Software Requirements Specification (SRS)

For Clinic Appointment Management System
Based on IEEE 830-1998 Standard

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) provides a comprehensive description of the functionalities, features, and constraints of the Clinic Appointment Management System. The intended audience includes software developers, testers, project managers, clinical administrators, and any other stakeholders involved in the development and deployment of this system.

1.2 Scope

The Clinic Appointment Management System is designed for small to mid-sized clinics in semi-urban and rural regions that face challenges in managing patient appointments due to lack of digital infrastructure. The system will streamline appointment booking, reduce patient waiting times, and provide a better planning structure for doctors and staff. It will include patient and doctor management, appointment booking, reminders, reporting, and support offline functionality.

1.3 Definitions, Acronyms, and Abbreviations

- SRS: Software Requirements Specification

- GUI: Graphical User Interface

- SMS: Short Message Service

- API: Application Programming Interface

- DBMS: Database Management System

- SQLite/MySQL: Lightweight and popular DBMS options

1.4 References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

- Field studies and interviews with staff of small clinics in Raipur, India

1.5 Overview

This document outlines the general and specific requirements of the software, including system behavior, user interface details, functional and non-functional requirements, and design constraints.

2. Overall Description

2.1 Product Perspective

This software is a stand-alone system designed to operate independently on low-end systems. It will be installed locally at clinics and optionally connected to internet services for SMS-based alerts.

2.2 Product Functions

- Patient registration and information management

- Doctor schedule management

- Appointment booking, updating, and cancellation

- Appointment reminders through SMS or messaging platforms

- Tracking missed appointments and cancellations

- Generating reports and visual dashboards

- Language toggle support (e.g., Hindi/English)

2.3 User Characteristics

- Reception Staff: Basic computer skills

- Doctors: Require schedule clarity and patient history

- Patients: General population, possibly unfamiliar with digital tools

2.4 Constraints

- Must work on low-spec devices (e.g., 2GB RAM systems)

- Offline functionality is essential

- Must have a lightweight and simple interface

- Data must be stored securely and backed up regularly

2.5 Assumptions and Dependencies

- Clinic has at least one functioning computer

- SMS gateway services are available and affordable

- Some initial training will be provided to staff

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface

- Forms for patient data entry

- Calendar-based interface for booking

- Tabular views for reports

- Button-based navigation with language switch

3.1.2 Hardware Interface

- Standard desktop PC or laptop

- Optional: printer for report generation

3.1.3 Software Interface

- Frontend: HTML/CSS/JavaScript or desktop GUI framework

- Backend: SQLite or MySQL

- External APIs: SMS Gateway API for alerts

3.1.4 Communication Interface

- System should be functional offline

- Reminders and alerts will require internet access

3.2 Functional Requirements

3.2.1 Patient Management

- Add/edit/delete patient information

- Store medical history and previous visits

3.2.2 Appointment Scheduling

- Book new appointments with details: patient, doctor, time

- View/update/cancel existing appointments

- Block time slots for breaks or holidays

3.2.3 Doctor Schedule Management

- Define daily/weekly availability

- View doctor-wise appointment lists

3.2.4 Reminder System

- Send appointment reminders via SMS

- Notify staff about cancellations or delays

3.2.5 Feedback and Complaint Tracking

- Record patient feedback and complaints

- Analyze feedback trends

3.2.6 Reporting and Analytics

- Generate reports (daily/weekly/monthly)

- Export to PDF or Excel

- Visualize peak hours, missed appointments, and trends

3.3 Performance Requirements

- System should respond within 2 seconds for any user action

- Should support up to 100 patient records and 5 concurrent users

3.4 Design Constraints

- Simple and minimalistic interface

- Cross-platform compatibility desirable (optional)

- Use of open-source libraries and technologies

3.5 Software System Attributes

3.5.1 Reliability

- The system should operate error-free for at least 8 hours of daily clinic operations

3.5.2 Availability

- Should function offline and sync data when online

3.5.3 Security

- Password protection for admin-level access

- Encrypted patient data storage (where feasible)

3.5.4 Maintainability

- Modular codebase with documentation

- Ability to add/remove modules (e.g., reporting, feedback)

3.5.5 Portability

- Must run on Windows 7 and above

- Optional future support for Android

4. Appendices

- Sample mockups of UI forms (to be attached separately)

- Glossary of Hindi-English terms used in the interface

- SMS reminder template examples