Software Requirements Specification

for

Golden Medical Hospital

Prepared by:

MD. Shihaful Islam Ornob (2254901012)
Mahadi Hasan Shakib (2254901020)
MD. Shahrukh Hossain (2254901054)
MD. Sakib Hasan (2254901060)
Shameha Fatema Shemonty (2254901092)

Bangladesh University of Professionals, Dhaka
Information and Communication Technology

October 06, 2025

Contents

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to formally define the requirements for the Golden Medical Hospital, a modern web-based system designed to streamline healthcare interactions. The portal aims to empower patients by providing an efficient and intuitive platform to search for doctors, schedule appointments, access prescriptions, and manage health records. For doctors, it provides a centralized system to manage appointments, respond to patient requests, and interact seamlessly with patients. Administrators will maintain oversight by verifying doctor credentials, monitoring feedback, and generating actionable reports.

This document ensures that all stakeholders including developers, testers, project managers, and clients share a unified understanding of the system's scope, functionality, and operational constraints.

1.2 Intended Audience

The Golden Medical Hospital SRS is intended for:

- **Developers:** To understand the requirements, system architecture, and technical details for implementation.
- **Testers:** To design and execute test cases based on functional and non-functional requirements.
- **Project Supervisors and Instructors:** To evaluate the completeness and quality of the project.
- **End Users:** To gain a high-level understanding of the platform's capabilities.

Readers who are primarily interested in the overall goals and features may focus on Section 3, while developers and testers should pay closer attention to Sections 3 and 4 for detailed requirements.

1.3 Intended Use

The Golden Medical Hospital serves as a comprehensive online healthcare management tool. Its core functionalities include:

- **For Patients:** Effortlessly search for doctors by category, book appointments, and manage personal health records.
- **For Doctors:** Organize schedules, respond to patient requests, and deliver healthcare services efficiently.

- **For Administrators:** Oversee doctor verification, manage patient feedback, and maintain system integrity.

The portal is designed to bridge the gap between patients and healthcare providers, offering a secure, user-friendly, and accessible platform that fosters timely medical interactions.

1.4 Project Scope

The Golden Medical Hospital Portal is a self-contained, web-based application developed using the MERN stack. It is designed to streamline basic hospital management functions for patients, doctors, and administrators.

Key objectives include:

- Allowing patients to search for doctors by specialization, book appointments, and view prescriptions and reports.
- Enabling doctors to manage appointments, track patient interactions, and update schedules efficiently.
- Providing administrators with tools to verify doctor credentials, manage system content, monitor feedback, and generate reports.

1.5 Risk Definition

The development and deployment of the Golden Medical Hospital Portal involve several important risks that must be carefully managed:

- **Data Security:** The system stores sensitive health information, including patient reports and prescriptions. Ensuring encryption, secure login, and controlled access is critical to prevent unauthorized access or data breaches.
- **System Availability:** Even brief system downtime can disrupt appointment management or delay access to medical reports. Reliable hosting and regular monitoring are necessary to minimize downtime.
- **User Errors:** Patients or doctors may enter incorrect information during registration, appointment booking, or report uploads. Proper validation and administrative oversight are essential to maintain data accuracy.
- **Scalability and Performance:** The system should handle expected numbers of concurrent users (patients, doctors, and administrators) without noticeable slowdowns. Poor performance could reduce user satisfaction.
- **Regulatory Compliance:** The system must comply with applicable data protection and healthcare regulations to avoid legal and financial consequences. This includes secure storage of medical records and proper handling of personal information.

Chapter 2

Overall Description

2.1 User Classes and Characteristics

The system is designed to accommodate three primary user groups:

- **Patients:** Individuals seeking healthcare services. They require a simple, intuitive interface to search for doctors, schedule appointments, and manage personal health information. Basic digital literacy is assumed.
- **Doctors:** Licensed healthcare professionals managing their schedules and interacting with patients. They require tools to accept or decline appointments and monitor their commitments efficiently.
- **Administrators:** Users with elevated privileges responsible for system management. They verify doctor credentials, manage accounts, monitor feedback, and generate comprehensive analytical reports.

2.2 User Needs

Each user group has distinct needs:

- **Patients:** Reliable, secure, and convenient access to healthcare services without visiting clinics for routine tasks.
- **Doctors:** Organized tools to manage patient requests and appointments, minimizing administrative overhead.
- **Administrators:** Oversight capabilities to ensure fairness, maintain system integrity, and provide a seamless user experience.

Addressing these needs creates a balanced ecosystem, promoting faster, transparent, and accessible healthcare delivery.

2.3 Operating Environment

The Doctor-Patient Portal will operate on:

- **Browsers:** Google Chrome, Mozilla Firefox, Safari, Microsoft Edge.

- **Operating Systems:** Windows, macOS, Linux, and Android.
- **Devices:** Desktop computers, laptops, tablets, and smartphones.
- **Communication:** HTTP/HTTPS protocols, with HTTPS mandatory for secure data transmission.

The backend will leverage a reliable database hosted on a robust server infrastructure.

2.4 Constraints

Key constraints include:

- Frontend implementation must use React+Next.js.
- The system will initially support only the English language (unless localization is planned).
- Compliance with data privacy and healthcare regulations is mandatory.
- Automated daily database backups must be maintained and easily restorable in case of system failure.
- The interface must remain intuitive for users with minimal technical expertise.

2.5 Assumptions

The project assumes:

- Users have stable internet access and compatible devices.
- Patients and doctors have at least basic digital literacy to operate the system.
- Doctors provide accurate information during registration, subject to administrative verification.
- The hospital management will ensure that using this portal aligns with local healthcare laws and privacy regulations.

Chapter 3

Requirements

3.1 Functional Requirements

3.1.1 User Login

As a registered user (Patient, Doctor, or Admin), I want to log into the system securely so that I can access my respective dashboard and services.

Success:

- Patients, Doctors, and Admins can log in using valid credentials.
- Upon successful login, the system redirects the user to their corresponding dashboard.
- Session tokens are generated to maintain secure access.

Failure:

- If credentials are invalid, an error message "Invalid username or password" is displayed.
- If the account is not verified (doctor) or deactivated (patient), the login attempt is denied.
- Unauthorized attempts are blocked, and users are redirected to the login page .

3.1.2 User Registration

As a new user (Patient or Doctor), I want to register on the system so that I can access healthcare services or provide medical services.

Success:

- Patients can register with personal details such as name, age, gender, email, and password.
- Doctors can register with their credentials, specialization, and qualifications.
- Doctors' accounts remain pending until verified by the admin.
- A confirmation notification is sent after successful registration

Failure:

- If required fields are missing, the system alerts the user to complete them.
- If the email is already registered, the system prevents duplicate accounts.
- If server issues occur, an error message is displayed, and registration does not proceed.

3.1.3 Search Doctor

As a patient, I want to search for doctors based on specialization, category, or filters so that I can find the most suitable doctor.

Success:

- Patients can search doctors by category, specialization, or name.
- Filters allow narrowing results based on location, availability, or ratings.
- Search results are displayed clearly with doctor details.

Failure:

- If no doctor matches the search, a message “No doctors found” is displayed.
- If a technical error occurs, the system shows a friendly error message.

3.1.4 Select Doctor and Book Appointment

As a patient, I want to select a doctor and book an appointment so that I can receive medical consultation.

Success:

- Patients can view a doctor’s profile and available slots.
- Patients can select a date and time, then confirm the appointment.
- A notification is sent to both patient and doctor after booking.

Failure:

- If the selected time slot is unavailable, an alert notifies the patient.
- If booking fails due to system error, a retry option is provided.
- Unauthorized patients cannot proceed without login.

3.1.5 View Prescription and Notifications

As a patient, I want to view my doctor’s prescriptions and receive notifications so that I can follow medical instructions properly.

Success:

- Patients can access prescriptions uploaded by doctors.
- Patients are notified when a new prescription or report is added.
- Patients can download prescriptions for offline use.

Failure:

- If the prescription is unavailable, a message is shown.
- If unauthorized access is attempted, the system redirects to login.

3.1.6 Doctor Dashboard Management

As a doctor, I want to manage patient requests and appointments so that I can organize my schedule efficiently.

Success:

- Doctors can view pending patient appointment requests.
- Doctors can accept/reject appointment requests.
- Upcoming appointments are listed in an organized manner.

Failure:

- If a request update fails, an error message is displayed.
- If the doctor account is not verified, dashboard access is restricted.

3.1.7 Admin Dashboard Management

As an admin, I want to verify doctors, manage platform content, and oversee feedback so that the platform remains trustworthy.

Success:

- Admins can approve or reject doctor registrations.
- Admins can delete or deactivate suspicious accounts.
- Patient feedback is reviewed, and valid feedback is published.
- Admins can view system-wide appointment summaries in charts/reports.

Failure:

- If verification data is missing, the system alerts the admin.
- If server errors occur, changes are not saved, and retry is prompted.

3.1.8 Profile Management

As a user (Patient or Doctor), I want to manage my profile so that my information remains accurate.

Success:

- Patients can update contact information and personal details.
- Doctors can update their specialization, availability, and credentials.
- Admins can modify user information when necessary.

Failure:

- If mandatory fields are left empty, the update fails.
- Unauthorized modifications are blocked with an alert.

3.1.9 Patient Dashboard

As a patient, I want to access my dashboard so that I can manage appointments and view history.

Success:

- Patients can view past appointments, upcoming appointments, and medical history.
- Patients receive suggestions for doctors based on their previous visits or specialty searches.
- Notifications are displayed for updates or cancellations.

Failure:

- If dashboard data fails to load, an error message is displayed.
- If the session expires, the patient is redirected to login.

3.1.10 Emergency Condition Handling

As a patient, I want to quickly notify doctors in emergencies so that I can receive priority assistance.

Success:

- Patients can trigger an emergency alert by pressing a red emergency button.
- Notifications are immediately sent to available doctors and admins.
- The system prioritizes emergency cases over normal appointments.

Failure:

- If the emergency button fails, a backup notification is triggered via system alerts.
- If no doctors are available, the system informs the patient and escalates to admin.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- **Response Time:** The system should respond to user actions (e.g., appointment booking, prescription viewing) within 2 seconds under normal operating conditions to ensure a smooth user experience.
- **Scalability:** The system must be scalable to support at least 1,000 concurrent users, with the ability to handle a 20% increase during peak usage without significant degradation in performance.

3.2.2 Safety Requirements

- **User Data Protection:** The system must implement strong measures to safeguard sensitive health information, ensuring confidentiality and integrity to prevent unauthorized access or data breaches.
- **Data Backup and Recovery:** In the event of a system failure, regular backups and recovery mechanisms must be in place to avoid loss of patient records, prescriptions, or appointments.

3.2.3 Security Requirements

- **User Authentication:** Users (Patient, Doctor, Admin) must undergo secure authentication processes, with role-based access control to prevent unauthorized actions.
- **Data Encryption:** All sensitive information, including personal details, prescriptions, and appointment records, must be encrypted during both transmission and storage.
- **Audit Logging:** All critical actions (e.g., doctor approval, appointment acceptance, profile updates) must be logged for security monitoring and accountability.

3.2.4 Software Quality Attributes

- **Usability:** The system should provide an intuitive, user-friendly interface that requires minimal training, with at least 90% satisfaction in user feedback.
- **Reliability:** The system should ensure at least 99.9% uptime, making healthcare services accessible at all times.
- **Maintainability:** The system should be modular and easy to maintain, allowing developers to fix issues or add new features with minimal disruption.
- **Portability:** The system should be compatible with modern browsers and mobile devices, ensuring accessibility across platforms.

3.2.5 Business Rules

- **Doctor Verification:** Only doctors approved by the Admin after proper verification can provide services through the portal.
- **Patient Registration:** Only registered patients can book appointments, view prescriptions, or access the dashboard.
- **Appointment Policy:** Appointments can only be booked during available time slots, and cancellation policies must be enforced by the system.
- **Emergency Handling:** Emergency requests must always be prioritized, overriding regular appointment scheduling when necessary.

Appendices

Appendix A

Glossary

- **SRS:** Software Requirements Specification — A formal document that defines the functional and non-functional requirements of the system.
- **TBD:** To Be Determined — Placeholder used for requirements, design decisions, or details that have not yet been finalized.
- **Portal:** The Doctor-Patient web-based system described in this document.
- **User Classes:** Distinct categories of system users: Patients, Doctors, and Administrators, each with specific roles and privileges.