    - The room status is updated and notifications are sent to reception if needed.

Test Case RA-5: Special Room Request Handling (e.g., ICU)

- **Objective:** Verify that requests for special rooms (e.g., ICU) follow prioritized allocation logic.
- **Preconditions:**
    - A patient requires admission to a special room and such rooms exist with priority rules.
- **Test Steps:**
    1. Initiate a room allocation request specifying ICU or special requirements.
    2. Submit the request.

- **Expected Outcome:**
    - The system prioritizes and assigns an ICU room according to availability and workflow rules.

5. Medication Store

Test Case MS-1: Valid Medication Stock Update

- **Objective:** Confirm that medication restocking updates the inventory accurately.
- **Preconditions:**
    - A medication exists in inventory.
    - A pharmacist is logged in.
- **Test Steps:**
    1. Navigate to the Medication Inventory module.
    2. Select a medication and enter the restocked quantity.
    3. Submit the update.
- **Expected Outcome:**
    - The medication quantity is updated, and a confirmation message is displayed.

Test Case MS-2: Low Stock Alert Generation

- **Objective:** Ensure that an alert is triggered when stock for a medication falls below its threshold.
- **Preconditions:**
    - A medication's quantity is below the minimum threshold.
- **Test Steps:**
    1. Verify the inventory dashboard for low-stock alerts.
- **Expected Outcome:**
    - An alert or notification highlights the low stock, prompting reorder.

Test Case MS-3: Prevent Dispensing Over Available Stock

- **Objective:** Validate that dispensing requests exceeding available stock are rejected.
- **Preconditions:**
    - Medication exists with a known quantity.

- **Test Steps:**
    1. Initiate a medicine dispensing operation requesting a quantity greater than available.
- **Expected Outcome:**
    - The system rejects the request with an error message about insufficient stock.

Test Case MS-4: Barcode Scanning Integration

- **Objective:** Confirm that the barcode scanning feature accurately identifies medication and updates stock.
- **Preconditions:**
    - Barcode scanning is enabled and medications have valid barcodes.
- **Test Steps:**
    1. Scan the barcode during dispensing or inventory update.
- **Expected Outcome:**
    - The medication is correctly identified and record updates occur as expected.

Test Case MS-5: Medication Inventory Report Generation

- **Objective:** Verify that the system can generate downloadable inventory reports showing current stock and usage trends.
- **Preconditions:**
    - Multiple medication records exist.
    - The pharmacist or admin is logged in.
- **Test Steps:**
    1. Click on the “Generate Report” option in the inventory module.
    2. Select parameters such as date range or category.
- **Expected Outcome:**
    - A report (PDF/Excel) is generated that accurately reflects inventory details.

6. Diagnostic Services

Test Case DS-1: Schedule Diagnostic Test

- **Objective:** Verify that a diagnostic test can be scheduled for a patient by a doctor.
- **Preconditions:**
  - Patient record exists and doctor is logged in.
- **Test Steps:**
  1. Access the Diagnostic Services module.
  2. Select a patient and test type along with a designated date/time.
  3. Submit the test order.
- **Expected Outcome:**
  - The test is scheduled, a test ID is generated, and the patient is notified.

Test Case DS-2: Successful Upload of Diagnostic Report

- **Objective:** Ensure that lab technicians can upload diagnostic reports linked to the patient's test order.
- **Preconditions:**
  - A diagnostic test is scheduled.
  - Lab technician is logged in.
- **Test Steps:**
  1. Locate the scheduled test in the Diagnostic Services module.
  2. Upload the diagnostic report file.
- **Expected Outcome:**
  - The report is attached to the test order, a timestamp is recorded, and a confirmation is sent.

Test Case DS-3: Test Status Tracking

- **Objective:** Verify that the test status updates (Scheduled, In Progress, Completed) are correctly displayed in real time.
- **Preconditions:**
  - A diagnostic test order is active.
- **Test Steps:**
  1. Monitor the test order status as it changes through its lifecycle.

- **Expected Outcome:**
    - The status is updated in real time with appropriate notifications to the patient and doctor.

Test Case DS-4: Handling Invalid File Format for Report Upload

- **Objective:** Validate that only supported file formats can be used to upload diagnostic reports.
- **Preconditions:**
    - A lab technician attempts to upload a diagnostic report.
- **Test Steps:**
    1. Attempt to upload a file in an unsupported format (e.g., .exe or .txt).
- **Expected Outcome:**
    - An error message is displayed, and the file is rejected with a prompt to use a valid format.

Test Case DS-5: Diagnostic Test Rescheduling

- **Objective:** Ensure that an existing diagnostic test can be rescheduled if needed.
- **Preconditions:**
    - A diagnostic test order exists with a scheduled date/time.
    - The remediating user (doctor or lab coordinator) is logged in.
- **Test Steps:**
    1. Select a scheduled test order in the Diagnostic Services module.
    2. Choose a different date/time.
    3. Resubmit the scheduling details.
- **Expected Outcome:**
    - The test order is updated with the new schedule and notifications are sent to the relevant parties.

7. Role-Based Login

Test Case RL-1: Successful Login for a Valid User

- **Objective:** Confirm that a user with valid credentials and the correct role can log in and view their dashboard.
- **Preconditions:**
    - A valid user account exists.
- **Test Steps:**
    1. Enter correct username and password.
    2. Click “Login.”
- **Expected Outcome:**
    - Successful authentication and redirection to the user-specific dashboard.

Test Case RL-2: Login Attempt with Incorrect Credentials

- **Objective:** Verify that access is denied when invalid credentials are provided.
- **Preconditions:**
    - A valid user account exists.
- **Test Steps:**
    1. Enter an incorrect username or password.
    2. Attempt to log in.
- **Expected Outcome:**
    - An error message (e.g., “Invalid credentials”) is displayed, and access is denied.

Test Case RL-3: Role-Specific Access Control

- **Objective:** Ensure that users cannot access modules outside their assigned role.
- **Preconditions:**
    - A user logged in (e.g., a patient) does not have permissions to access administrative modules.
- **Test Steps:**
    1. Attempt to navigate to an admin or doctor-specific URL/module.
- **Expected Outcome:**
    - Access is blocked with an “Access Denied” message, and the action is logged.

Test Case RL-4: Password Recovery Functionality

- **Objective:** Validate that users can initiate the “Forgot Password” process successfully.
- **Preconditions:**
    - The user has a registered email or phone number.
- **Test Steps:**
    1. Click on “Forgot Password” on the login page.
    2. Enter the registered contact information.
    3. Submit the request.
- **Expected Outcome:**
    - A password reset link or code is sent, allowing the user to recover access.

Test Case RL-5: Session Timeout Handling

- **Objective:** Verify that inactive sessions are automatically logged out after a set period.
- **Preconditions:**
    - A user is logged in.
- **Test Steps:**
    1. Leave the session idle for the configured timeout duration (e.g., 15 minutes).
- **Expected Outcome:**
    - The system automatically logs out the user, requiring a fresh login.

8. Ambulance Booking

Test Case AB-1: Valid Emergency Ambulance Booking

- **Objective:** Verify that an emergency ambulance booking request is processed immediately.
- **Preconditions:**
    - Patient/caregiver is logged in.
    - At least one ambulance is available.

- **Test Steps:**
    1. Navigate to the Ambulance Booking module.

2. Select “Emergency” and complete the required fields (location, contact details).
3. Submit the request.

- **Expected Outcome:**
  - The system confirms a dispatch of the ambulance with immediate notification to the user.

Test Case AB-2: Scheduled Ambulance Booking

- **Objective:** Validate that non-emergency ambulance bookings can be scheduled for future dates and times.
- **Preconditions:**
  - Patient or staff is planning a scheduled transfer.
- **Test Steps:**
  1. Choose “Scheduled” mode in the Ambulance Booking module.
  2. Select a future date/time and provide necessary details.
  3. Submit the booking request.
- **Expected Outcome:**
  - The booking is recorded for the specified time, and a confirmation is sent.

Test Case AB-3: Handling No Ambulance Availability

- **Objective:** Ensure that when all ambulances are booked, the system provides a clear notification.
- **Preconditions:**
  - All ambulances are currently in use (booked or en route).
- **Test Steps:**
  1. Attempt an ambulance booking during a high-demand period.
- **Expected Outcome:**
  - An error message is displayed stating “No ambulance available at this time,” with alternative suggestions if applicable.

Test Case AB-4: Ambulance Booking Cancellation

- **Objective:** Verify that an existing ambulance booking can be cancelled.
- **Preconditions:**
  - A valid ambulance booking exists.

- **Test Steps:**
    1. Navigate to the Ambulance Booking history.
    2. Select the booking to cancel.
    3. Confirm cancellation.
- **Expected Outcome:**
    - The booking is marked “Cancelled” and a confirmation is sent.

Test Case AB-5: Ambulance Rebooking After Cancellation

- **Objective:** Confirm that after cancelling an ambulance booking, the patient can rebook a new appointment easily.
- **Preconditions:**
    - A cancelled ambulance booking exists.
- **Test Steps:**
    1. Cancel an existing ambulance booking.
    2. Return to the booking module and initiate a new booking.
    3. Provide any updated details and submit.
- **Expected Outcome:**
    - A new booking is created successfully with confirmation, and availability is rechecked.

9. Prescription Management

Test Case PM-1: Creating a Prescription

- **Objective:** Confirm that a doctor can create and issue a new prescription accurately.
- **Preconditions:**
    - Doctor is logged in and a valid patient record is available.
- **Test Steps:**
    1. Navigate to the Prescription Management module.
    2. Select the patient’s profile.
    3. Fill in details (medication, dosage, instructions).
    4. Submit the new prescription.

- **Expected Outcome:**
    - A prescription with a unique ID is generated and a confirmation is displayed.

Test Case PM-2: Modifying an Existing Prescription

- **Objective:** Validate that an existing prescription can be updated before finalization.
- **Preconditions:**
    - A prescription exists for the patient.
    - Doctor is logged in.
- **Test Steps:**
    1. Access the patient's prescription history.
    2. Select a prescription to modify.
    3. Update relevant fields.
    4. Submit the changes.
- **Expected Outcome:**
    - The prescription is updated, changes are logged, and a confirmation message is displayed.

Test Case PM-3: Link Prescription to Medication Inventory

- **Objective:** Ensure that a prescription includes references to medications from the inventory.
- **Preconditions:**
    - Valid medication data exists in the inventory.
- **Test Steps:**
    1. In the prescription creation module, select medications from a provided list (or via barcode scanning).
    2. Complete and submit the prescription.
- **Expected Outcome:**
    - The prescription displays correct medication details and reflects current inventory status.

Test Case PM-4: Download Prescription as PDF

- **Objective:** Verify that the system can generate and download a PDF copy of a prescription.
- **Preconditions:**
    - A valid prescription exists.

- **Test Steps:**
    1. Navigate to the prescription details page.
    2. Click “Download PDF.”
- **Expected Outcome:**
    - A PDF file is generated and downloaded successfully.

Test Case PM-5: Prescription Forwarding to Pharmacy Notification

- **Objective:** Ensure that once a prescription is created, a notification is sent to the pharmacy for dispensing.
- **Preconditions:**
    - A valid prescription is issued.
- **Test Steps:**
    1. Create and submit the prescription.
    2. Verify that the prescription is forwarded automatically to the pharmacy module.
- **Expected Outcome:**
    - The pharmacy receives the prescription details and a notification is logged.

10. Billing & Payment System

Test Case BP-1: Automatic Bill Generation Post-Appointment

- **Objective:** Confirm that completing an appointment automatically generates a billing record with correct service charges.
- **Preconditions:**
    - A completed appointment exists with associated services.
- **Test Steps:**
    1. Complete an appointment.
    2. Allow the system to process the billing generation.
- **Expected Outcome:**
    - A billing record with a unique bill ID is generated, and its status is set to “unpaid.”

Test Case BP-2: Successful Multi-Mode Payment Processing

- **Objective:** Validate that a patient can pay a generated bill using various payment methods.
- **Preconditions:**
    - A billing record exists for a completed service.
- **Test Steps:**
    1. Navigate to the Payment section for the bill.
    2. Select a payment method (credit card, cash, online transfer).
    3. Enter required information and process the payment.
- **Expected Outcome:**
    - The payment is processed successfully, the bill status updates to "paid," and a receipt is generated.

Test Case BP-3: Applying Discounts and Insurance

- **Objective:** Ensure that discounts or insurance details are properly applied, modifying the final bill amount.
- **Preconditions:**
    - A billing record exists; discount/insurance options are available.
- **Test Steps:**
    1. During payment processing, input valid discount/insurance codes.
    2. Submit and recalculate the amount.
- **Expected Outcome:**
    - The final bill reflects the discount/insurance adjustment accurately and the updated invoice is displayed.

Test Case BP-4: Handling Payment Gateway Failures

- **Objective:** Validate that payment processing errors (e.g., gateway issues) are handled gracefully.
- **Preconditions:**
    - A billing record exists; simulate an error (e.g., incorrect payment info or gateway outage).
- **Test Steps:**
    1. Initiate the payment with faulty details.
- **Expected Outcome:**

- An error message is displayed, the transaction is aborted, and no update is made to the billing record.

Test Case BP-5: Invoice Download and Billing History

- **Objective:** Verify that users can view and download invoices from their billing history.
- **Preconditions:**
  - Previous billing records with generated invoices exist.
- **Test Steps:**
  1. Navigate to the Billing History module.
  2. Select a past invoice and click “Download Invoice.”
- **Expected Outcome:**
  - A downloadable PDF invoice is generated and matches the on-screen billing details.

Conclusion

This full test suite addresses 10 critical features of the Hospital Management System—covering Appointment Booking, Patient History, Staff Management, Room Allocation, Medication Store, Diagnostic Services, Role-Based Login, Ambulance Booking, Prescription Management, and Billing & Payment System. Each feature has been validated through detailed test cases designed to verify positive scenarios, input validations, error handling, role-specific restrictions, and inter-module integrations. Integrate these test cases into both manual and automated testing environments to ensure a robust, user-friendly, and reliable system as part of the overall quality assurance plan.