

SOFTWARE REQUIREMENTS SPECIFICATION

for

CLINICAL MANAGEMENT SYSTEM

Prepared by:

Anirudh M K (20)

Hiba Nm (33)

Aashil (6)

Abhinav (4)

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Intended Audience	2
1.3	Scope	2
1.4	Definitions And Acronyms	3
2	Overall Description	3
2.1	User Needs	3
2.2	Assumptions And Dependencies	4
3	System Features And Requirements	4
3.1	Functional Requirements	4
3.2	External Interface Requirements	5
3.3	System Features	5
3.4	Nonfunctional Requirements	6

1 Introduction

1.1 Purpose

The "Clinical Management System" (CMS) isn't just another piece of software—it's a game-changer for healthcare. Imagine a bustling clinic where patients dread long waits, doctors juggle paper records, and administrators drown in scheduling chaos. CMS steps in to save the day, offering a web-based powerhouse that streamlines appointment scheduling, patient record management, lab test processing, medicine dispensing, billing, and even website administration. Our goal? To slash wait times (studies show patients wait an average of 20 minutes past their appointment time!), empower doctors with instant data access, and give administrators a bird's-eye view of operations. This SRS lays out the blueprint—features, interfaces, constraints, and all—serving as the ultimate guide for developers and a pitch to healthcare authorities for approval.

1.2 Intended Audience

This document is crafted for a diverse crew, each with a stake in CMS's success:

- **Clinic Administrators:** The masterminds who'll use CMS to orchestrate resources—think of them as the conductors of a healthcare symphony, ensuring every instrument (staff, equipment, meds) plays in tune.
- **Doctors and Medical Staff:** The frontline heroes who'll tap CMS to track patient histories, schedule visits, and order tests, cutting through red tape to focus on healing.
- **Patients:** The heart of it all—everyday folks who'll book appointments with a click, check lab results from their couch, and settle bills without a hassle.
- **Developers and Testers:** The tech wizards who'll bring CMS to life, using this SRS as their treasure map to code and debug.
- **Stakeholders:** The big-picture thinkers—managers, investors, even insurers—who'll see CMS as a profitability booster and a patient satisfaction rocket.

1.3 Scope

The CMS is a Swiss Army knife for clinics, tackling everything from appointment chaos to medicine mix-ups. Patients will love booking a 10 AM slot with Dr. Smith from their phone, while doctors can pull up Mrs. Jones's allergy history in seconds. Administrators will track dwindling penicillin stocks and update the clinic's website with a new "Flu Shot Season" banner—all from one dashboard. Powered by SQLite databases ('users.db' for logins, 'clinic.db' for everything else), CMS ensures data flows smoothly, bills get paid fast, and HIPAA keeps it all under lock and key. Future dreams? Adding AI to predict appointment no-shows or a mobile app for on-the-go access—all while staying within budget and regulatory lines.

1.4 Definitions And Acronyms

Let's decode the CMS lingo:

- **CMS:** Clinical Management System—the star of this show, a web-based titan of healthcare efficiency.
- **Appointment:** A sacred pact between patient and doctor—say, a 15-minute slot to check that nagging cough.
- **Patient Records:** The digital dossier of a patient's life—past flus, current meds, that broken arm from '09.
- **Lab Tests:** Bloodwork, X-rays, you name it—ordered, tracked, and delivered via CMS.
- **Medicines:** The pharmacy's lifeblood—CMS knows if there's enough ibuprofen for the week.
- **Website Management:** The clinic's online face—think updating "Meet Our Staff" or tracking visitor stats.
- **SQLite Databases:** The unsung heroes—'users.db' guards logins, 'clinic.db' holds the clinic's pulse.
- **HIPAA:** The law that says, "Patient data stays private, or else!"

2 Overall Description

2.1 User Needs

CMS is built for real people with real problems:

- **Patients:** They're tired of phone tag for appointments—CMS lets them book online, see if their cholesterol test is back, and pay without digging through wallets.
- **Doctors:** They need speed—imagine diagnosing a kid's ear infection with last year's records at their fingertips, not buried in a filing cabinet.
- **Administrators:** They crave control—knowing Dr. Patel's booked solid, the lab's swamped, or aspirin's running low, all in one glance.
- **All Users:** They demand security (no data breaches!), accuracy (no prescribing penicillin to the allergic!), and a system that doesn't require a PhD to use.

2.2 Assumptions And Dependencies

Here's what we're betting on and leaning on:

- **Assumptions:**
 - Patients and staff have internet—CMS isn't mailing floppy disks!
 - Data's entered right—garbage in, garbage out, right?
 - HIPAA's our North Star—privacy isn't optional.
- **Dependencies:**
 - SQLite databases must be rock-solid—'users.db' and 'clinic.db' are the backbone.
 - Clinic systems (like lab machines) need to handshake with CMS—no lone wolves here.
 - Funding and staff buy-in—without cash and cooperation, we're sunk.

3 System Features And Requirements

3.1 Functional Requirements

Here's the meat of CMS—its superpowers:

- **User Registration and Profile Management:**
 - Newbies (patients, doctors, admins) sign up with a name, email, and phone—verified by a quick text code.
 - Profiles update anytime—Dr. Lee adds her new cardiology certification; Mr. Patel swaps his old number.
- **Appointment Scheduling:**
 - Patients pick a slot—say, Tuesday at 3 PM with Dr. Kim—and get a confirmation email.
 - Doctors see their day's lineup; admins reshuffle if Dr. Kim's out sick.
- **Lab Test Ordering:**
 - Patient requests a blood sugar test; CMS pings the lab.
 - Results land in the system—Dr. Chen reviews them, flags high glucose, and plans next steps.
- **Record Management:**
 - Doctors log every visit—Mrs. Gupta's migraine meds, little Tim's flu shot.
 - Admins pull reports—like how many patients saw Dr. Rao last month.

- **Medicine Dispensing:**
 - Admins track stock—50 Advil left, 20 ordered yesterday.
 - Patient gets meds post-visit; CMS deducts from inventory and flags reorder at 10 units.
- **Billing and Payment:**
 - CMS tallies a visit—\$50 consult, \$30 test—and sends the bill.
 - Patient pays via card or insurance; CMS updates their account, no paper chase needed.
- **Website Administration:**
 - Admins post "Closed for July 4th" or tweak the homepage banner.
 - Analytics show 200 visitors checked "Our Services" last week—time for a refresh?

3.2 External Interface Requirements

CMS doesn't live in a bubble:

- **User Interface:** A clean web dashboard—patients see "Next Appointment: May 10," doctors get a patient list, admins tweak settings.
- **Clinic Systems Integration:** Syncs with the lab's test machines and the old patient database—no double-entry nightmares.
- **Lab Integration:** Sends test orders and grabs results—think of it as a digital lab assistant.

3.3 System Features

The gears that make CMS hum:

- **User Authentication:** Passwords and roles—patients can't accidentally edit Dr. Singh's schedule!
- **Record Management Module:** A searchable archive—find "John Doe, 2023 flu" in seconds.
- **Scheduling Engine:** Smart enough to avoid double-booking Dr. Patel at 9 AM.
- **Database Management:** 'users.db' locks down logins; 'clinic.db' tracks every pill and visit.
- **Transaction Processing:** Bills and payments flow like clockwork—\$80 in, receipt out.

3.4 Nonfunctional Requirements

CMS isn't just functional—it's fabulous:

- **Performance:** Handles 100 patients booking at once—peak flu season, no sweat, under 2-second responses.
- **Security:** Fort Knox for data—encryption, HIPAA rules, no leaks even if hackers knock.
- **Scalability:** Grows from a 5-doctor clinic to a 50-doctor empire without a hiccup.
- **Reliability:** 99.9% uptime—down for 8 minutes a year, max, because patients don't wait.
- **Usability:** So simple, a sleep-deprived intern nails it first try—no manual needed.