

Darshan University

A Project Report on

**“Hospital Management System”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name:Savaliya Ronit | Enrollment No.:23010101247 |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2301CS405)** for entitled **“Hospital Management System”** submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of R. B. Gondaliya and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

Ronit Savaliya

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Hospital Management System” has** been satisfactorily prepared by **Ronit Savaliya** (**23010101247**) under my guidance in the fulfillment of the course **Software Engineering (2301CS405)** work during the academic year 2024-2025.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. R. B. Gondaliyaand all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Ronit Savaliya**

**ABSTRACT**

The Hospital Management System (HMS) is an integrated software solution designed to streamline and automate hospital operations, ensuring efficient management of patients, staff, and administrative tasks. This system enables functionalities such as patient registration, appointment scheduling, electronic health records (EHR) management, billing, inventory tracking, and reporting. By centralizing and digitizing hospital workflows, the HMS aims to enhance operational efficiency, reduce manual errors, and improve patient care. The system supports multiple user roles, including patients, doctors, administrative staff, and pharmacists, providing role-specific access to functionalities through an intuitive interface. Key features include real-time appointment booking and notifications, secure data handling with role-based access and encryption, pharmacy and inventory management to monitor stock levels and reduce wastage, and financial management tools to process payments and generate invoices. The HMS integrates with third-party systems such as laboratory management tools and insurance platforms, ensuring seamless communication and data sharing. It is designed to comply with healthcare regulations, such as HIPAA and GDPR, to maintain data privacy and security. Scalability and flexibility are core attributes, enabling the system to cater to hospitals of various sizes, including multi-branch setups. The HMS is cloud-based, ensuring high availability and accessibility across devices, including desktops and mobile platforms. Additionally, advanced analytics and reporting tools allow hospital administrators to monitor operational trends, optimize resource allocation, and make informed decisions. By providing a unified platform for hospital management, the HMS addresses the challenges of manual workflows, enhances the patient experience, and empowers healthcare professionals to focus more on delivering quality care. This SRS outlines the functional and non-functional requirements of the HMS, covering its scope, features, design constraints, and intended user interactions, serving as a comprehensive guide for stakeholders, developers, and testers involved in the system's implementation.

**Table of Contents**

[List of Figures I](#_Toc193457222)

[List of Tables II](#_Toc193457223)

[1 Introduction 1](#_Toc193457224)

[1.1 Purpose 1](#_Toc193457225)

[1.2 Scope 1](#_Toc193457226)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc193457227)

[1.4 Definitions, Acronyms, and Abbreviations 2](#_Toc193457228)

[1.5 Functional Requirements 2](#_Toc193457229)

[1.5.1 Administrative 2](#_Toc193457230)

[1.5.2 Receptionist 2](#_Toc193457231)

[1.5.3 Doctor 2](#_Toc193457232)

[1.5.4 Staff/Nurse 2](#_Toc193457233)

[1.5.5 Patient 3](#_Toc193457234)

[1.5.6 Pharmacist 3](#_Toc193457235)

[1.6 Non-Functional Requirement 3](#_Toc193457236)

[1.6.1 Usability: 3](#_Toc193457237)

[1.6.2 Accuracy: 3](#_Toc193457238)

[1.6.3 Availability: 3](#_Toc193457239)

[1.6.4 Maintainability: 3](#_Toc193457240)

[2 Design and Implementation Constraints 4](#_Toc193457241)

[2.1 Use case diagram 4](#_Toc193457242)

[2.2 Activity diagram and Swimlane diagram 6](#_Toc193457243)

[2.3 Sequence diagram 9](#_Toc193457244)

[2.4 State diagram 11](#_Toc193457245)

[2.5 Class diagram 12](#_Toc193457246)

[2.6 Data flow diagram 13](#_Toc193457247)

[2.6.1 Context diagram (level-0) 13](#_Toc193457248)

[2.6.2 DFD Level-1 14](#_Toc193457249)

[3 External interface requirement (Screens) 15](#_Toc193457250)

[3.1 Screen-1: View & Feedback Handling 15](#_Toc193457251)

[3.2 Screen-2: Emergency Response 16](#_Toc193457252)

[3.3 Screen-3: Schedule Appointments 17](#_Toc193457253)

[3.4 Screen-4: Access Patient Records 18](#_Toc193457254)

[3.5 Screen-5: View Billing Details 19](#_Toc193457255)

[4 Database design 20](#_Toc193457256)

[4.1 List of Tables 20](#_Toc193457257)

[5 Stories and Scenario 22](#_Toc193457258)

[5.1 Story-1: Schedule Appointments 22](#_Toc193457259)

[5.1.1 Scenario# S1.1 22](#_Toc193457260)

[5.1.2 Scenario# S1.2 22](#_Toc193457261)

[5.1.3 Scenario# S1.3 22](#_Toc193457262)

[5.2 Story-2: Pay Bills Online 23](#_Toc193457263)

[5.2.1 Scenario# S2.1 23](#_Toc193457264)

[5.2.2 Scenario# S2.2 23](#_Toc193457265)

[5.3 Story-3: View & Handling Feedback 23](#_Toc193457266)

[5.3.1 Scenario# S3.1 24](#_Toc193457267)

[5.3.2 Scenario# S3.2 24](#_Toc193457268)

[6 Test cases 25](#_Toc193457269)

[7 References 28](#_Toc193457270)

# List of Figures

[Figure 2.1‑1 Use case diagram for hospital management system 4](#_Toc193457544)

[Figure 2.1‑2 Use case diagram for hospital management system 5](#_Toc193457545)

[Figure 2.2‑1 Activity diagram for payment 6](#_Toc193457546)

[Figure 2.2‑2 Activity diagram for Book appointment 7](#_Toc193457547)

[Figure 2.2‑3 Swimlane diagram for Book appointment 8](#_Toc193457548)

[Figure 2.3‑1 Sequence diagram for Book appointment 9](#_Toc193457549)

[Figure 2.3‑2 Sequence diagram for check bed availability 10](#_Toc193457550)

[Figure 2.4‑1 State diagram of Book 11](#_Toc193457551)

[Figure 2.5‑1 Class diagram for Library management system 12](#_Toc193457552)

[Figure 2.6‑1 Context diagram for hospital management system 13](#_Toc193457553)

[Figure 3.1‑1 Screen-1: View Feedback 15](#_Toc193457554)

[Figure 3.2‑1 Screen-2: Emergency Response 16](#_Toc193457555)

[Figure 3.3‑1 Screen-3: Schedule Appointment 17](#_Toc193457556)

[Figure 3.4‑1 Screen-4: Access Patient Records 18](#_Toc193457557)

[Figure 3.5‑1 Screen-5: Billing Details 19](#_Toc193457558)

# List of Tables

[Table 3.1‑1 Screen element of View Feedback 15](#_Toc193457559)

[Table 3.2‑1 Screen element of Emergency Response 16](#_Toc193457560)

[Table 3.3‑1 Screen element of Schedule Appointments 18](#_Toc193457561)

[Table 3.4‑1 Screen element of Access Patient Records 18](#_Toc193457562)

[Table 3.5‑1 Screen element of View Billing Details 19](#_Toc193457563)

[Table 4.1‑1 Table: Doctor 20](#_Toc193457564)

[Table 4.1‑2 Table: Patient 20](#_Toc193457565)

[Table 4.1‑3 Table: Staff/Nurse 20](#_Toc193457566)

[Table 4.1‑4 Table: Appointment 21](#_Toc193457567)

[Table 4.1‑5 Table: Bill 21](#_Toc193457568)

# Introduction

## Purpose

The purpose of this document is to define the Software Requirements Specification (SRS) for the **Hospital Management System (HMS)**. This system is designed to streamline the operations and management of hospitals by automating various administrative and patient-care processes. It provides a centralized platform for managing patient information, appointments, medical records, billing, inventory, and staff details, ensuring efficiency and improved patient care.

This document serves as a reference for stakeholders, including system developers, testers, hospital management, and end-users, to ensure a common understanding of the system’s functionality and performance.

## Scope

The **Hospital Management System (HMS)** is a web-based application aimed at managing hospital operations in a digital and efficient manner. It is designed to handle patient registration, appointment scheduling, medical records storage, billing and invoicing, staff management, and inventory control.  
The system will serve various users such as hospital administrators, doctors, nurses, and patients, with role-based access to ensure data security and privacy. HMS is intended to reduce paperwork, improve resource utilization, and enhance the overall quality of healthcare services.

The scope of this project includes:

* **Patient Management**: Registration, appointment scheduling, and medical record tracking.
* **Billing and Payments**: Generation of invoices, online payment integration, and financial reporting.
* **Inventory Management**: Tracking and managing medical supplies and equipment.
* **Staff Management**: Employee details, scheduling, and role assignments.

The system will be developed using a user-friendly interface and scalable architecture to accommodate future enhancements.

## Intended Audience and Reading Suggestions

This document is intended for:

* **Developers**: To understand the technical requirements and system features for implementation.
* **Testers**: To define testing criteria and ensure the system meets functional and non-functional requirements.
* **Hospital Management and Stakeholders**: To provide insights into system functionality and evaluate its alignment with business needs.
* **End Users**: To understand the operational capabilities of the system.

## Definitions, Acronyms, and Abbreviations

* **HMS**: Hospital Management System
* **EMR**: Electronic Medical Records
* **API**: Application Programming Interface
* **UI/UX**: User Interface/User Experience
* **Admin**: System Administrator

## Functional Requirements

### Administrative

* **View & Handling Feedback**: Review and address patient feedback submitted through the system.
* **Add Billing**: Generate, update, and finalize bills for patient services.
* **User Account Creation**: Create user accounts for staff, doctors, and patients with role-based permissions.
* **Modify User Permissions**: Update user roles and access rights when needed.
* **Generate Reports**: Produce detailed reports on hospital operations, including financial and staff performance summaries.
* **Event Scheduling**: Schedule hospital-wide events or meetings and notify relevant users.
* **Monitor System Logs**: Access detailed logs for all system activities, such as billing, user actions, and updates.

### Receptionist

* **Schedule Appointments**: Book, reschedule, or cancel patient appointments based on availability.
* **Verify Patient Identity**: Confirm patient identity during registration or appointment check-ins.
* **Check Doctor Availability**: Display doctors’ schedules for easy slot allocation.
* **Issue Patient ID**: Generate unique identification numbers for new patients.
* **Print Receipts**: Provide printed or digital receipts for payments and services.

### Doctor

* **Access Patient Records**: View comprehensive patient details, including medical history and previous visits.
* **Update Treatment Plans**: Add or modify treatment strategies, including medications and lifestyle recommendations.
* **Request Diagnostic Tests**: Order lab tests and set priorities based on patient conditions.
* **Prescribe Medications**: Create digital prescriptions specifying dosage and frequency.
* **Record Progress Notes**: Add detailed progress notes for patients undergoing long-term care.
* **Refer Specialists**: Generate referral documents for patients requiring specialized treatment.

### Staff/Nurse

* **Emergency Response**: Record emergency cases, including vital signs and immediate care actions.
* **Record Patient Vitals**: Log and update vital signs like blood pressure, heart rate, and oxygen levels.
* **Bed Assignment**: Allocate hospital beds for admitted patients and manage ward assignments.
* **Medication Administration**: Confirm and document administered doses of prescribed medicines.
* **Support Discharge Process**: Assist in preparing discharge summaries and coordinating patient releases.

### Patient

* **View Billing Details**: Access itemized bills for treatments, consultations, and tests.
* **Pay Bills Online**: Use integrated payment gateways for settling bills securely.
* **View Notifications**: Receive updates on upcoming appointments, test results, or doctor availability.
* **Access Medical Records**: View historical data, including prescriptions and diagnostic reports.
* **Submit Feedback**: Share feedback on services and care received at the hospital.
* **Track Appointment Status**: Check the confirmation or cancellation status of booked appointments.

### Pharmacist

* **Process Prescriptions**: Retrieve and dispense medications based on electronic prescriptions.
* **Update Stock Levels**: Log changes in inventory, including newly received or dispensed medicines.
* **Receive Restock Alerts**: Get notified when stock levels fall below a predefined threshold.
* **Track Expiry Dates**: Maintain a record of medicines nearing expiration for timely action.
* **Generate Sales Records**: Create daily, weekly, and monthly summaries of issued medications.

## Non-Functional Requirement

### Usability:

* The UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements.

### Accuracy:

* The data stored about the books and the fines calculated should be correct, consistent, and reliable.

### Availability:

* The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

### Maintainability:

* The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

# Design and Implementation Constraints

## Use case diagram

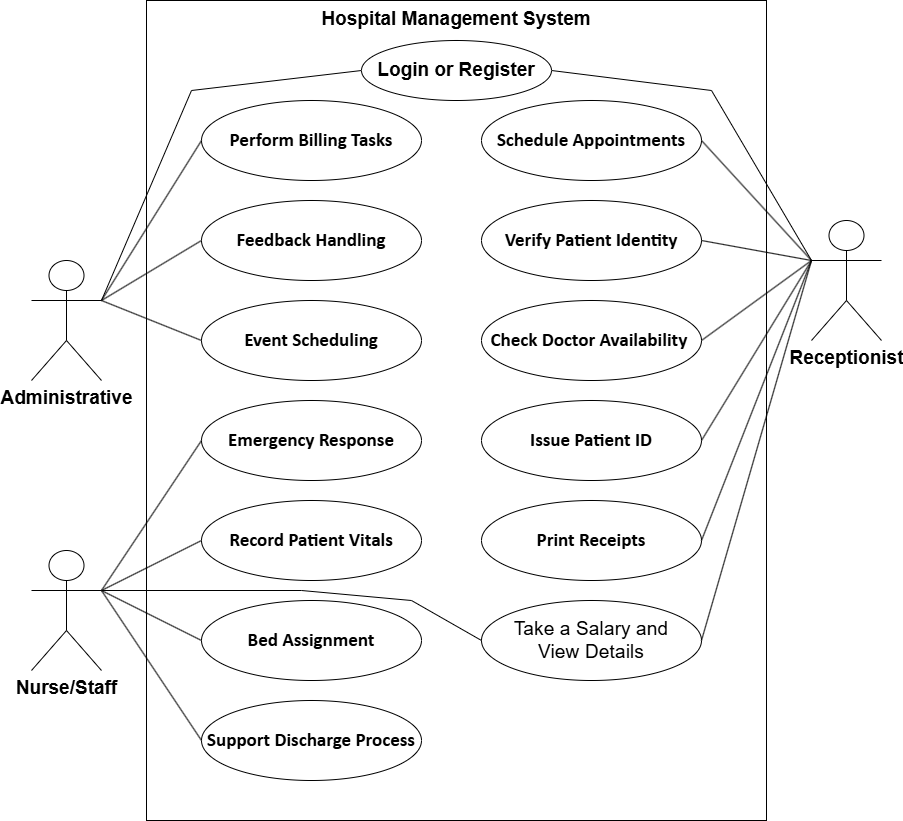


Figure 2.1‑1 Use case diagram for hospital management system

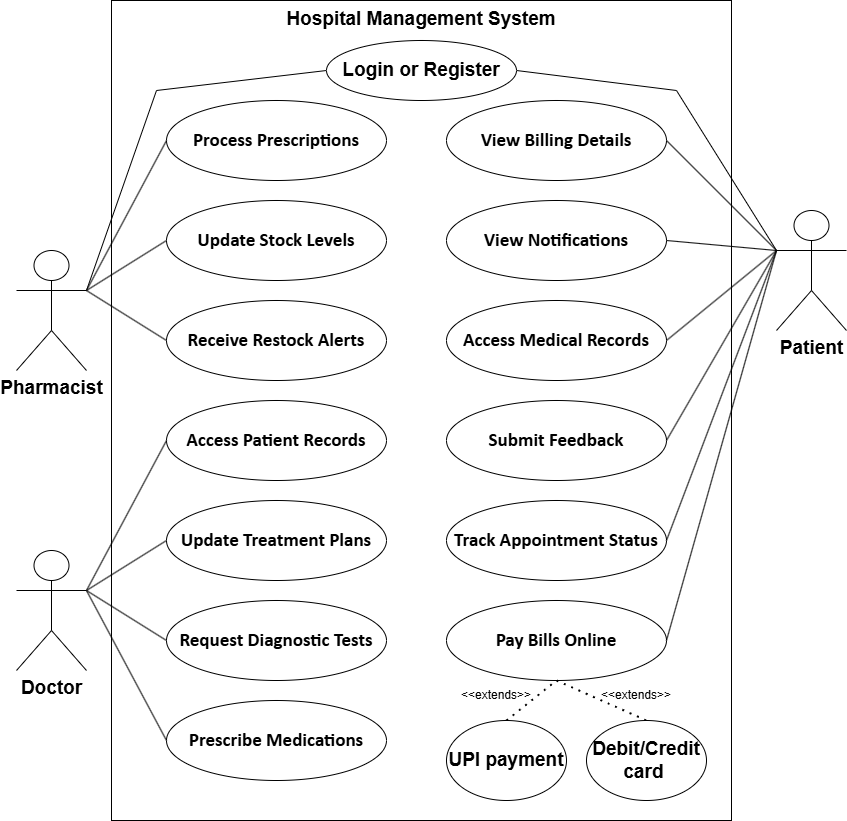


Figure 2.1‑2 Use case diagram for hospital management system

## Activity diagram and Swimlane diagram

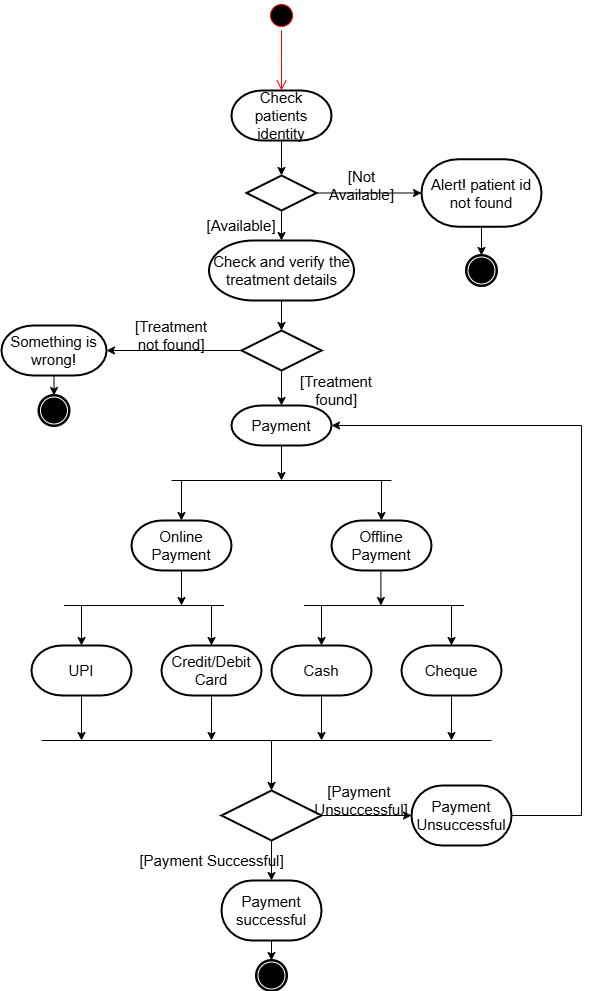


Figure 2.2‑1 Activity diagram for payment

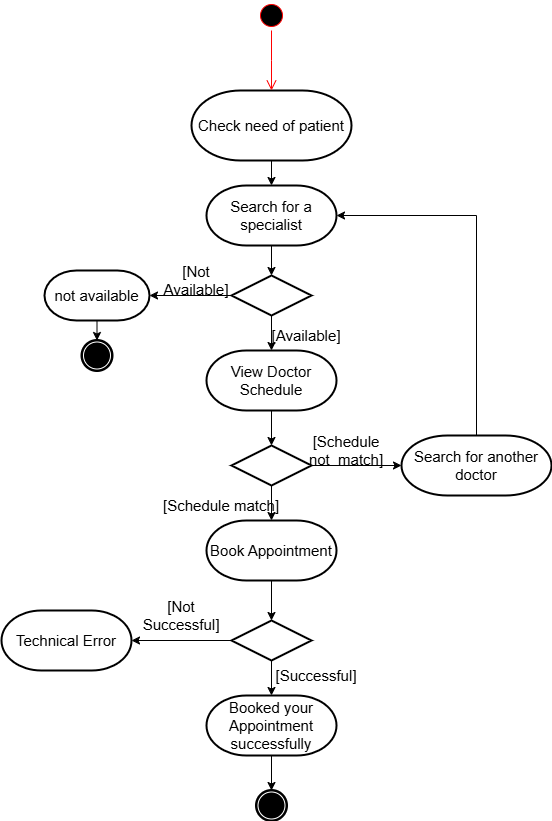


Figure 2.2‑2 Activity diagram for Book appointment

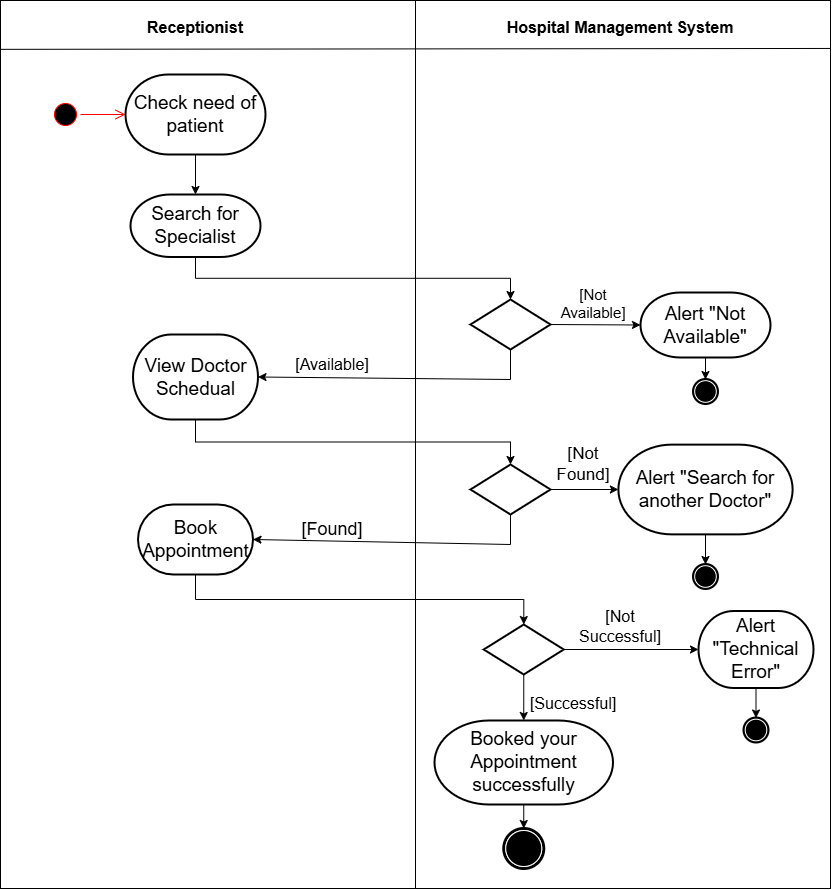


Figure 2.2‑3 Swimlane diagram for Book appointment

## Sequence diagram

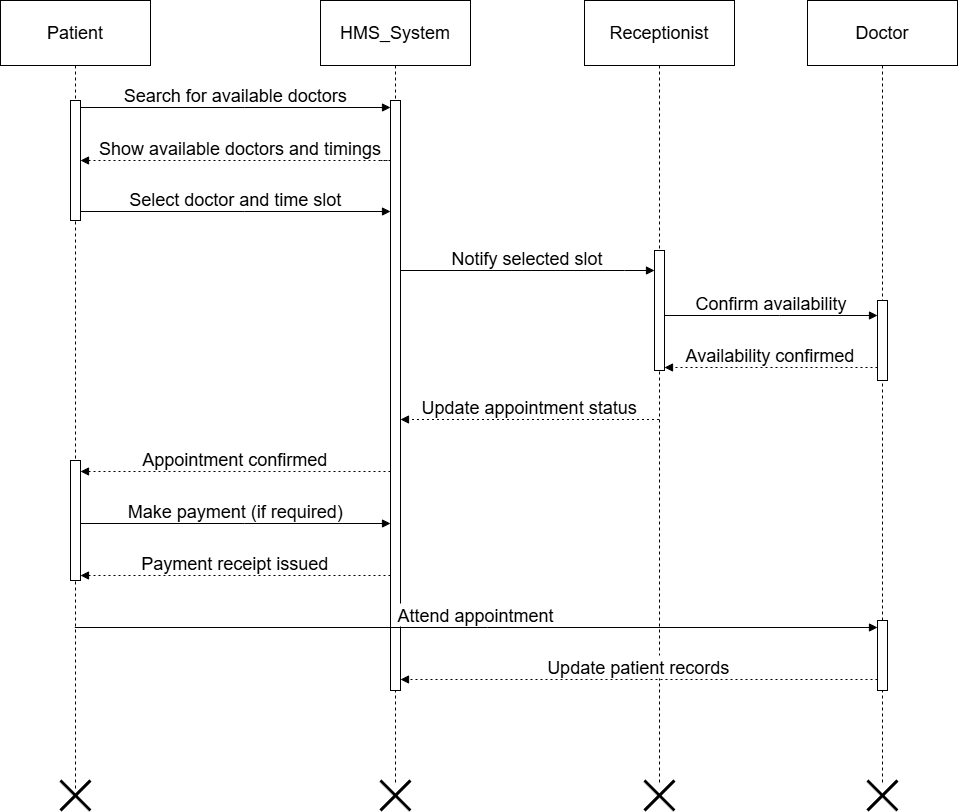


Figure 2.3‑1 Sequence diagram for Book appointment

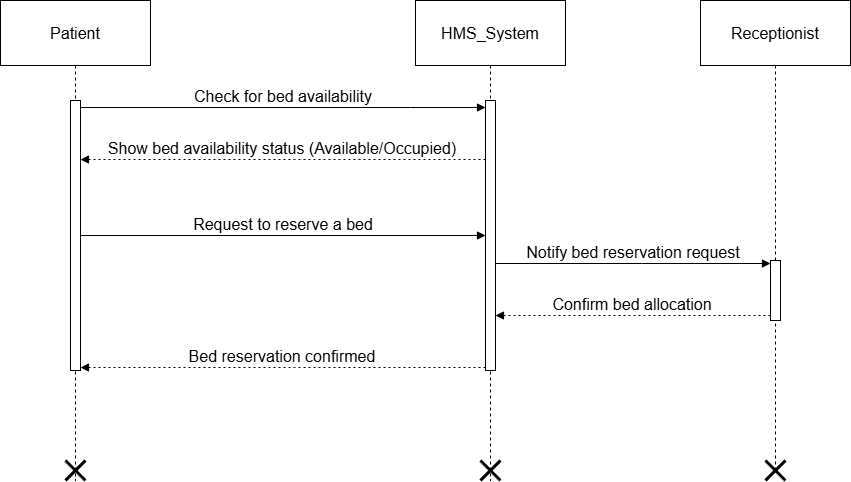


Figure 2.3‑2 Sequence diagram for check bed availability

## State diagram

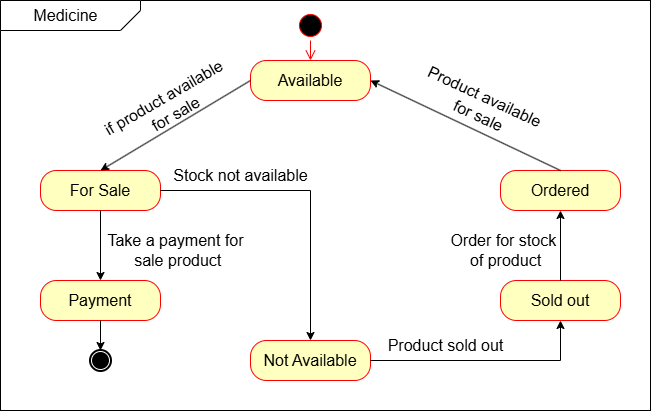


Figure 2.4‑1 State diagram of Book

## Class diagram

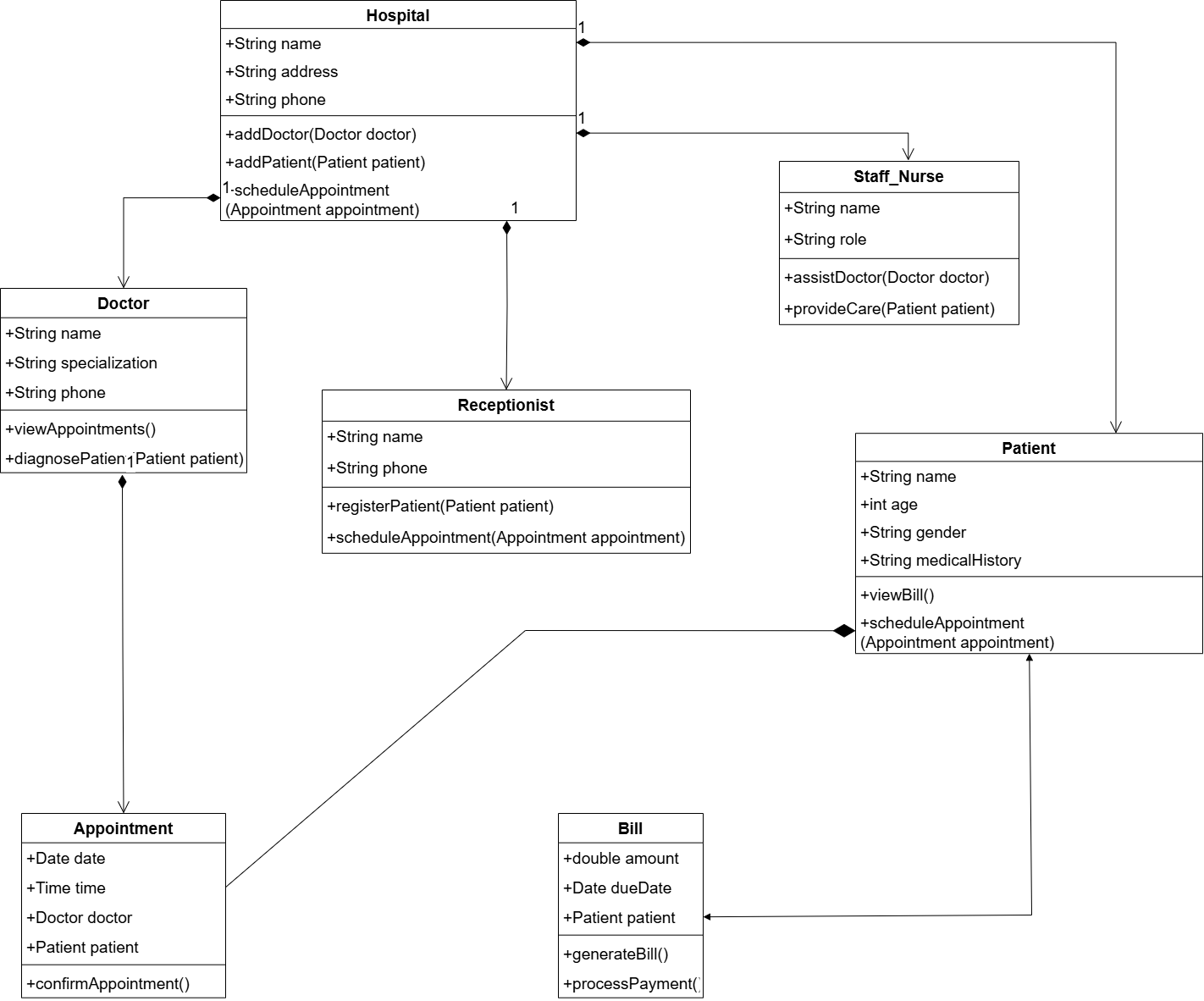


Figure 2.5‑1 Class diagram for Library management system

## Data flow diagram

### Context diagram (level-0)

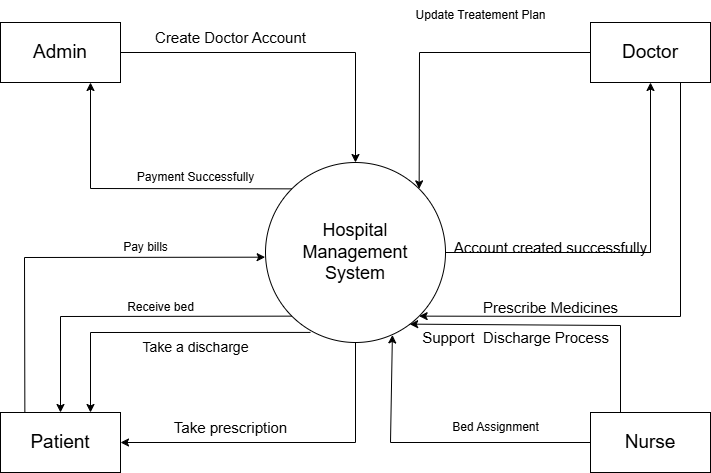


Figure 2.6‑1 Context diagram for hospital management system

### DFD Level-1

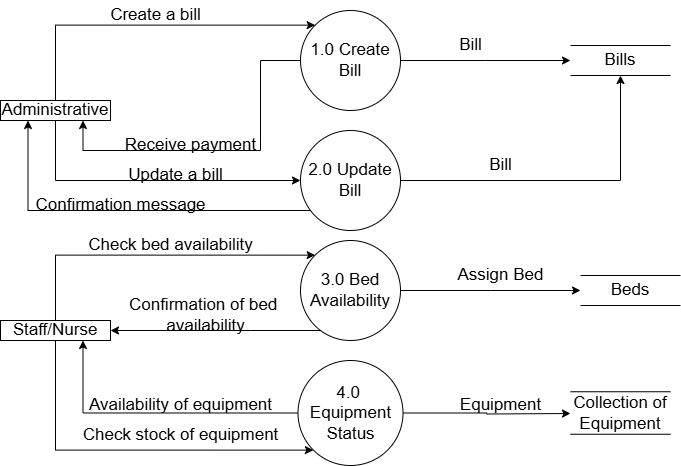


Figure 2.6‑2 DFD level-1 for Library management system

# External interface requirement (Screens)

## Screen-1: View & Feedback Handling

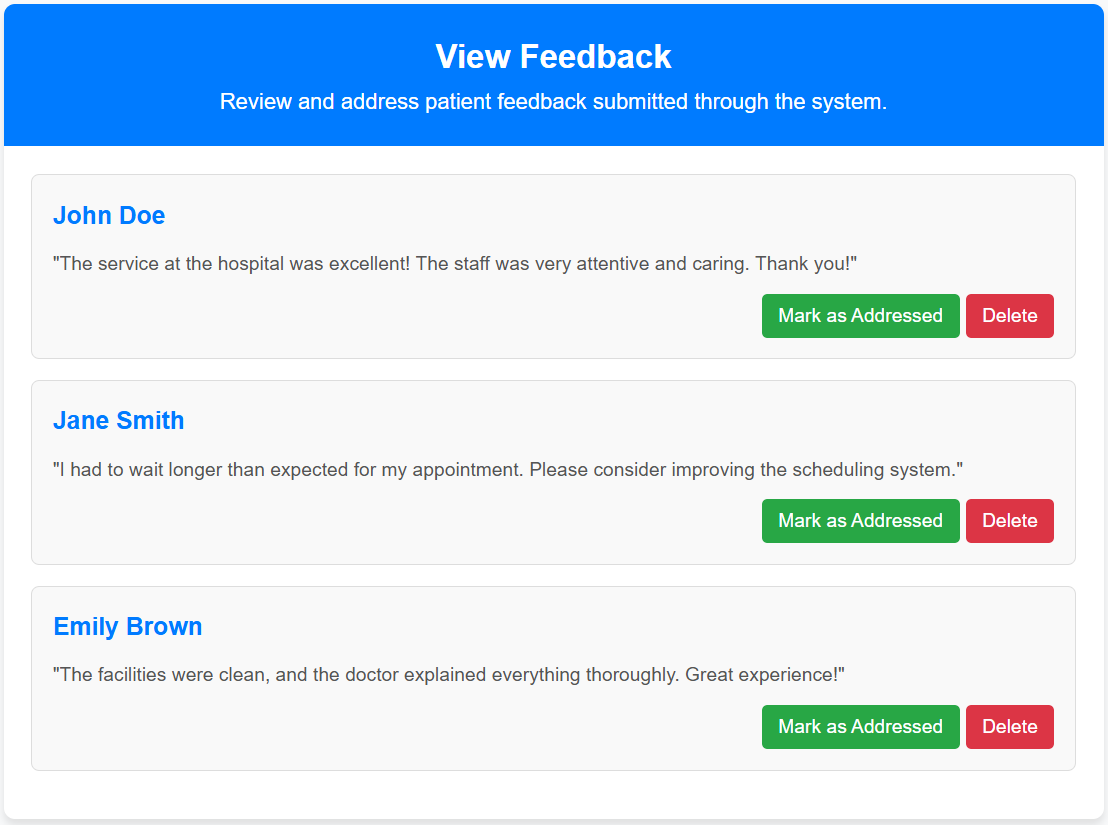


Figure 3.1‑1 Screen-1: View Feedback

**Purpose:** Enables reviewing and addressing patient feedback to improve service quality.

Table 3.1‑1 Screen element of View Feedback

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Header (View Feedback) | Text | M | 1 | Displays the page title and description. |
| **2** | Feedback List | Container | M | 1 | Displays a list of patient feedback items. |
| **3** | Feedback Item (Name) | Text | M | 1 | Displays the name of the patient providing feedback. |
| **4** | Feedback Item (Message) | Text | M | 1 | Displays the feedback message from the patient. |
| **5** | Mark as Addressed Button | Button | M | 1 | Button to mark feedback as addressed. |
| **6** | Delete Button | Button | M | 1 | Button to delete feedback. |

## Screen-2: Emergency Response

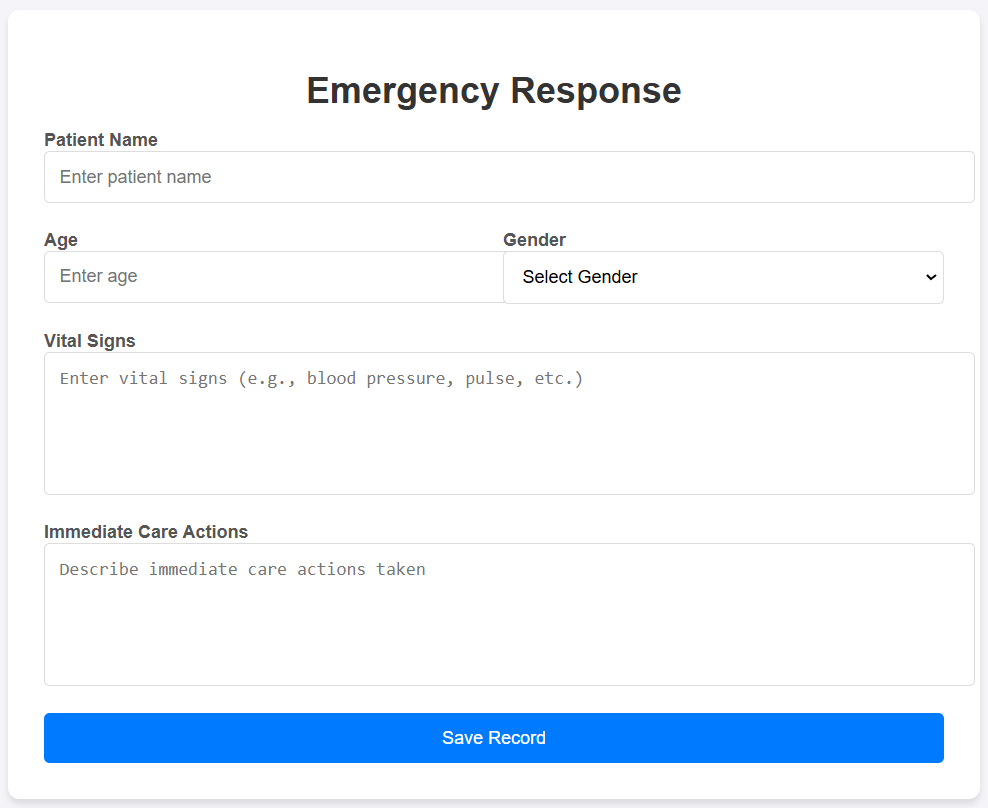


Figure 3.2‑1 Screen-2: Emergency Response

**Purpose:** Records and manages emergency cases, including vital signs and immediate care actions.

Table 3.2‑1 Screen element of Emergency Response

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Patient Name | Textbox | M | 1 | This field is for users to input the patient’s name. |
| **2** | Age | Number | M | 1 | Users input the patient’s age, typically used for identification and medical purposes. |
| **3** | Gender | Dropdown | M | 1 | A dropdown to select the patient’s gender (Male, Female, or Other). |
| **4** | Vital Signs | Textarea | M | 1 | A textarea to input vital signs like blood pressure, pulse, etc. |
| **5** | Immediate Care Actions | Textarea | M | 1 | A textarea to describe immediate care actions taken for the patient. |
| **6** | Save Record | Button |  |  | For save above information |

## Screen-3: Schedule Appointments

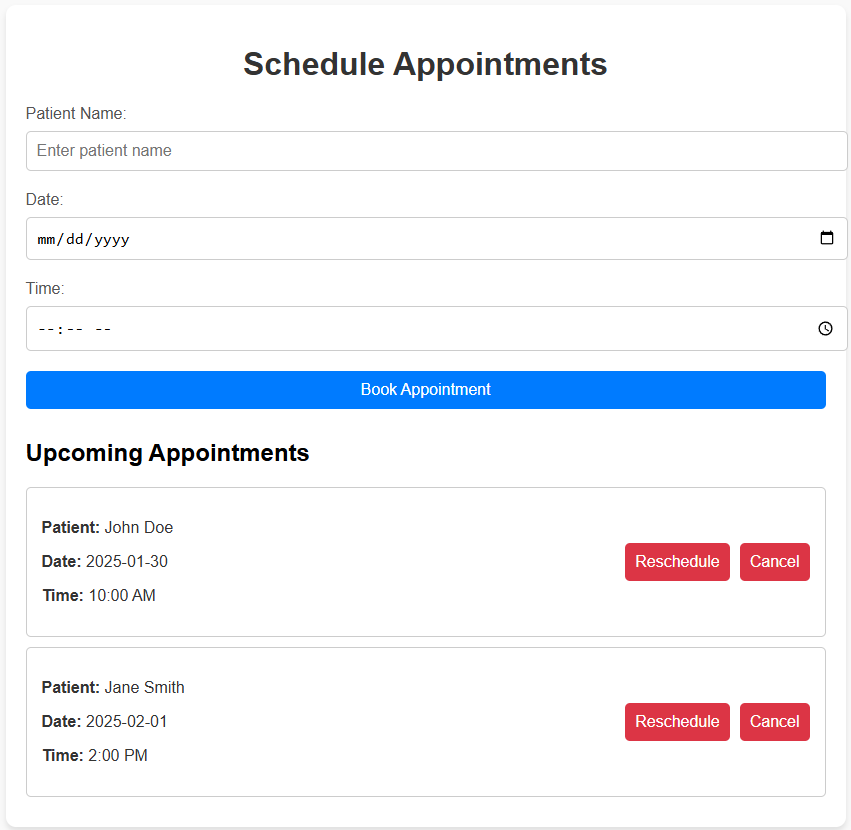


Figure 3.3‑1 Screen-3: Schedule Appointment

**Purpose:** Facilitates booking, rescheduling, or canceling patient appointments based on availability.

Table 3.3‑1 Screen element of Schedule Appointments

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| 1 | Patient Name | Text Input | M | 1 | Field to enter the name of the patient for the appointment. |
| 2 | Date | Date Picker | M | 1 | Allows the user to select the date for the appointment. |
| 3 | Time | Time Picker | M | 1 | Allows the user to select the time for the appointment. |
| 4 | Book Appointment Button | Button | M | 1 | Submits the entered information to book an appointment. |
| 5 | Upcoming Appointments | Section Header | M | 1 | Displays the list of upcoming appointments for patients. |
| 6 | Appointment Card | Display | O | N | A card displaying details (Patient Name, Date, Time) for each appointment. |
| 7 | Reschedule Button | Button | O | N | Button to reschedule the selected appointment. |
| 8 | Cancel Button | Button | O | N | Button to cancel the selected appointment. |

## Screen-4: Access Patient Records

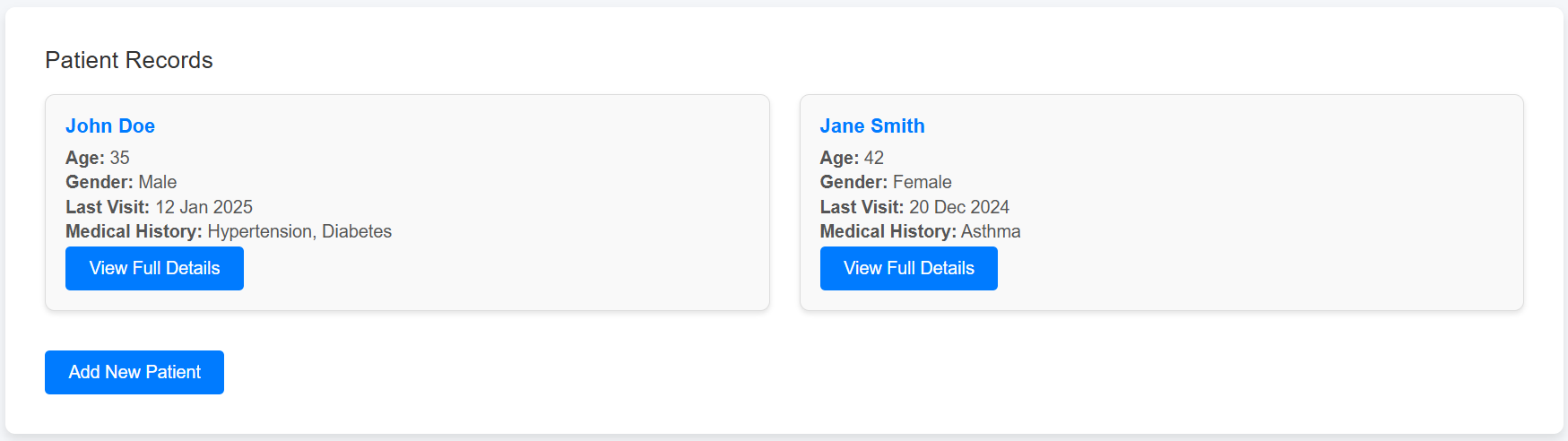


Figure 3.4‑1 Screen-4: Access Patient Records

**Purpose:** Provides access to comprehensive patient details, including medical history and previous visits.

Table 3.4‑1 Screen element of Access Patient Records

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| 1 | Patient Records Header | Header | M | 1 | Header for the patient records section. |
| 2 | Patient Card (John Doe, Jane Smith, etc.) | Card | M | N | Card displaying the details of each patient. |
| 3 | View Full Details Button | Button | O | N | Button to view more detailed information about the patient. |
| 4 | Add New Patient Button | Button | M | 1 | Button to add a new patient record. |
| 5 | Patient Details (Age, Gender, Last Visit, Medical History) | Display | O | N | Displays patient-specific details like age, gender, last visit, and medical history. |

## Screen-5: View Billing Details

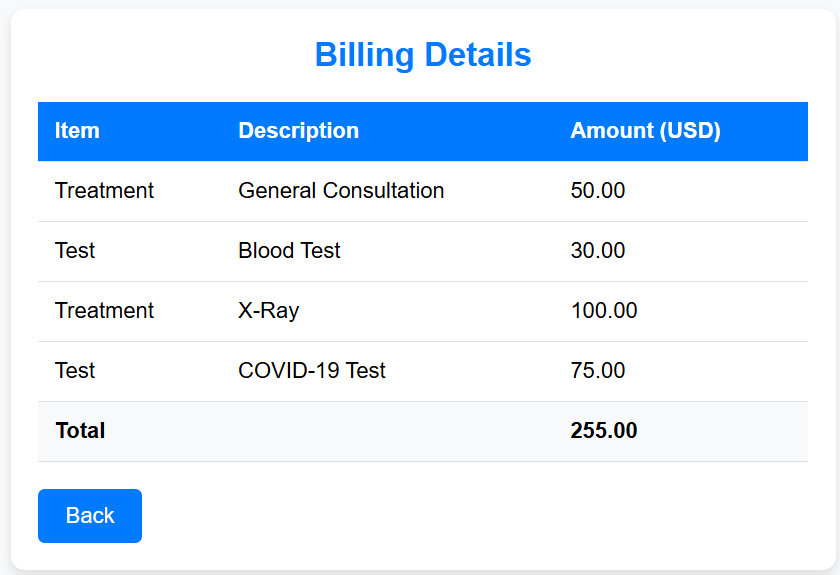


Figure 3.5‑1 Screen-5: Billing Details

**Purpose:** Displays itemized bills for treatments, consultations, and tests for transparency.

Table 3.5‑1 Screen element of View Billing Details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| 1 | Header 'Billing Details' | Text Display | M | 1 | Displays the title of the billing details screen. |
| 2 | Table Header 'Item' | Text Display | M | 1 | Header of the table showing the type of item (e.g., Treatment, Test). |
| 3 | Table Header 'Description' | Text Display | M | 1 | Header of the table showing the description of the item. |
| 4 | Table Header 'Amount (USD)' | Text Display | M | 1 | Header of the table showing the amount in USD. |
| 5 | Table Rows (Items) | Text Display | M | N | Displays rows for items, descriptions, and amounts. |
| 6 | Total Row | Text Display | M | 1 | Displays the total amount of all the items. |
| 7 | 'Back' Button | Button | M | 1 | Button to navigate back to the previous screen. |

# Database design

## List of Tables

* Doctor
* Patient
* Staff/Nurse
* Appointment
* Bill

Table 4.1‑1 Table: Doctor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| DoctorID | int | AN | PK (Auto Increment) | Unique Identifier for each patient. |
| Name | int | NN |  | Name of the patients |
| Specializations | varchar(100) | NN |  | Specialization field of the doctor |
| Qualification | varchar(100) | AN |  | Qualifications of the doctor |
| Department | varchar(10) | NN |  | Department doctor belongs to |
| Salary | decimal | NN |  | Salary of the doctor |

Table 4.1‑2 Table: Patient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| PatientID | int | NN | PK (Auto Increment) | Unique Identifier for each patient. |
| Name | varchar(100) | NN |  | Name of the patient |
| Contact | number(10, 0) | AN |  | Contact information of patient |
| Email | varchar(20) | NN | UNIQUE | Email address of the patient |
| Address | varchar(100) | AN |  | Home address of the patient |
| DOB | Date | NN |  | Date of the birth of the patient |
| Age | int | AN |  | Age of the patient |
| Gender | varchar(100) | NN |  | Gender of the patient |
| Medical History | varchar(255) | AN |  | Medical History of the patient |

Table 4.1‑3 Table: Staff/Nurse

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| StaffID | int | NN | PK (Auto Increment) | Unique identifier for each staff |
| Name | varchar(100) | AN |  | Name of the Staff |
| DoctorID | int | AN | FK | Reference to Doctor Table |
| Department | varchar(10) | AN |  | Department the staff belongs |
| Salary | decimal | NN |  | Salary of the staff |

Table 4.1‑4 Table: Appointment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| AppoinmentID | int | NN | PK (Auto Increment) |  |
| PatientName | varchar(100) | NN |  |  |
| Gender | varchar(100) | AN |  |  |
| Date | DateTime | AN |  |  |
| Department | varchar(100) | NN |  |  |
| Contact | number(10,0) | AN |  |  |

Table 4.1‑5 Table: Bill

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| BillID | int | NN | PK (Auto Increment) |  |
| PatientID | Int | NN | FK | Reference to Patient Table |
| PatientName | varchar(100) | NN |  |  |
| Gender | varchar(100) | AN |  |  |
| Date | DateTime | NN |  |  |
| Amount | decimal | NN |  |  |
| Contact | number(10,0) | AN |  |  |

# Stories and Scenario

## Story-1: Schedule Appointments

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Patient,  I want to make an appointment with a doctor at specific date and time  So that I can receive timely medical attention |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | Booking appointments is critical for managing the patient flow and ensuring timely medical consultations. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Booking an Appointment with Valid Information. |
| Prerequisite | **:** | The patient is logged into the hospital management system. |
| Acceptance Criteria | **:** | **Given:** The patient is on the appointment booking page.  **When:**  The patient selects a doctor from the list.  The patient chooses a valid appointment date and time.  The patient clicks the “Confirm Appointment” button..  **Then:** The system successfully schedules the appointment, and the patient receives a confirmation message with the appointment details (doctor’s name, time, and reference number). |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Appointment Slot Not Available |
| Prerequisite | **:** | The patient is logged into the hospital management system. |
| Acceptance Criteria | **:** | **Given:** The patient selects a doctor and tries to book an appointment.  **When:** The chosen date and time slot is already taken by another patient and there is no slot left.  **Then:** The system displays an error message informing the patient that the slot is unavailable and suggests alternative dates and times. |

### Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.3* | : | Booking an Appointment with Missing Information |
| Prerequisite | **:** | The patient is logged into the hospital management system. |
| Acceptance Criteria | **:** | **Given**: The patient is on the appointment booking page.  **When**: The patient does not fill in all the required fields (e.g., doctor, date, time).  The patient clicks the “Confirm Appointment” button.  **Then**: The system displays an error message prompting the patient to complete the missing information before proceeding. |

## Story-2: PayBillsOnline

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Patient,  I want to pay for my medical consultation  So that I can complete the appointment |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Payments must be processed efficiently to ensure smooth booking and prevent delays in providing healthcare services. |

### Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Successful Payment with Valid Card Details |
| Prerequisite | **:** | The patient has booked an appointment and is on the payment page. |
| Acceptance Criteria | **:** | **Given:** The patient has entered valid payment information, including card details (card number, expiration date, CVV).  **When:**  The patient clicks the “Pay” button and procedure regarding payment starts.  **Then:** The system processes the payment, deducts the amount, and displays a confirmation message with the payment reference number and appointment details. |

### Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# 21.2* | : | Payment with Missing or Invalid Card Information |
| Prerequisite | **:** | The patient has booked an appointment and is on the payment page. |
| Acceptance Criteria | **:** | **Given:** The patient has not entered complete or valid card details (e.g., incorrect card number, expired card).  **When:** The patient clicks the “Pay” button and payment procedure does. Not start due to invalid card information.  **Then:** The system displays an error message highlighting the incorrect or missing information and instructs the patient to correct the details before retrying. |

## Story-3: View & Handling Feedback

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As an Admin,  I want to view, manage, and respond to patient feedback,  So that I can address concerns, improve service quality, and ensure patient satisfaction. |
| Priority | **:** | High |
| Estimate | **:** | L |
| Reason | **:** | Managing patient feedback helps improve healthcare services and maintain quality assurance. |

### Scenario# S3.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Viewing Patient Feedback. |
| Prerequisite | **:** | The admin is logged into the hospital management system. |
| Acceptance Criteria | **:** | **Given:** The admin navigates to the feedback management page.  **When:** The admin selects a specific patient or appointment from the feedback list.  The system retrieves and displays the feedback, including the patient’s comments and ratings.  **Then:** The admin can successfully view the feedback details. |

### Scenario# S3.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.2* | : | Responding to Feedback. |
| Prerequisite | **:** | The admin is logged into the hospital management system. |
| Acceptance Criteria | **:** | **Given:** The admin is on the feedback management page and has selected a feedback entry.  **When:** The admin types a response addressing the patient’s concern or appreciation.  The admin clicks the “Send Response” button.  **Then:** The system successfully records and sends the response to the patient, notifying them of the reply. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Hospital Management System | Test Designed by: | R. A. Savaliya |
| Module Name: | **Appointment Booking** | **Test Designed date:** | 01-10-2023 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 15-01-2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: The user should have the necessary permissions to book, reschedule, or cancel an appointment. | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | Schedule an appointment with valid details | Functional | Verify that a user can successfully schedule an appointment by providing a valid patient name, date, and time. | TC\_001 |
| TC\_002 | Reschedule an existing appointment | Functional | Verify that the user can successfully reschedule an appointment to a new date and time. | TC\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Schedule an appointment with valid details |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | The user should have the necessary permissions to book, reschedule, or cancel an appointment. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Open the appointment scheduling page | The page should load successfully with all input fields visible | Page loaded successfully | Pass |  | URL of the appointment page | - |
| 2 | Enter valid patient name, date, and time | System should allow input and accept valid details | Input accepted successfully | Pass |  | Patient: John Doe, Date: 2025-02-15, Time: 10:00 AM | - |
| 3 | Click on the "Book Appointment" button | Appointment should be booked and saved in the system | Appointment successfully booked | Pass |  | - | - |
| 4 | Verify the appointment appears in the "Upcoming Appointments" list | The booked appointment should be visible with correct details | Appointment is listed correctly | Pass |  | Displayed: John Doe, 2025-02-15, 10:00 AM | - |

|  |  |
| --- | --- |
| **Test Case Title** | Reschedule an existing appointment |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | The user should have the necessary permissions to book, reschedule, or cancel an appointment. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Open the "Upcoming Appointments" section | The list should display all scheduled appointments | Appointments are visible | Pass |  | - | - |
| 2 | Click on the "Reschedule" button for an existing appointment | The system should allow editing the appointment date and time | Reschedule option enabled | Pass |  | - | - |
| 3 | Select a new date and time for the appointment | The system should accept the new date and time | New date and time accepted | Pass |  | New Date: 2025-02-20, Time: 2:00 PM | - |
| 4 | Verify that the rescheduled appointment appears with updated details | The new date and time should be reflected in the appointment list | Appointment updated successfully | Pass | Check displayed details match the rescheduled data | Displayed: John Doe, 2025-02-20, 2:00 PM | - |

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Hospital Management System | Test Designed by: | R. A. Savaliya |
| Module Name: | **Patient Details** | **Test Designed date:** | 01-10-2023 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 15-01-2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: User should have the necessary permissions to view and add patient records. | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | Verify Viewing Patient Details | Functional | Ensures patient details are displayed correctly when viewing full details. | TC\_001 |
| TC\_002 | Verify Adding a New Patient | Functional | Checks if a new patient can be added successfully and appears in the records list. | TC\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Verify Viewing Patient Details |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | User should have the necessary permissions to view and add patient records.. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Open the Patient Records page | The page should load successfully with all patient details visible | Page loaded successfully | Pass |  | - | - |
| 2 | Click on the "View Full Details" button for a patient | System should open a detailed view of the selected patient | Detailed view opened successfully | Pass |  | Patient: John Doe | - |
| 3 | Verify patient details (Name, Age, Gender, Last Visit, Medical History) | The details should match the displayed records | Details match correctly | Pass |  | John Doe, 35, Male, 12 Jan 2025, Hypertension, Diabetes | - |
| 4 | Navigate back to the Patient Records page | The page should return to the main records screen | Navigation successful | Pass |  | - | - |

|  |  |
| --- | --- |
| **Test Case Title** | Verify Adding a New Patient |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | User should have the necessary permissions to view and add patient records.. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Click on the "Add New Patient" button | The system should open a form to add a new patient | Form opened successfully | Pass | Ensure all input fields are present | - | - |
| 2 | Enter valid patient details (Name, Age, Gender, Last Visit, Medical History) | The system should accept the entered details | Input accepted successfully | Pass | Ensure valid inputs are entered | Name: Alex Brown, Age: 30, Gender: Male | - |
| 3 | Click on the "Save" or "Submit" button | The system should save the patient data and return to the main records screen | Patient record added successfully | Pass | Check if the new record is saved | - | - |
| 4 | Verify the newly added patient appears in the Patient Records list | The newly added patient should be listed with correct details | Patient record displayed correctly | Pass | Cross-check displayed details with input data | Alex Brown, 30, Male | - |

# References

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* https://www.javatpoint.com/uml