Software Requirements Specification

For

Healthcare System

Version 1.0 approved

Prepared by Shejal Dhadse

HCL Training

Date: 10/01/2026

Submitted in partial fulfillment
of the requirements of
Software Engineering (Agile Methodology)

Abstract

The Healthcare System project focuses on the design and development of a secure, scalable, and efficient digital platform to manage healthcare services and patient information using modern software engineering practices. In today’s healthcare environment, hospitals and clinics face challenges such as manual record handling, delayed access to patient data, lack of coordination between departments, and inefficiencies in appointment and treatment management. These issues can lead to errors, increased workload, and reduced quality of patient care. This project aims to address these challenges by providing an integrated and automated healthcare management solution.

The proposed system enables patients to register, book appointments, access medical records, view prescriptions, and receive notifications digitally. Doctors can manage patient histories, update diagnoses, prescribe medications, and monitor treatment progress through a centralized system. Administrative staff can efficiently manage doctor schedules, patient records, billing, and reports. Role-based access control ensures that patients, doctors, and administrators access only authorized information, maintaining data confidentiality and system security.

The system is developed with a focus on reliability, data accuracy, and ease of use. It supports real-time access to patient information, reducing dependency on paper-based records and minimizing the risk of data loss or duplication. Secure authentication, data encryption, and access controls are implemented to protect sensitive medical information and comply with healthcare data protection requirements.

The architecture of the Healthcare System is designed to support scalability and future growth. It can be extended to include features such as telemedicine, online consultations, electronic health records (EHR) integration, lab report management, and advanced analytics for decision support. Overall, this project demonstrates how a well-designed healthcare management system can improve operational efficiency, enhance patient experience, and support healthcare providers in delivering timely and high-quality medical services.

Table of Contents

List of Figures

List of Tables

2. Introduction

2.1 Introduction

The Healthcare System project aims to develop a digital platform that simplifies and automates healthcare management processes. Traditional healthcare systems rely heavily on manual record keeping, which often leads to data loss, delays, and errors. With increasing patient load and demand for quality healthcare services, there is a strong need for a reliable and centralized system. This project provides an efficient solution for managing patient records, appointments, and healthcare services through a secure online system.

2.2 Problem Identification

Healthcare organizations face several challenges due to manual and fragmented systems. Patient records are often stored in paper format or isolated systems, making data retrieval slow and inefficient. Appointment scheduling, billing, and report management require significant manual effort, increasing workload and chances of errors. Lack of real-time access to patient information affects decision-making and quality of care.

Example:
A doctor may not get complete patient history on time due to missing or misplaced records.

2.3 Need of the Project

The need for a Healthcare System arises from the requirement to improve efficiency, accuracy, and accessibility of healthcare services. A centralized digital system helps in maintaining accurate patient data, reducing administrative workload, and improving coordination among doctors, staff, and patients.

Example:
Patients can book appointments online and access their medical reports without visiting the hospital multiple times.

2.4 Project Scheduling

The project follows an Agile-based schedule, where development is divided into short sprints. Each sprint focuses on delivering specific features such as patient registration, appointment management, or report handling. Regular reviews and testing ensure timely delivery and quality output.

2.5 Objectives

- Automate patient and appointment management
- Maintain accurate and secure medical records
- Improve communication between patients and healthcare providers
- Reduce manual work and operational errors
- Support future enhancements and system scalability

3. Software Requirement Specification (SRS)

3.1 Purpose

The purpose of this document is to define the functional and non-functional requirements of the Healthcare System. It serves as a reference for developers, testers, and stakeholders to understand system behavior, scope, and constraints. This document ensures that all requirements are clearly defined before development begins.

3.2 Scope

The Healthcare System provides an integrated platform for managing patient information, appointments, medical records, and administrative activities. The system supports doctors, patients, and administrative staff through role-based access.

System Scope Includes:

- Patient registration and profile management
- Online appointment scheduling
- Electronic medical records management
- Doctor and staff management
- Secure access and data handling

3.3 Hardware Requirement / Software Requirement

S. No	Hardware Component	Minimum Requirement
1	**Processor**	**Intel i3 or higher**
2	**RAM**	**4 GB**
3	**Hard Disk**	**100 GB**
4	**Internet**	**Broadband Connection**
5	**Display**	**14 inch or higher**

Hardware Requirements (Minimum):

- Processor: Intel i3 or higher
- RAM: 4 GB
- Hard Disk: 100 GB
- Internet Connection

S. No	Software Component	Specification
1	**Operating System**	**Windows / Linux**
2	**Frontend**	**HTML, CSS, JavaScript**
3	**Backend**	**Java / Python / PHP**
4	**Database**	**MySQL**
5	**Web Browser**	**Chrome / Firefox**

Software Requirements:

- Operating System: Windows / Linux
- Web Browser: Chrome / Firefox
- Backend: Java / Python / PHP
- Database: MySQL

Frontend: HTML, CSS, JavaScript3.4 Tools

- IDE: VS Code / Eclipse
- Database Tool: MySQL Workbench
- Version Control: Git
- Testing Tool: Manual Testing / Selenium
- Documentation Tool: MS Word / Canva

S. No	Tool Name	Purpose
1	VS Code	Code Development
2	MySQL Workbench	Database Management
3	Git	Version Control
4	Canva	Documentation
5	Selenium	Testing

3.5 Software Process Model

The project follows the **Agile Software Development Model**. Development is divided into small iterations called sprints. Each sprint delivers a working module and incorporates feedback for improvement. Agile helps in flexibility, faster delivery, and better quality control.

4. System Design

System Design defines the overall architecture and data flow of the Healthcare System. It explains how different components interact with each other to deliver required functionalities. A well-structured design improves system reliability, scalability, and maintainability.

S. No	User Role	Responsibilities
1	Admin	Manage users, doctors, reports
2	Doctor	View appointments, update records
3	Patient	Book appointments, view reports

4.1 Data Dictionary

The Data Dictionary describes important data elements used in the system along with their meaning and data type.

Key Data Elements:

- **Patient_ID** – Unique identification number for each patient
- **Patient_Name** – Full name of the patient
- **Doctor_ID** – Unique identifier for doctors
- **Appointment_ID** – Unique appointment reference
- **Appointment_Date** – Scheduled consultation date
- **Medical_Record** – Patient diagnosis and treatment details
- **User_Role** – Role of user (Admin, Doctor, Patient)

The data dictionary ensures consistency and accuracy in database design.

4.2 ER Diagram

The Entity Relationship (ER) Diagram represents the logical structure of the database.

Main Entities:

- Patient
- Doctor
- Appointment
- Medical Record
- Admin

Relationships:

- One patient can have multiple appointments
- One doctor can attend many patients
- Each appointment is linked to one medical record

The ER Diagram helps in defining database tables and relationships clearly.

4.3 Data Flow Diagram (DFD)

The Data Flow Diagram shows the movement of data within the system.

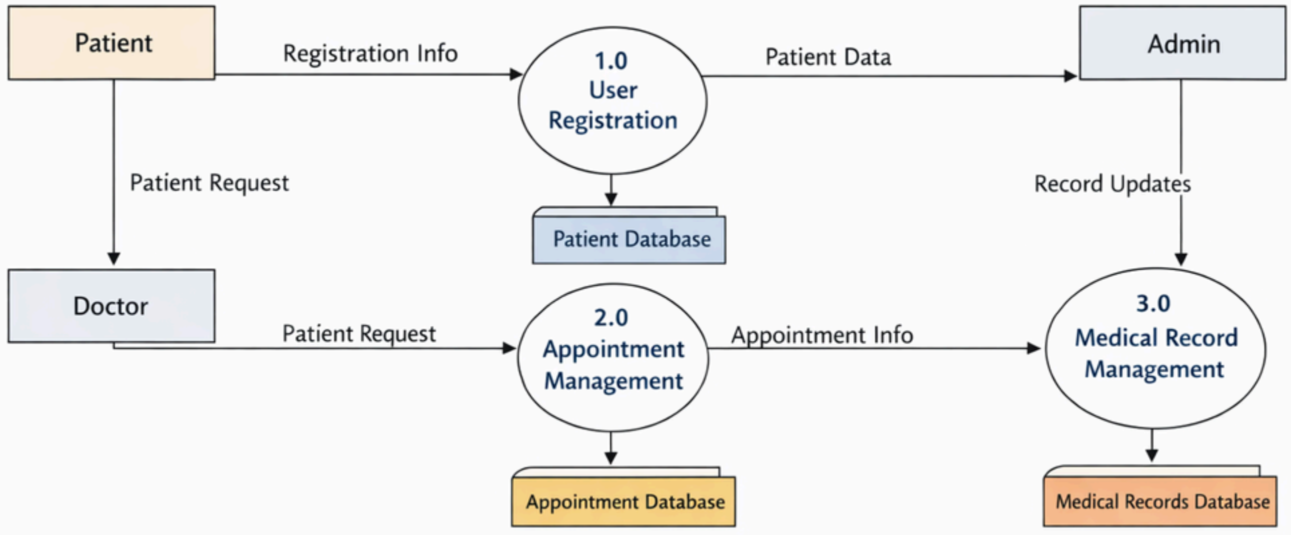
DFD Level 0 (Context Diagram):

- Users interact with the Healthcare System
- System processes data and stores it in the database

DFD Level 1:

- Patient registration process
- Appointment booking process
- Medical record update process
- Billing and report generation

DFD improves understanding of system functionality and data processing.

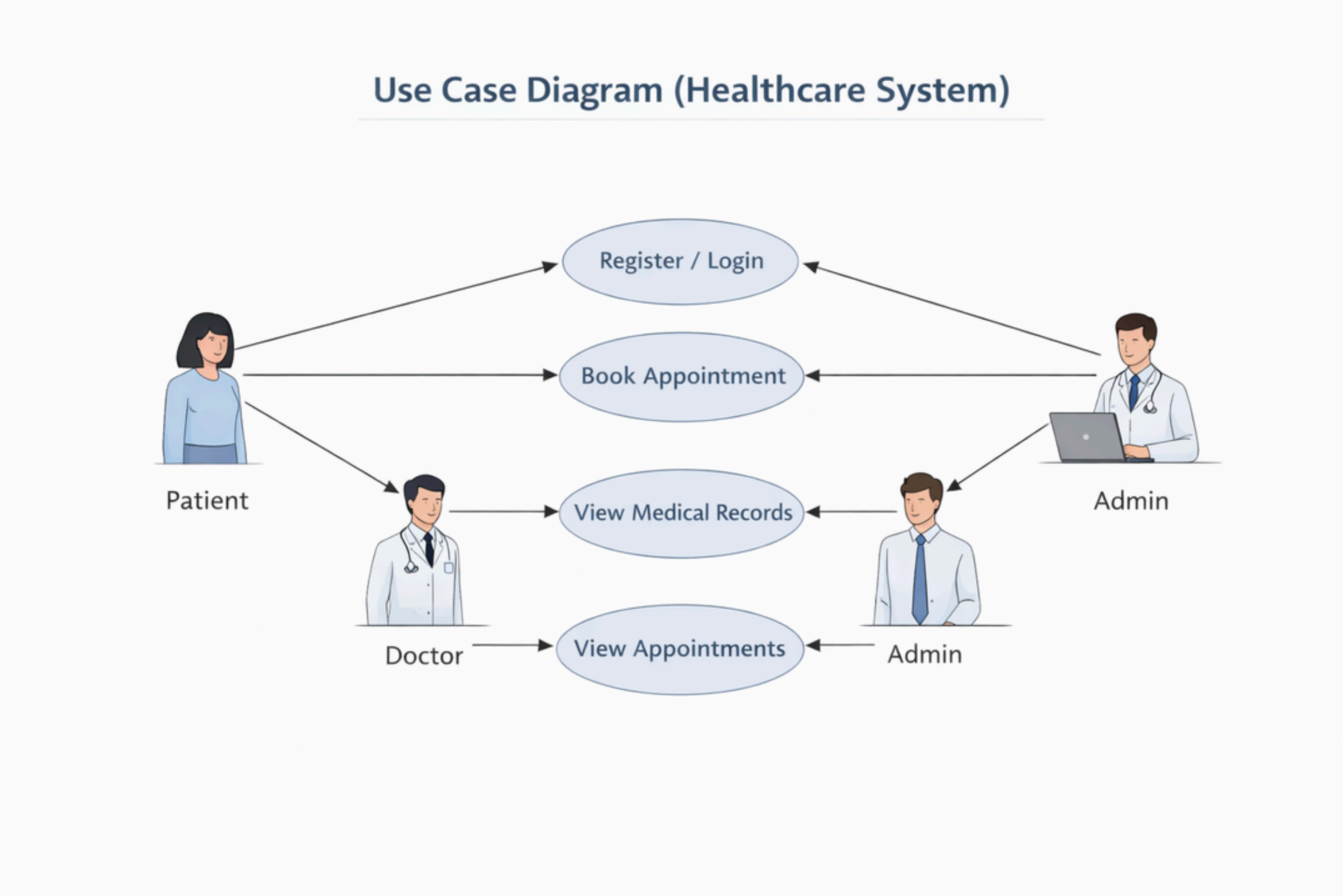


DFD Level 1 — Healthcare System

4.4 System Flow Chart / Use Case Diagram

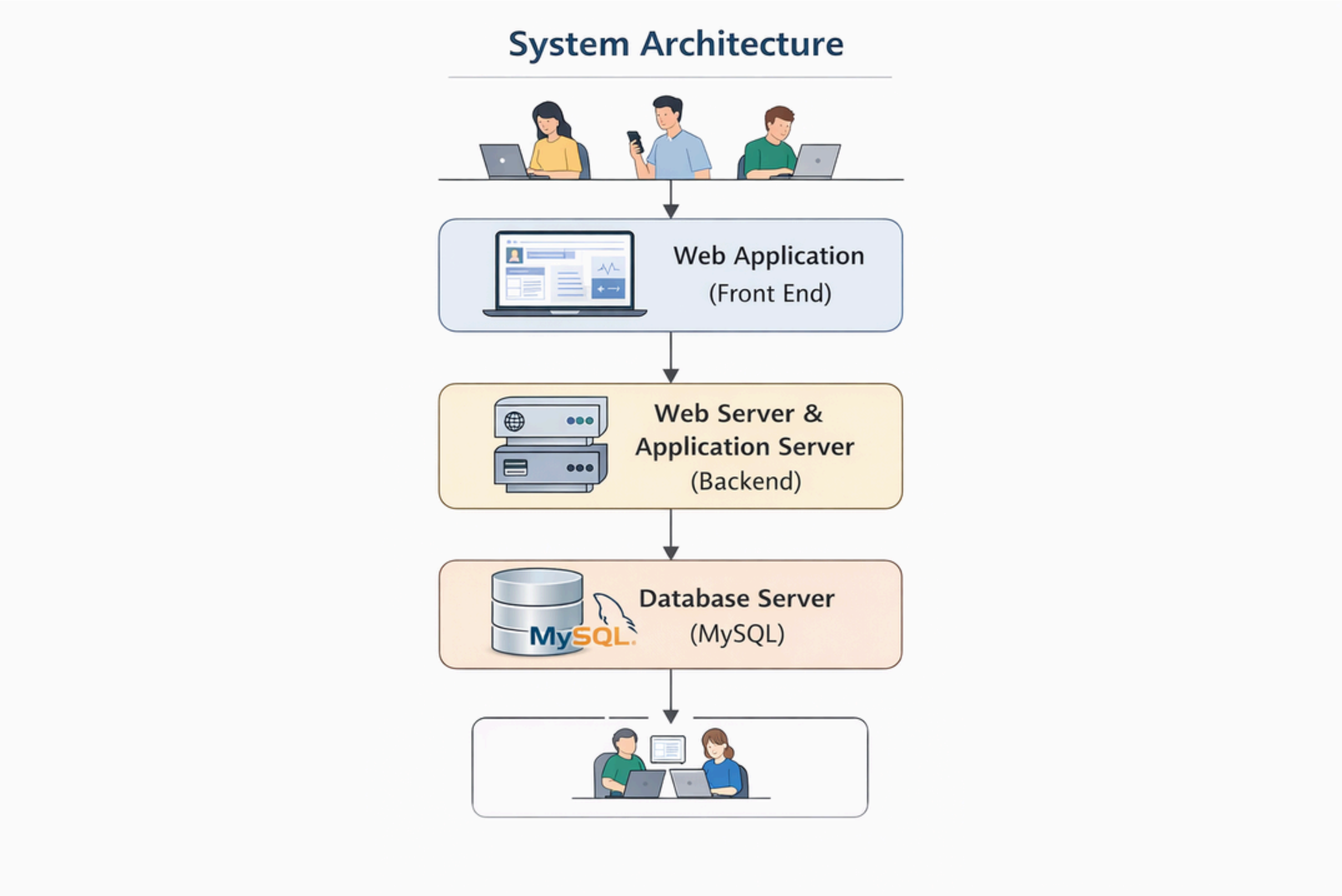
Use Case Diagram:

- Patient: Register, Login, Book Appointment, View Reports
- Doctor: View Appointments, Update Medical Records
- Admin: Manage Users, Doctors, System Reports



System Flow Chart:

1. User Login
2. Role Verification
3. Access Dashboard
4. Perform Assigned Operations
5. Logout



These diagrams provide a clear visual representation of system behavior.

5. Implementation

The implementation phase converts the system design into a working Healthcare System. In this phase, all planned modules are developed, integrated, and configured according to the requirements defined in the SRS. Agile methodology is followed, where features are implemented in small increments and reviewed at the end of each sprint.

The system is implemented as a web-based application to ensure easy access, scalability, and centralized data management.

5.1 Program Code

The application is divided into multiple modules, and each module is implemented separately to maintain clean and modular architecture.

Major Implemented Modules:

- User Authentication Module
- Patient Management Module
- Doctor Management Module
- Appointment Scheduling Module
- Medical Records Management Module
- Admin Dashboard Module

Implementation Approach:

- Frontend handles user interaction and validation
- Backend processes business logic and controls data flow
- Database stores structured healthcare data securely

Each module follows standard coding practices to ensure readability, maintainability, and future enhancement. Agile sprints ensure that tested code is delivered continuously

5.2 Output Screens

The system provides different output screens based on user roles.

Main Output Screens:

- Login and Registration Screen
- Patient Dashboard

S. No	Screen Name	Description
1	Login Screen	User authentication
2	Patient Dashboard	Appointment & reports
3	Doctor Dashboard	Patient management
4	Admin Panel	System control

- Doctor Dashboard
- Appointment Booking Screen
- Medical Record View Screen
- Admin Control Panel

Each screen is designed with a simple and user-friendly interface. Role-based access ensures that users can only view and modify permitted information, improving system security and usability.

6. Testing

Testing ensures that the Healthcare System works as expected and meets all functional and non-functional requirements. In Agile methodology, testing is performed continuously during each sprint to detect defects early and reduce overall risk. Both functional and basic non-functional checks are performed to validate system quality.

6.1 Test Data

Test data is prepared to verify different system scenarios and user roles.

Sample Test Data:

- Patient Login Credentials (valid and invalid)
- Doctor Profiles with different specializations
- Appointment Dates (past, current, future)
- Medical Records with sample diagnosis and prescriptions
- Admin credentials for system management

Test data covers both normal and boundary conditions to ensure system stability.

Test ID	Module	Test Description	Expected Result	Status
TC01	Login	Valid user login	Successful login	Pass
TC02	Appointment	Book appointment	Appointment confirmed	Pass
TC03	Records	View medical record	Record displayed	Pass

6.2 Test Result

Testing results confirm that the system performs according to requirements.

Test Results Summary:

- User authentication works correctly for all roles
- Appointment booking and cancellation functions correctly
- Medical records are stored and retrieved accurately
- Role-based access control is enforced properly
- No critical defects found during final testing

Minor issues identified during testing were fixed within the same sprint. Overall, the system achieved a high level of reliability and functional correctness.

7. User Manual

The User Manual provides clear instructions for using the Healthcare System. It helps users understand system features and perform tasks efficiently without technical knowledge. The system is designed to be user-friendly and role-based.

7.1 How to Use Project Guidelines

General Guidelines:

- Open the system using a supported web browser
- Login using valid username and password
- Access features based on assigned role
- Logout after completing tasks

User Roles and Usage:

- **Patient:** Register, login, book appointments, view medical records
- **Doctor:** View appointments, update patient diagnosis
- **Admin:** Manage users, doctors, and system data

7.2 Screen Layouts and Description

Login Screen:
Allows users to authenticate securely.

Patient Dashboard:
Displays appointments, reports, and profile details.

Doctor Dashboard:
Shows scheduled appointments and patient records.

Admin Dashboard:
Provides system monitoring and user management tools.

Each screen uses clear labels and navigation to ensure ease of use.

8. Project Applications and Limitations

This section explains where the Healthcare System can be applied in real-world scenarios and also highlights its current limitations.

Project Applications

The Healthcare System can be effectively used in various healthcare environments.

Applications:

- Hospitals for patient and appointment management
- Clinics for maintaining electronic medical records
- Diagnostic centers for report access and scheduling
- Healthcare administrators for system monitoring

Example:
A clinic can reduce manual paperwork by managing patient records digitally.

Project Limitations

Despite its advantages, the system has certain limitations.

Limitations:

- Requires stable internet connectivity
- Limited support for advanced analytics
- No built-in mobile application in current version
- Integration with external medical devices is limited

These limitations can be addressed in future enhancements.

9. Conclusion and Future Enhancement

Conclusion

The Healthcare System project successfully demonstrates the development of a secure, scalable, and user-friendly healthcare management platform using Agile methodology. The system automates key healthcare operations such as patient registration, appointment scheduling, medical record management, and administrative control. By replacing manual processes with digital workflows, the system improves efficiency, accuracy, and accessibility. The Agile development approach enabled incremental delivery, continuous testing, and regular feedback, resulting in better software quality and reduced development risk. The system meets both functional and non-functional requirements and is suitable for real-world healthcare environments.

Future Enhancement

The system can be enhanced further to support advanced healthcare needs.

Future Enhancements:

- Mobile application integration
- Online video consultation support
- AI-based health recommendations
- Integration with laboratory systems
- Advanced analytics and reporting

These enhancements will improve system usability, reach, and intelligence.

S. No	Enhancement	Description
1	Mobile App	Android & iOS support
2	Video Consultation	Online doctor visit
3	AI Analytics	Smart health insights
4	Lab Integration	Report automation

10. Bibliography & References

1. Pressman, R. S., *Software Engineering: A Practitioner's Approach*, McGraw-Hill.
2. Sommerville, I., *Software Engineering*, Pearson Education.
3. Schwaber, K., *Agile Project Management with Scrum*, Microsoft Press.
4. IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*.
5. Healthcare Information Systems – Online Research Articles.
6. Agile Software Development Methodology – Industry White Papers.
7. Web-based Healthcare System Case Studies.