

Software Requirements Specification (SRS)

Project Title:

Medcor Unified Healthcare Platform

(Including Patient Portal, Doctor Portal, Clinic Admin, and AI ChatBot Widget)

1. Introduction

1.1 Purpose

The purpose of this SRS is to define the software requirements for the **Medcor platform**, which includes four major modules:

- Patient Portal
- Doctor Portal
- Clinic Admin Portal
- Medcor AI ChatBot Widget

This document ensures alignment between stakeholders, designers, developers, and testers by outlining clear functional and non-functional requirements.

1.2 Scope

Medcor is a cloud-based SaaS platform that enables hospitals and clinics to manage doctor-patient workflows efficiently while integrating conversational AI for patient engagement. It allows:

- Clinics to register and create subdomains
- Doctors to manage appointments and profiles

- Patients to book appointments and interact with an AI assistant
- Web widget (AI ChatBot) for real-time assistance, bookings, and visual tools

1.3 Intended Audience

- Product Owners
- Engineering Team
- QA Team
- UX/UI Designers
- DevOps and Security Teams
- End Clients (Hospitals, Clinics)

1.4 Definitions

- **AI ChatBot:** An embeddable frontend widget with voice/text interactions.
 - **HeyGen:** Third-party tool for avatar/video generation.
 - **Face Editor:** Real-time camera-based image editor.
 - **Subdomain:** Custom portal URL created per clinic (e.g., `clinicname.medcor.ai`)
-

2. Overall Description

2.1 Product Perspective

Medcor is a modular, web-based platform that uses a central API service, with frontend applications for patients, doctors, and admins. The AI ChatBot functions as an embeddable widget powered by JavaScript.

2.2 User Classes & Characteristics

User Role	Description
Patient	Books appointments, chats with AI assistant
Doctor	Manages appointments, views patient data
Clinic Admin	Registers clinic, manages doctors and treatments
Guest	Anonymous user with limited access via widget

2.3 Assumptions and Dependencies

- Internet access is required.
 - Browser must support JavaScript and HTML5.
 - External services like HeyGen may be integrated for AI avatar features.
-

3. Functional Requirements

3.1 Clinic Admin Portal

- FR-1.1: Admin can register a new clinic via <https://app.medcor.ai/>
- FR-1.2: Upon registration, the system generates a unique subdomain
- FR-1.3: Admin can add/edit doctors, patients, treatments
- FR-1.4: Admin can generate and copy a chatbot integration script
- FR-1.5: Admin can monitor appointments and AI interactions

3.2 Doctor Portal

- FR-2.1: Doctor logs in to view patient bookings
- FR-2.2: Doctor can add/edit their profile and specialties

- FR-2.3: Doctor can view treatment history per patient

3.3 Patient Portal

- FR-3.1: Patients can register, log in, and reset password
- FR-3.2: Patients can browse doctors and treatments
- FR-3.3: Patients can book or cancel appointments
- FR-3.4: Patients can view appointment history and status

3.4 AI ChatBot Widget

- FR-4.1: Chatbot auto-loads on any integrated clinic webpage
 - FR-4.2: Users can interact with the chatbot via voice or text
 - FR-4.3: Users can browse doctor portfolios within the widget
 - FR-4.4: Authenticated users can book appointments via chatbot
 - FR-4.5: Users can launch the real-time face editor module
 - FR-4.6: Session expiration and mute/unmute features are available
-

4. Non-Functional Requirements

4.1 Performance

- NFR-1.1: Chatbot must load within 2 seconds of page load
- NFR-1.2: Appointment booking confirmation must be under 3 seconds

4.2 Scalability

- NFR-2.1: The system must support at least 1000 concurrent chatbot users

4.3 Security

- NFR-3.1: All login endpoints must be HTTPS and follow encryption standards
- NFR-3.2: Chatbot voice/camera access requires user consent
- NFR-3.3: Sensitive data (e.g., health records) must follow HIPAA/GDPR compliance (if applicable)

4.4 Usability

- NFR-4.1: UI must follow accessibility standards (WCAG 2.1)
- NFR-4.2: Chatbot interface must be mobile responsive

4.5 Maintainability

- NFR-5.1: Widget updates should require no user-side manual steps

5. System Interface Requirements

5.1 External Interfaces

Type	Interface	Purpose
Web App	https://app.medcor.ai/	Portal login and admin tasks
Widget	<script> Integration Tag	ChatBot frontend integration
API	REST APIs	Auth, booking, profile data
Third-Party	HeyGen API	AI avatar video generation

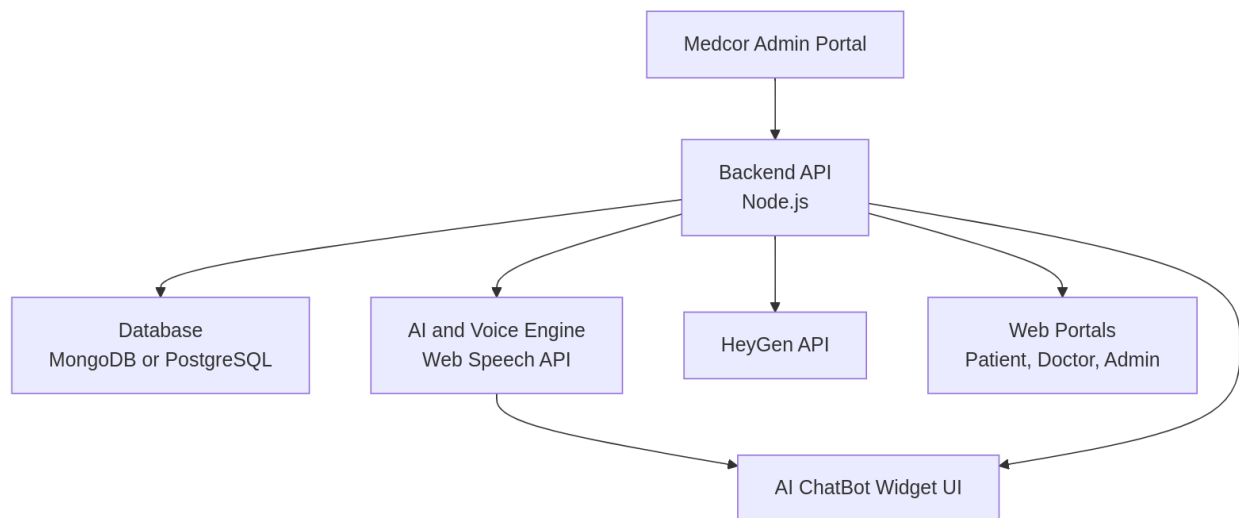
5.2 Hardware Interfaces

- No dedicated hardware required, but user-facing features (camera, mic) must be supported by the user's device.

5.3 Software Interfaces

- Frontend: React.js + Webpack (widget)
- Backend: Node.js, Express
- DB: MongoDB/PostgreSQL
- Hosting: AWS or equivalent
- Voice AI: Web Speech API / HeyGen

6. System Architecture Overview



7. Constraints

- Widget must be deployable on third-party sites via script tag
- Deployment must not require page redesign
- Widget must not interfere with existing JavaScript on clinic websites

8. Acceptance Criteria

- Clinics can register and activate a chatbot without technical support
- Widget loads and works in Chrome, Safari, Firefox, and Edge
- Patients can complete an appointment booking within 5 steps
- Face Editor and AI Voice Assistant function across mobile and desktop
- Data transmission is secure and permissions are properly managed

9. Appendix

- **Sample Widget Script**

```
<script id="customer_medcorhospital"
src="https://app.medcor.ai/static/chat_widget/dist/chatWidget.min
.js?uuid=XYZ&schema=medcorhospital" defer></script>
```

- **Permissions Prompt Example**
Screenshots showing browser permissions for camera/mic.
- **Widget UI Layout**
Chat bubble, header, close button, and dynamic content area.

Deliverables

- Full-featured SRS for engineering and QA
- Aligned with PRD and user manuals
- Ready for handoff to Dev and QA teams