

Software Requirement Specification

BookMyDoc

Document:

System Requirement Specification Document

Title:

System Requirement Specification for Online Doctor Appointment Booking Application

Team:

Mohan Malappa Talavar (233169)

Pranay Jagdish Patil (233185)

Objective (Purpose):

The objective of the Online Doctor Appointment Booking Web Application is to offer users a convenient, efficient, and user-friendly solution for scheduling medical appointments. Our platform presents a comprehensive list of doctors, including their schedules and availability in specific hospitals. This empowers users to make well-informed decisions and seamlessly book appointments, eliminating the need for unnecessary visits. By minimizing such visits, our system enhances hospital efficiency, reduces wait times, and ensures a smoother experience for both doctors and patients.

Scope:

This System is limited to offering a user-friendly solution for scheduling medical appointments and services exclusively within a single, chosen hospital. The application will empower users with the ability to access doctor information, schedules, and availability within the confines of this hospital, ensuring a seamless and informed booking experience. While presently focused on a specific hospital, the project's potential expansion to encompass multiple healthcare facilities remains a possibility for future iterations.

Functional Requirements:

The Doctor's Appointment System is composed of three distinct modules, each designed to serve a specific role within the application:

Admin Module:

- Admin can log in to the system.
- Admin can register a newly recruited doctor.
- Admin can view a comprehensive list of all registered doctors and patients.
- Admin possesses the authority to view detailed information about doctors and patients, as well as the capability to remove them.
- Admin can access and review the complete schedule of appointment bookings for all doctors.
- Admin is empowered to approve or reject patient appointment cancellation requests.

Doctor Module

- Doctors can log in to their accounts.
- Doctors have the ability to add or modify their availability schedules.
- Doctors can access details of revisiting patients to enhance personalized care.
- Doctors can view their own appointment schedules.

Patient Module

- Patients can register within the system and subsequently book appointments.
- Registered patients can securely log in to their accounts.
- Patients can initiate the booking and cancellation of appointments.
- Patients can access their prescribed treatment plans provided by the doctors.
- Patients can request password reset in case of forgotten credentials.

Additional Functionalities:

- User Registration and Login: Users can create accounts, log in securely, and access personalized profiles.
- Specialized Doctor Listing: Users can retrieve a list of doctors categorized by specialization.
- Forgot Password: Users can initiate the process of resetting their password in case it's forgotten.

Non Functional Requirements:

Performance:

- **Response Time:** The system should provide near-instantaneous response times for basic interactions such as viewing doctor lists and scheduling appointments.
- **Scalability:** The application should be able to handle increased user load without significant degradation in performance.

Security:

- **User Data Protection:** All sensitive user data, including personal and medical information, should be encrypted, and securely stored.
- **Authentication and Authorization:** User authentication and authorization should be implemented to ensure that only authorized individuals can access specific modules and functionalities.

Usability:

- **User-Friendly Interface:** The user interface should be intuitive and easy to navigate, catering to users of varying levels of technical expertise.
- **Accessibility:** The system should be designed to accommodate users with disabilities, adhering to accessibility standards.

Availability:

- **Uptime:** The system should aim for a high level of availability, minimizing downtime for maintenance or technical issues.
- **Backup and Recovery:** Regular backups of data should be performed, and mechanisms for quick data recovery in case of failures should be in place.

Compatibility:

- **Cross-Platform Compatibility:** The application should be compatible with various web browsers and operating systems to ensure a consistent user experience.
- **Mobile Responsiveness:** The system should be responsive and accessible across a range of mobile devices.

Reliability:

- **Data Integrity:** The system should ensure the accuracy and integrity of data, preventing data loss or corruption.
- **Error Handling:** Proper error handling mechanisms should be implemented to provide informative error messages and guide users in resolving issues.

Regulatory Compliance:

- **Privacy Regulations:** The system should comply with relevant privacy regulations and standards (such as GDPR or HIPAA) to safeguard patient data.

Performance Testing:

- Load Testing: The system should undergo load testing to determine its performance under various user load conditions.
- Stress Testing: Stress testing should be conducted to assess the application's stability under extreme conditions.

Maintenance:

- Maintenance Window: Scheduled maintenance should be communicated to users in advance, and the application should be brought down for maintenance during low-usage periods.
- Version Updates: The system should be designed with flexibility to accommodate future updates and enhancements.