***Software Requirement Specification Document***

**For project ON**

**“Online Healthcare Management System”**

**By -**:**Ashmit Gupta**

## Submitted To: Dr. Chandani Joshi

**TABLE OF CONTENTS**

1. SRS DEFINITION
2. INTRODUCTION
   1. Purpose
   2. Scope
   3. Definitions, Acronyms and Abbreviations
   4. References
   5. Overview
3. OVERALL DESCRIPTION
   1. Product Perspectives
      1. System interfaces
      2. User interfaces
      3. Memory constraints
      4. Operations
   2. Product Functions
   3. User Characteristics
   4. Hardware Constraints
   5. Assumptions & Dependencies
4. SPECIFIC REQUIREMENTS
   1. External Interface
      1. User interface
      2. Hardware Interface
      3. Software Interface
      4. Communication Interface
   2. Functional Requirement
   3. Performance Requirements
   4. Design Constraints
      1. Accuracy
      2. Future Enhancements
      3. User-Friendly Interface
      4. User Satisfaction
      5. Safety & Security
   5. Software System Attributes
      1. Security
      2. Reliability
      3. Maintainability
      4. Flexibility
      5. Availability
5. APPENDIX

# SRS Definition

A Software Requirements Specification (SRS) is a complete description of the behavior of the system to be developed. It also provides complete description of requirements for software.

It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. Use cases are also known as [functional requirements](http://en.wikipedia.org/wiki/Functional_requirements). In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. [Non-functional](http://en.wikipedia.org/wiki/Non-functional_requirements) [requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

# Introduction

The **Online Healthcare Management System** is a web-based platform designed to transform and digitize the traditional healthcare services provided by clinics, hospitals, and medical professionals. With the rising demand for streamlined, patient-centric services, this system ensures that healthcare delivery becomes more accessible, efficient, and transparent. The platform enables patients to register, book appointments, consult doctors remotely, receive prescriptions, and maintain their medical history digitally. Doctors benefit from tools to manage appointments, access patient records, and prescribe medications securely. Administrators can monitor overall system activities, generate reports, and manage users.

.

## Purpose

The purpose of this Software Requirements Specification (SRS) is to define the functionality, features, and constraints of the **Online Healthcare Management System (OHMS)**. This system is designed to serve as a comprehensive, user-friendly, and secure digital solution that addresses the operational needs of modern healthcare facilities, including hospitals, clinics, and telemedicine providers. This document establishes a mutual understanding between developers, project stakeholders, testers, and end-users regarding the system’s capabilities. It serves as a contractual foundation for development, ensuring all requirements are clearly identified, traceable, and testable throughout the software lifecycle.

## Scope

The Online Healthcare Management System will provide:

* Patient registration and profile management
* Appointment scheduling and reminders
* Doctor portal with medical record access
* Prescription and diagnostic history management
* Secure chat and consultation feature
* Admin control for managing users, data, and system settings

### Benefits

* Centralized patient data management
* Easy and flexible appointment scheduling
* Improved doctor-patient communication
* Reduced administrative workload
* Secure, encrypted, and role-based access
* 24/7 system availability
* Scalable and adaptable for any healthcare facility
* Real-time notifications and reminders

**Definitions,Acronyms, Abbreviations**

**Definitions**

* **Patient** – End-user receiving healthcare services
* **Doctor** – Medical professional providing consultation and prescriptions
* **Admin** – System manager controlling users, roles, and settings

**Acronyms**

* EMR – Electronic Medical Record
* OTP – One-Time Password
* UI – User Interface
* API – Application Programming Interface
* DBMS – Database Management System

## References

https://www.healthit.gov/topic/health-it-and-health-information-exchange-basics

https://www.geeksforgeeks.org/hospital-management-system-project/

<https://www.ncbi.nlm.nih.gov/books/NBK305666/>

## 2.5. Overview

The Online Healthcare Management System is a comprehensive web-based solution aimed at streamlining healthcare services for patients, doctors, and administrators. It provides an integrated platform that enables users to manage appointments, access electronic medical records (EMR), communicate securely, and handle administrative tasks with ease. The system is designed to reduce manual workload, minimize errors, and enhance the efficiency of healthcare delivery. With features such as patient registration, doctor dashboards, real-time notifications, and data encryption, it ensures a smooth and secure healthcare experience. Accessible anytime from any internet-connected device, the system supports both small clinics and large hospitals, offering scalability

# Overall Description

## Product Perspective

This system is a web-based and mobile-responsive platform allowing interaction among patients, doctors, and administrators. It includes:

* Secure login for all users
* Doctor dashboard for patient access, prescriptions, and appointments
* Patient dashboard for bookings, medical history, and doctor communication
* Admin dashboard for system monitoring and maintenance.

### ****3.2 Product Functions****

* Patient registration/login
* Book, reschedule, or cancel appointments
* Maintain EMR including symptoms, reports, and prescriptions
* Doctor-patient chat and consultation window
* Prescription upload and pharmacy integration
* Admin control for managing hospital/clinic workflows

### ****3.3 User Characteristics****

* **Patient**: Basic computer/mobile knowledge
* **Doctor**: Medical professionals with system access to patient records
* **Admin**: IT personnel managing the system

### ****3.4 Hardware Constraints****

* Internet-connected devices (smartphone, PC, tablet)
* Modern browser support
* Minimum 2 GB RAM for client devices
* Server: Quad-core CPU, 8 GB RAM, 200 GB SSD storage

### ****3.5 Assumptions & Dependencies****

* Internet availability for accessing the system
* Patients must register for services
* Doctors are verified before access
* Secure server and backup system are maintained

## ****4. Specific Requirements****

This section outlines all functional and non-functional requirements necessary for the successful development and operation of the Online Healthcare Management System. These include interface definitions, core functionalities, performance expectations, and design constraints.

### ****4.1 External Interface Requirements****

#### ****4.1.1 User Interface****

* The system will provide a responsive and intuitive Graphical User Interface (GUI) for all users—patients, doctors, and administrators.
* Interfaces will be optimized for both desktop and mobile devices, ensuring accessibility via modern browsers (Chrome, Firefox, Safari, Edge).
* Different dashboards will be provided based on user roles:
  + **Patient Dashboard**: View medical history, book appointments, access prescriptions, chat with doctors.
  + **Doctor Dashboard**: View patient list, manage appointments, update EMRs, issue prescriptions.
  + **Admin Dashboard**: Manage users, system settings, generate reports, and perform system backups.

#### ****4.1.2 Hardware Interface****

* **Server Requirements**:
  + Minimum 4-core processor, 8 GB RAM, 200 GB SSD storage
  + Reliable internet connectivity for uptime maintenance
* **Client Devices**:
  + Smartphones, tablets, laptops, or desktop systems
  + Minimum 2 GB RAM and screen resolution of 1024×768

#### ****4.1.3 Software Interface****

* **Frontend Technologies**: HTML5, CSS3, JavaScript (React.js or Angular)
* **Backend Technologies**: Node.js, Python Flask, or Java Spring Boot
* **Database**: MySQL / PostgreSQL / MongoDB for structured patient and doctor data
* **Authentication**: JWT (JSON Web Token)-based login system
* **APIs**: RESTful APIs for frontend-backend communication
* **Operating System Compatibility**: Cross-platform web compatibility

#### ****4.1.4 Communication Interface****

* HTTPS protocol for all data transmission to ensure secure communication
* Email and SMS notifications through third-party services (e.g., Twilio, SendGrid)
* WebSocket or Firebase for real-time chat and appointment updates

### ****4.2 Functional Requirements****

#### ****4.2.1 Patient Functions****

* **Registration/Login**: Patients can create an account using email, phone, or social login. Secure login via OTP or password.
* **Profile Management**: Patients can update personal information, contact details, and medical history.
* **Appointment Booking**: Real-time availability checking and scheduling of appointments with doctors.
* **Medical Records**: Access to past diagnoses, prescriptions, lab reports, and doctor notes.
* **Online Consultation**: Chat-based or video consultation module for remote healthcare.
* **Prescription Access**: View and download prescriptions issued by doctors.
* **Notifications**: SMS/Email reminders for upcoming appointments and new prescriptions.

#### ****4.2.2 Doctor Functions****

* **Login & Dashboard Access**: Role-based dashboard with patient list, calendar view, and alerts.
* **Patient EMR Access**: Full access to the patient’s medical history and visit records.
* **Add Diagnosis & Prescriptions**: Enter symptoms, diagnosis, treatment plan, and prescribe medicines.
* **Appointment Management**: Confirm, reschedule, or cancel patient appointments.
* **Consultation Tools**: Integrated secure chat and (optional) video consultation module.

#### ****4.2.3 Administrator Functions****

* **User Management**: Add/remove doctors and patients, reset passwords, and assign roles.
* **System Configuration**: Manage hospital/clinic details, department settings, and consultation fees.
* **Report Generation**: Generate reports on patient visits, consultation statistics, and revenue.
* **Data Backup**: Trigger scheduled or manual backups of the system database.
* **Audit Logs**: Track login attempts, data access events, and system changes for security compliance.

### ****4.3 Performance Requirements****

* The system should load patient data in under 3 seconds under normal network conditions.
* Chat messages and real-time updates (appointments, prescriptions) should reflect within 500 milliseconds.
* System should support up to 1000 concurrent users without degradation.
* Uptime of 99.5% must be maintained with real-time health monitoring of services.
* The database should handle at least 10,000 patient records efficiently.

### ****4.4 Design Constraints****

#### ****4.4.1 Accuracy****

* Medical records must be stored and retrieved with 100% accuracy.
* Appointment time slots must reflect real-time availability to avoid double bookings.

#### ****4.4.2 Future Enhancements****

* Integration with wearable health devices for real-time vitals tracking.
* Multi-language support for regional accessibility.
* Integration with pharmacies and diagnostic labs for prescription refill and test scheduling.

#### ****4.4.3 User-Friendly Interface****

* Clean layout with minimal clicks to access key features.
* Help tooltips, FAQs, and onboarding for first-time users.

#### ****4.4.4 User Satisfaction****

* Designed to reduce patient wait time and streamline medical processes.
* Ensures availability of accurate medical information at all times.

#### ****4.4.5 Safety & Security****

* SSL encryption for all data transfers.
* Role-based access control to prevent unauthorized data access.
* Session timeout and automatic logout for idle users.
* Regular penetration testing and vulnerability assessments.

### ****4.5 Software System Attributes****

#### ****4.5.1 Security****

* All sensitive data is encrypted at rest and in transit.
* Multi-factor authentication for doctor and admin accounts.
* Logs of all user actions for traceability.

#### ****4.5.2 Reliability****

* Failover system in place for high availability.
* Daily backups and integrity checks for recovery readiness.

#### ****4.5.3 Maintainability****

* Modular design allows quick patching and feature upgrades.
* Admin panel includes system diagnostic and health tools.

#### ****4.5.4 Flexibility****

* Easily customizable to include additional departments or functionalities.
* API-first architecture allows integration with third-party systems.

#### ****4.5.5 Availability****

* Cloud-based deployment ensures global access 24/7.
* CDN and load balancer support for fast and distributed delivery.

|  |  |
| --- | --- |
| **Purpose** | Enable secure login and logout for all system users (Students, Institutions, Administrators). |
| **Input** | Username (email or ID) and Password. |
| **Process** | 1. Verify user credentials from the database. 2. Authenticate user role (Patient, Doctor, Admin). 3. Grant access to corresponding dashboard. |
| **Output** | Authenticated access to the platform. |

|  |  |
| --- | --- |
| **Purpose** | Allow suser to upload credentials, view report status, and share documents. |
| **Input** | Body documents (PDFs), personal identification, email ID, and credential details. |
| **Process** | 1. Upload documents securely. 2. Generate Health Report 3. Allow document sharing via unique link or QR code. |
| **Output** | It shows the Disease Report or health Report |

| **Purpose** | **Enable doctors to access patient records, manage appointments, provide diagnoses, and issue prescriptions.** |
| --- | --- |
| **Input** | Patient medical history, symptoms, uploaded reports, appointment requests, doctor login credentials. |
| **Process** | 1. Authenticate doctor login and provide dashboard access.  2. View scheduled appointments and patient history.  3. Record diagnosis and treatment.  4. Generate and upload prescriptions or recommendations. |
| **Output** | Updated patient medical records, issued prescriptions, and appointment status updates accessible to patients. |

# 5.Appendix

## Use-Cases

**Admin/Doctors/Patients**

Login and authentication

# Patients

Login and

authentication

ViewHealth Record

**Doctors**

Take Appointment

Hospital\_Id

Check report and give Prescription