ASSIGNMENT-5

1. Write project report on Customer Relationship Management using Agile.

---

1. Abstract

The Customer Relationship Management (CRM) System is designed to help organizations manage and analyze customer interactions throughout the customer lifecycle. The main objective of this project is to improve business relationships, assist in customer retention, and drive sales growth. The system centralizes customer information such as contact details, communication history, service requests, and feedback in a single platform.

The project is developed using the Agile Scrum methodology, which emphasizes iterative development, continuous user feedback, and incremental delivery of functional modules. The system is developed in multiple sprints, each delivering working features like customer registration, interaction tracking, lead management, and reporting dashboards. This allows stakeholders to review progress frequently and suggest changes early in the development cycle.

The CRM system improves operational efficiency by reducing paperwork, minimizing data duplication, and enabling faster access to customer information. It also enhances decision-making by providing analytical reports and trends based on customer behavior. The system supports role-based access so that sales, support, and management teams can access relevant data securely. This project demonstrates the effectiveness of Agile in developing scalable, flexible, and user-centric enterprise applications.

---

2. Introduction

2.1 Introduction

CRM is a business strategy and technology solution that helps organizations understand customer needs, manage interactions, and build long-term relationships. In competitive markets, maintaining customer satisfaction and loyalty is essential for business success.

2.2 Problem Identification

Many organizations manage customer data manually or across multiple disconnected systems, leading to data inconsistency, delayed responses, and poor customer experience.

2.3 Need of the Project

The need for a CRM system arises to automate customer data management, improve service response time, and support business decision-making through analytics.

2.4 Project Scheduling

The project is divided into sprints:

- Sprint 1: Requirement analysis and UI design
- Sprint 2: Customer and lead management module
- Sprint 3: Reporting and analytics
- Sprint 4: Testing and deployment

2.5 Objectives

- Centralize customer information
- Improve customer engagement
- Provide analytical insights
- Increase operational efficiency

---

3. Software Requirement Specification (SRS)

3.1 Purpose

To define functional and non-functional requirements for the CRM system.

3.2 Scope

The system will be used by sales, support, and management teams in an organization.

3.3 Hardware and Software Requirements

- Hardware: 4GB RAM, Intel i3 processor
- Software: Windows/Linux, Java, MySQL, Web Browser

3.4 Tools

Java, MySQL, HTML/CSS, GitHub, Eclipse

3.5 Software Process Model

Agile Scrum model with iterative sprints and continuous feedback.

---

4. SYSTEM DESIGN (CRM SYSTEM)

4.1 Data Dictionary

Entity	Attribute	Data Type	Description
Customer	ID	Integer	Unique customer ID
Customer	Name	String	Customer full name
Customer	Email	String	Customer email address
Customer	Phone	String	Contact number
Interaction	ID	Integer	Interaction ID
Interaction	Date	Date	Date of interaction
Interaction	Type	String	Call, Email, Meeting
Interaction	Notes	String	Description of interaction
SalesRep	ID	Integer	Sales representative ID
SalesRep	Name	String	Sales representative name

4.2 ER Diagram

Entities

- Customer
- Interaction
- Sales Representative

Relationships

- A Customer **can have many** Interactions

- A Sales Representative **handles many** Interactions

ER Diagram (ASCII Representation)

```
+-----------+      +--------------+      +------------------+
| Customer  |      | Interaction  |      | SalesRep         |
+-----------+      +--------------+      +------------------+
| ID (PK)   |<------ | ID (PK)     | ------>| ID (PK)        |
| Name      |  1   | Date        |   N  | Name           |
| Email     |      | Type        |      |                |
| Phone     |      | Notes       |      |                |
+-----------+      +--------------+      +------------------+
```

Meaning:
Customer (1) — (N) Interaction — (1) SalesRep

4.3 Data Flow Diagram (DFD)

DFD Level 0 (Context Diagram)

```
+-------+      +-----------------+      +-----------+
| User  | -----> |  CRM System   | -----> | Database  |
+-------+      +-----------------+      +-----------+
       <---------------------------------->
              Data / Reports
```

DFD Level 1

```
User
 |
 | 1. Login Request
 v
+-----------------+
| Authentication  |
+-----------------+
 |
 | 2. Customer Data
 v
+-----------------+
| Customer Module |
+-----------------+
 |
 | 3. Interaction Data
 v
+-------------------+
| Interaction Module|
+-------------------+
 |
 | 4. Reports
 v
+-----------------+
| Report Module   |
+-----------------+
 |
 Database
```

4.4 Use Case Diagram

Actors

- User (Admin / Sales Staff / Support Staff)

Use Cases

- Login
- Add Customer
- Update Customer
- Delete Customer
- View Customer
- Record Interaction
- Generate Reports

Use Case Diagram (ASCII)

```
+-------------------+
|      User         |
+-------------------+
   |  |  |  |
   |  |  |  |
   v  v  v  v
 +-------------------+
 |  CRM System       |
 +-------------------+
 | Login             |
 | Add Customer      |
 | Update Customer   |
 | Delete Customer   |
 | Record Interaction|
 | Generate Reports  |
 +-------------------+
```

Diagram Summary

Diagram	**Purpose**
Data Dictionary	Defines data elements
ER Diagram	Shows entity relationships
DFD	Shows data flow
Use Case Diagram	Shows user interaction

---

5. Implementation

5.1 Program Code

The CRM system is implemented using Java for backend development and MySQL as the database. Java handles business logic such as managing customer records, tracking interactions, generating reports, and validating user inputs. JDBC is used to connect Java with the MySQL database.

The application follows a modular structure, with separate modules for authentication, customer management, interaction tracking, and reporting. This structure improves maintainability and makes the system easier to update or extend in the future. Agile development is followed, and the system is developed in iterations with continuous testing and feedback.

5.2 Output Screens

The system includes the following main screens:

- Login page for user authentication
- Dashboard displaying system overview
- Customer form for adding and updating customer details
- Interaction form to record calls and meetings
- Reports screen for viewing analytics and summaries

All screens are designed to be simple and user-friendly.

---

6. Testing

6.1 Test Data

Test data includes valid customer records, invalid email formats, empty input fields, and duplicate entries. These inputs are used to verify system behavior under different conditions.

6.2 Test Result

The system was tested using unit and integration testing. All major functionalities worked as expected, and errors were handled correctly. The system was found stable and reliable.

---

7. User Manual

7.1 How to Use

Users log in to the system, add and manage customer records, track interactions, and generate reports through the dashboard interface.

7.2 Screen Layout

The layout is menu-driven with clearly labeled options and simple navigation, allowing users to perform tasks easily.

---

8. Applications and Limitations

The system is used in sales, marketing, and customer support departments. Limitations include dependence on internet connectivity and the need for basic user training.

---

9. Conclusion and Future Enhancement

The CRM system improves customer management and business efficiency. Future enhancements include mobile app integration, AI-based insights, and chatbot support.

2 Write project report on Write project report on Travel Booking Platform using Agile.

---

1. Abstract

The Travel Booking Platform is a web-based system designed to allow users to search, compare, and book travel services such as flights, hotels, and tour packages online. The system simplifies the travel planning process by providing a single platform where users can view available options, compare prices, and make secure bookings.

The project is developed using the Agile methodology, which supports iterative development, continuous feedback, and flexible requirement changes. The system is built in multiple sprints, each delivering functional components such as user registration, search functionality, booking management, and payment processing. This approach ensures early delivery of usable features and allows improvements based on user feedback.

The platform enhances user convenience, reduces manual booking errors, and improves service efficiency for travel agencies. It also provides administrative tools for managing inventory, pricing, and bookings.

---

2. Introduction

2.1 Introduction

With the growth of digital services, travelers prefer online platforms to plan and book their trips. A travel booking platform helps users access multiple travel services from one place, saving time and effort.

2.2 Problem Identification

Travelers often need to visit multiple websites to compare prices and availability. Manual booking processes are time-consuming and prone to errors.

2.3 Need of the Project

The project is needed to automate the travel booking process, provide real-time availability, and improve customer experience.

2.4 Project Scheduling

The project is developed in four sprints:

- Sprint 1: Requirement analysis and UI design

- Sprint 2: Search and booking module
- Sprint 3: Payment and confirmation module
- Sprint 4: Testing and deployment

2.5 Objectives

- Provide a centralized booking platform
- Enable real-time search and comparison
- Ensure secure online payments
- Improve user experience

---

3. Software Requirement Specification

3.1 Purpose

To define the functional and non-functional requirements of the travel booking platform.

3.2 Scope

The system is used by travelers, travel agencies, and administrators.

3.3 Hardware and Software Requirements

- Hardware: 4GB RAM, Internet-enabled PC
- Software: Windows/Linux, Java, MySQL, Browser

3.4 Tools

Java, MySQL, HTML/CSS, JavaScript, Git

3.5 Software Process Model

Agile Scrum Model.

Great 👍 — below is the **System Design section written in full detail** for your **Travel Booking Platform using Agile**.

You can directly replace **Section 4** in your report with this.

---

4. System Design — Travel Booking Platform

4.1 Data Dictionary

Entity	Attribute	Data Type	Description
User	UserID	Integer	Unique user identifier
User	Name	String	User full name
User	Email	String	User email
User	Phone	String	Contact number
TravelService	ServiceID	Integer	Flight/Hotel/Bus ID
TravelService	Type	String	Flight, Hotel, Bus
TravelService	Source	String	Departure city
TravelService	Destination	String	Arrival city
TravelService	Price	Double	Cost
Booking	BookingID	Integer	Booking identifier
Booking	Date	Date	Booking date
Booking	Status	String	Confirmed/Cancelled
Payment	PaymentID	Integer	Payment reference
Payment	Amount	Double	Paid amount
Payment	Mode	String	Card, UPI, Net Banking

4.2 ER Diagram

Entities

- User
- TravelService
- Booking
- Payment

Relationships

- A User can make many Bookings.

- A Booking is associated with one TravelService.
- Each Booking has one Payment.

ER Diagram (Text Representation)

```
+-------+     +---------+      +---------+      +---------+
| User  | 1   N | Booking | 1    1 | Payment |      | Travel  |
+-------+-------+---------+--------+---------+-------+ Service |
|UserID |     |BookingID|      |PaymentID|      |ServiceID|
|Name   |     |Date     |      |Amount   |      |Type     |
|Email  |     |Status   |      |Mode     |      |Source   |
+-------+     +---------+      +---------+      |Dest     |
                                |Price    |
                                +---------+
```

4.3 Data Flow Diagram (DFD)

DFD Level 0

User → Travel Booking System → Database

User ← Confirmation / Results ← System

DFD Level 1

```
User
 |
 v
+-------------------+
| Search Module     |
+-------------------+
 |
 v
+-------------------+
| Booking Module   |
+-------------------+
 |
 v
+-------------------+
| Payment Module   |
+-------------------+
 |
 v
Database
```

4.4 Use Case Diagram

Actors

- User
- Admin

Use Cases

Actor	Use Case
User	Register/Login
User	Search Travel
User	Book Service
User	Make Payment
User	View Booking
Admin	Manage Services
Admin	View Reports

Use Case Diagram (Textual)

User → Login → Search → Book → Pay → Confirm

Admin → Add Service → Update Price → View Bookings

4.5 System Architecture

The system follows a three-tier architecture:

1. Presentation Layer: Web UI using HTML/CSS/JavaScript
2. Application Layer: Java backend handling logic
3. Data Layer: MySQL database storing records

4.6 Module Description

User Module

Handles registration, login, and profile management.

Search Module

Allows users to search flights/hotels based on location and date.

Booking Module

Handles booking creation and status updates.

Payment Module

Processes secure payments and stores transaction records.

Admin Module

Manages travel services and pricing.

---

5. Implementation

5.1 Program Code

The backend is developed using Java, and MySQL is used for database management. The system includes modules for user management, travel search, booking processing, and payment handling.

5.2 Output Screens

Login page, search screen, booking form, payment screen, and confirmation page.

---

6. Testing

6.1 Test Data

Valid bookings, invalid card details, empty search queries.

6.2 Test Result

All modules passed testing and worked as expected.

---

7. User Manual

7.1 How to Use

Login → Search Travel → Select Option → Book → Pay → Receive Confirmation.

7.2 Screen Layout

Simple interface with navigation menu and search filters.

---

8. Applications and Limitations

Used by travelers and travel agencies. Limited by internet dependency and payment gateway availability.

---

9. Conclusion and Future Enhancement

The Travel Booking Platform simplifies travel planning and improves service efficiency. Future enhancements include mobile app integration, AI-based recommendations, and multilingual support.

---

3 Write project report on Online -commerce website using Agile.

1. Abstract

The Online E-Commerce Website is a web-based application that allows users to browse products, add items to a shopping cart, make online payments, and track orders. The system provides a convenient platform for customers to shop from anywhere and helps businesses reach a wider audience.

The project is developed using the Agile methodology, which focuses on iterative development, continuous testing, and user feedback. The application is built in multiple sprints, each delivering functional modules such as product management, shopping cart, payment gateway integration, and order tracking.

This platform improves customer experience, reduces manual order processing, and enhances business efficiency.

---

2. Introduction

2.1 Introduction

E-commerce platforms enable online buying and selling of goods and services. With increasing internet usage, online shopping has become an essential part of modern retail.

2.2 Problem Identification

Traditional shopping is time-consuming and limited by location. Manual order management is inefficient and prone to errors.

2.3 Need of the Project

The project is needed to provide a digital shopping platform that is fast, reliable, and secure.

2.4 Project Scheduling

- Sprint 1: Requirements and UI design
- Sprint 2: Product and cart modules
- Sprint 3: Payment and order tracking
- Sprint 4: Testing and deployment

2.5 Objectives

- Provide a user-friendly shopping experience
- Support secure online payments
- Automate order management
- Increase business reach

---

3. Software Requirement Specification

3.1 Purpose

To define system requirements for the e-commerce website.

3.2 Scope

Used by customers, sellers, and administrators.

3.3 Hardware and Software Requirements

- Hardware: 4GB RAM, PC or mobile device
- Software: Java, MySQL, Browser

3.4 Tools

Java, MySQL, HTML, CSS, JavaScript, Git

3.5 Software Process Model

Agile Scrum Model.

---

4. System Design (Detailed)

4.1 Data Dictionary

Entity	Attribute	Data Type	Description
User	UserID	Integer	Unique user ID
User	Name	String	Customer name
User	Email	String	Customer email
User	Password	String	Encrypted password
Product	ProductID	Integer	Unique product ID

Entity	**Attribute**	**Data Type**	**Description**
Product	Name	String	Product name
Product	Category	String	Product category
Product	Price	Double	Product price
Product	Stock	Integer	Available quantity
Order	OrderID	Integer	Order number
Order	Date	Date	Order date
Order	Status	String	Placed/Shipped/Delivered
Payment	PaymentID	Integer	Transaction reference
Payment	Mode	String	Card/UPI/NetBanking
Payment	Amount	Double	Paid amount

4.2 ER Diagram

Entities:

- User
- Product
- Order
- Payment

Relationships:

- A User can place many Orders.
- An Order contains many Products.
- Each Order has one Payment.

ER Diagram (Textual)

```
User (1) —— (N) Order —— (1) Payment

         |

         |
      (N)
     Product
```

4.3 Data Flow Diagram (DFD)

DFD Level 0

Customer → E-Commerce System → Database

Customer ← Order Status / Confirmation ← System

DFD Level 1

```
Customer
 |
 v
+-----------------+
| Product Search  |
+-----------------+
 |
 v
+-----------------+
| Shopping Cart   |
+-----------------+
 |
 v
+-----------------+
| Payment Module  |
+-----------------+
 |
 v
Database
```

4.4 Use Case Diagram

Actors:

- Customer
- Admin

Use Cases:

Customer:

- Register/Login
- Browse Products
- Add to Cart
- Place Order
- Make Payment
- Track Order

Admin:

- Add Product
- Update Product
- Manage Orders
- View Reports

4.5 System Architecture

The system follows a three-tier architecture:

1. Presentation Layer: Web interface (HTML, CSS, JS)
2. Application Layer: Java backend logic
3. Data Layer: MySQL database

4.6 Module Description

User Module

Handles customer registration, login, and profile management.

Product Module

Manages product listings, categories, pricing, and stock.

Cart Module

Handles adding/removing products before checkout.

Order Module

Manages order creation and status updates.

Payment Module

Handles secure transactions.

Admin Module

Controls inventory, orders, and reports.

---

5. Implementation

Backend implemented using Java, frontend using HTML/CSS/JS, and MySQL for data storage.

---

6. Testing

System tested using valid and invalid data; all modules passed successfully.

---

7. User Manual

Login → Browse → Add to Cart → Checkout → Pay → Track Order

---

8. Applications and Limitations

Used by retailers and customers. Limited by internet availability and payment gateway dependency.

---

9. Conclusion and Future Enhancement

The system simplifies online shopping and improves business efficiency. Future enhancements include mobile apps, AI recommendations, and voice-based search.

---

4. Write project report on Automate Banking website using Agile.

1. Abstract

The Automated Banking Website is a web-based system designed to provide customers with online access to essential banking services such as account management, fund transfer, balance enquiry, bill payments, and transaction history. The system reduces dependency on physical bank visits and improves customer convenience.

The project is developed using the Agile methodology, which supports iterative development, continuous testing, and flexibility to incorporate changing requirements. The system is built in multiple sprints, each delivering functional modules such as user authentication, account services, transaction processing, and security controls. This approach ensures early delivery of usable features and continuous improvement.

The system enhances operational efficiency for banks, reduces manual workload, and improves service quality while maintaining security and reliability.

---

2. Introduction

2.1 Introduction

Modern banking increasingly relies on digital platforms to deliver fast and secure services. Automated banking systems allow customers to access their accounts anytime and perform transactions online.

2.2 Problem Identification

Traditional banking requires customers to visit branches for basic services, causing delays and inconvenience. Manual processing also increases the risk of errors and inefficiency.

2.3 Need of the Project

The project is needed to provide a secure, fast, and accessible online banking platform that reduces branch workload and improves customer satisfaction.

2.4 Project Scheduling

- Sprint 1: Requirement analysis and UI design
- Sprint 2: User authentication and account module
- Sprint 3: Fund transfer and bill payment module

- Sprint 4: Security testing and deployment

2.5 Objectives

- Enable online banking services
- Ensure secure transactions
- Reduce manual processes
- Improve customer convenience

---

3. Software Requirement Specification

3.1 Purpose

To define system requirements for the automated banking website.

3.2 Scope

The system is used by bank customers, bank administrators, and support staff.

3.3 Hardware and Software Requirements

- Hardware: 4GB RAM, Internet-enabled device
- Software: Java, MySQL, Browser

3.4 Tools

Java, MySQL, HTML, CSS, JavaScript, Git

3.5 Software Process Model

Agile Scrum Model.

---

4. System Design (Detailed)

4.1 Data Dictionary

Entity	Attribute	Data Type	Description
User	UserID	Integer	Unique customer ID
User	Name	String	Customer name
User	Email	String	Customer email

Entity	Attribute	Data Type	Description
Account	AccountNo	Integer	Bank account number
Account	Balance	Double	Current balance
Transaction	TxnID	Integer	Transaction ID
Transaction	Amount	Double	Transfer amount
Transaction	Type	String	Debit/Credit
Transaction	Date	Date	Transaction date
BillPayment	BillID	Integer	Utility bill ID
BillPayment	Provider	String	Electricity/Water/etc
BillPayment	Amount	Double	Paid amount

4.2 ER Diagram

Entities: User, Account, Transaction, BillPayment

Relationships:

- User owns Account
- Account has Transactions
- Account makes BillPayments

```
User (1) —— (1) Account —— (N) Transaction

             |

             |

            (N)

          BillPayment
```

4.3 Data Flow Diagram (DFD)

Level 0

User → Banking System → Database

User ← Confirmation / Balance ← System

Level 1

User

|

v

Authentication Module

|

v

Account Services

|

v

Transaction Processing

|

v

Database

4.4 Use Case Diagram

Actors: Customer, Admin

Customer:

- Login
- View Balance
- Transfer Funds
- Pay Bills
- View Transactions

Admin:

- Manage Users
- Monitor Transactions

4.5 System Architecture

Three-tier architecture:

1. UI Layer (Web Interface)
2. Application Layer (Java Backend)
3. Data Layer (MySQL Database)

4.6 Module Description

User Module

Handles registration, login, and authentication.

Account Module

Displays account details and balance.

Transaction Module

Processes fund transfers securely.

Bill Payment Module

Allows customers to pay utility bills.

Admin Module

Monitors system activity and manages users.

---

5. Implementation

Backend developed using Java, frontend using HTML/CSS/JS, database using MySQL.

---

6. Testing

Tested with valid and invalid inputs. All transactions were processed accurately.

---

7. User Manual

Login → View Balance → Transfer Funds / Pay Bills → View Transactions → Logout.

---

8. Applications and Limitations

Used by bank customers and staff. Limited by internet availability and cybersecurity threats.

---

9. Conclusion and Future Enhancement

The system improves banking efficiency and accessibility. Future enhancements include biometric authentication, mobile app integration, and AI fraud detection.

5 Write project report on Healthcare System using Agile.

1. Abstract

The Healthcare Management System is a web-based application designed to manage healthcare operations such as patient registration, appointment scheduling, doctor management, medical records, and billing. The system aims to improve the efficiency and accuracy of healthcare services while reducing manual paperwork and administrative workload.

The project is developed using the Agile software development methodology, which emphasizes iterative development, continuous feedback, and flexibility in handling changing requirements. The system is built in multiple sprints, each delivering functional modules such as patient management, appointment scheduling, electronic medical records, and reporting. This approach ensures early delivery of usable features and continuous improvement based on feedback from healthcare professionals.

The system enhances patient care by providing quick access to medical records, improves hospital efficiency, and supports better decision-making through accurate data management.

---

2. Introduction

2.1 Introduction

Healthcare organizations handle large volumes of patient data and administrative processes. An automated healthcare system helps manage these operations efficiently and improves the quality of healthcare services.

2.2 Problem Identification

Manual record-keeping leads to data inconsistency, delayed access to patient information, and increased administrative effort. Appointment scheduling and billing processes are often inefficient.

2.3 Need of the Project

The project is needed to digitize healthcare operations, reduce errors, and provide a centralized system for managing patients, doctors, and medical records.

2.4 Project Scheduling

- Sprint 1: Requirement analysis and system design
- Sprint 2: Patient and doctor management modules
- Sprint 3: Appointment scheduling and billing

- Sprint 4: Testing and deployment

2.5 Objectives

- Automate patient record management
- Improve appointment scheduling
- Enhance data accuracy and security
- Improve overall healthcare efficiency

---

3. Software Requirement Specification

3.1 Purpose

To define the functional and non-functional requirements of the healthcare management system.

3.2 Scope

The system is used by patients, doctors, hospital administrators, and support staff.

3.3 Hardware and Software Requirements

- Hardware: 4GB RAM, Internet-enabled device
- Software: Java, MySQL, Web Browser

3.4 Tools

Java, MySQL, HTML, CSS, JavaScript, Git

3.5 Software Process Model

Agile Scrum Model.

---

4. System Design (Detailed)

4.1 Data Dictionary

Entity	Attribute	Data Type	Description
Patient	PatientID	Integer	Unique patient identifier
Patient	Name	String	Patient full name

Entity	Attribute	Data Type	Description
Patient	Age	Integer	Patient age
Patient	Gender	String	Male/Female/Other
Patient	Contact	String	Contact number
Doctor	DoctorID	Integer	Unique doctor ID
Doctor	Name	String	Doctor name
Doctor	Specialization	String	Medical specialty
Appointment	AppointmentID	Integer	Appointment reference
Appointment	Date	Date	Appointment date
Appointment	Status	String	Scheduled/Completed
MedicalRecord	RecordID	Integer	Medical record ID
MedicalRecord	Diagnosis	String	Diagnosis details
MedicalRecord	Prescription	String	Prescribed medicines
Billing	BillID	Integer	Billing reference
Billing	Amount	Double	Bill amount
Billing	Status	String	Paid/Unpaid

4.2 ER Diagram

Entities:

- Patient
- Doctor
- Appointment
- MedicalRecord
- Billing

Relationships:

- Patient books Appointments

- Doctor attends Appointments
- Appointment generates MedicalRecord
- Appointment generates Billing

```
Patient (1) —— (N) Appointment —— (1) Doctor

          |

          |

          v

      MedicalRecord

          |

          v

        Billing
```

4.3 Data Flow Diagram (DFD)

DFD Level 0

Patient / Doctor → Healthcare System → Database

Patient / Doctor ← Records / Confirmation ← System

DFD Level 1

Patient

|

v

Patient Registration

|

v

Appointment Scheduling

|

v

Medical Records Management

|

v

Billing Module

|

v

Database

4.4 Use Case Diagram

Actors:

- Patient
- Doctor
- Admin

Use Cases:

Patient:

- Register/Login
- Book Appointment
- View Reports

Doctor:

- View Appointments
- Update Medical Records

Admin:

- Manage Users
- Generate Reports

4.5 System Architecture

The system follows a three-tier architecture:

1. Presentation Layer: Web interface
2. Application Layer: Java backend logic
3. Data Layer: MySQL database

4.6 Module Description

Patient Module

Handles patient registration and profile management.

Doctor Module

Manages doctor details and schedules.

Appointment Module

Schedules and tracks patient appointments.

Medical Records Module

Stores diagnoses and prescriptions securely.

Billing Module

Handles billing and payment records.

Admin Module

Manages system users and reports.

---

5. Implementation

The system is implemented using Java for backend processing, HTML/CSS/JavaScript for frontend, and MySQL for database management.

---

6. Testing

Test cases include valid and invalid patient entries, appointment conflicts, and billing errors. The system passed unit and integration testing.

---

7. User Manual

Login → Register Patient → Book Appointment → Record Diagnosis → Generate Bill → Logout.

---

8. Applications and Limitations

Used in hospitals, clinics, and diagnostic centers. Limited by internet connectivity and data privacy concerns.

---

9. Conclusion and Future Enhancement

The healthcare management system improves operational efficiency and patient care. Future enhancements include telemedicine support, AI-based diagnostics, mobile app integration, and cloud storage.
