SOFTWARE REQUIREMENTS SPECIFICATION

FOR

DOCTOR APPOINTMENT SYSTEM

BY

BHAVANI KUMAR KALAVAKUNTA (C314)

## Table of Contents:

## 1.Introduction 3

## 1.1 Purpose

## 1.2 Scope of the Project

## 2.Functional Requirements 4 - 5

## 2.1 Patient Registration

## 2.2 Doctor Registration

## 2.3 Appointment Scheduling

## 2.4 Appointment Management

2.5 Search

2.6 User Authentication

2.7 User Management

## 3.Non-Functional Requirements 6

## 3.1 performance

## 3.2 Security

## 3.3 Scalability

3.4 Reliability

## 4.Technical Requirements 6

## 5.Constraints 6

## 6. Assumptions and Dependencies 6-7

7.Glossary 7

**1. Introduction**

The purpose of this document is to specify the requirements for the Doctor Appointment system, which will allow patients to schedule appointments with doctors online. The system will be developed using Java8, Spring Boot, Hibernate, and MySQL.

* 1. **Purpose**

The purpose of the Doctor Appointment system is to provide patients with an online platform to schedule appointments with doctors in a user-friendly and efficient manner. The system aims to simplify the process of booking appointments, reduce wait times, and increase patient satisfaction.

* 1. **Scope of the project**

The scope of the Doctor Appointment system project is to develop an online platform that will allow patients to schedule appointments with doctors. The system will be developed using Java8, Spring Boot, Hibernate, and MySQL.

The system will allow patients to create an account, search for doctors based on their specialization, location, and availability, and schedule appointments with them. Patients will be able to view their upcoming appointments, cancel them, and receive SMS notifications as reminders.

Doctors will be able to create an account, manage their appointments, and mark them as complete. Doctors will also be able to view their appointment schedule and cancel appointments as needed.

The system will support user authentication to ensure the security of patient and doctor information. The system must be scalable, reliable, and able to handle a large number of concurrent users.

Overall, the Doctor Appointment system will provide a user-friendly and efficient way for patients to schedule appointments with doctors, and for doctors to manage their appointments.

**2. Functional Requirements**

The Doctor Appointment system must provide the following functionality:

**2.1 Patient Registration**

Patients must be able to create an account with the system. The registration process should include the following fields:

* First Name: Patient should provide Patient’s First Name while registering.
* Last Name: Patient should provide Patient’s Last Name while registering.
* Email Address: Patient should provide Patient’s Email Address while registering.
* Password: Patient should provide Patient’s Password while registering.
* Phone Number: Patient should provide Patient’s phone number while registering.

**2.2 Doctor Registration**

Doctors must be able to create an account with the system. The registration process should include the following fields:

* First Name: Doctor should register with his First Name.
* Last Name: Doctor should register with his Last Name.
* Email Address: Doctor should register with his Email address.
* Password: Doctor should register with his Password.
* Phone Number: Doctor should register with his Phone number.
* Specialization: Doctor should register with his Specialization.

**2.3 Appointment Scheduling**

Patients must be able to schedule appointments with doctors. The scheduling process should include the following steps:

* Select a Doctor
* The patient can select a doctor based on the specialization and the disease/illness they are facing.
* Choose an Available Time Slot
* Patient can have the ability to choose the availability of time slot based on the selection of the doctor.
* Confirm Appointment
* After selection and choosing the available time slot the patient can book his appointment with confirm appointment.

**2.4 Appointment Management**

Admin must be able to manage their appointments. The management process should include the following functionality:

* View Appointments
* Admin can have the privilege for this Feature.
* Admin can view the appointments that are booked for the doctor.
* Cancel Appointments
* Admin can cancel the appointments.
* While cancelling he/she can consider the scenarios like.

1. The selected doctor may be on leave /Emergency
2. The selected time slot is Expired
3. The selected doctor may be engaged in the other treatment in emergency.

* Mark Appointments as Complete
* Admin can mark the appointment as complete after the completion of the timeslot.

**2.5 Search**

Patients must be able to search for doctors by the following criteria:

* Specialization
* Patients can search the doctor’s specialization.
* Availability
* Patients can search the doctor’s availability to book the time slot.

**2.6 User Authentication**

The system must support user authentication to ensure the security of patient and doctor information.

* Patient Login
* Patient can login into the application with email and password which are given at the time of registration.
* Doctor Login
* Doctor can login into the application with email and password which are given at the time of registration.
* Admin Login
* Admin can login into the application with email and password which the hospital/Organization assigned or the registered.

**2.7 User Management**

Admin must be able to manage the users. The management process should include the following functionality.

* Remove Patients: Admin can remove Patients.
* Remove Doctors: Admin can remove Doctors.
* Add Doctors: Admin can ­­­­­­Add Doctors.
* Add/Update Specialization: Admin can Add/Update Doctors Specialization.
* Add Patients: Admin can add Patients.

**3. Non-Functional Requirements**

The Doctor Appointment system must meet the following non-functional requirements:

**3.1 Performance**

The system must be able to handle a large number of concurrent users and provide a responsive user experience.

**3.2 Security**

The system must be secure and protect patient and doctor information from unauthorized access.

**3.3 Scalability**

The system must be scalable and able to accommodate future growth.

**3.4 Reliability**

The system must be reliable and provide high availability to users.

**4. Technical Requirements**

The Doctor Appointment system must be developed using the following technologies:

• Java8

• Spring Boot

• Hibernate

• MySQL

**5. Constraints**

The Doctor Appointment system must adhere to the following constraints:

* The system must be developed with compatibility of following constraints
* Hardware
  + Processor: Pentium 2.4 GHz or above.
  + Memory: 256 MB RAM or above.
  + Hard disk: At least 3GB or above [At least 3 GB free space is required]
* Software
  + Operating System: windows 7/8/10/11 , Linux/Ubuntu/Centos e.t.c.
  + Web Browser: Mozilla Firefox/ Google Chrome e.t.c with updated Versions.

**6. Assumptions and Dependencies**

The Doctor Appointment system is dependent on the following assumptions:

• Patients and doctors will have access to a device with an internet connection.

• Patients and doctors will be able to use the system without any additional training.

• The system will be integrated with an SMS notification service to remind patients of their appointments.

**7. Glossary**

• Java8: A programming language used for developing applications.

• Spring Boot: A framework used for building Java applications.

• Hibernate: An object-relational mapping framework used for database access.

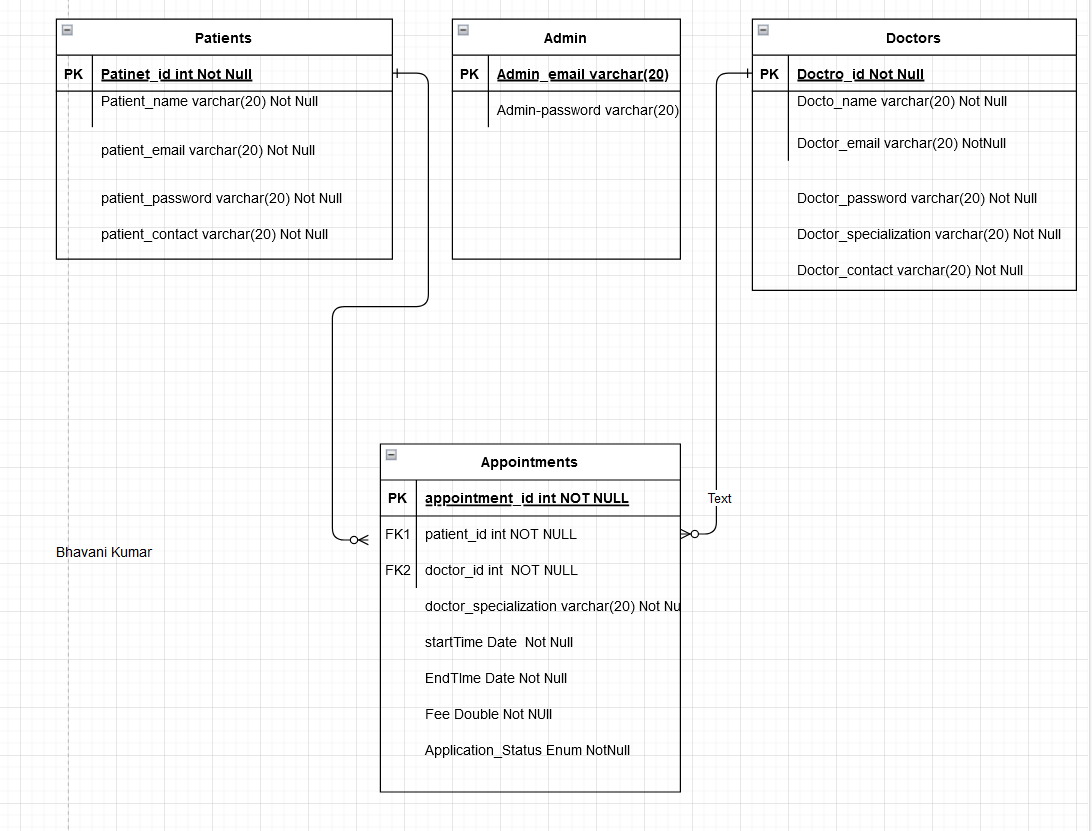
• MySQL: An open-source relational database management system.

• Specialization: The area of medicine in which a doctor is trained and practices.

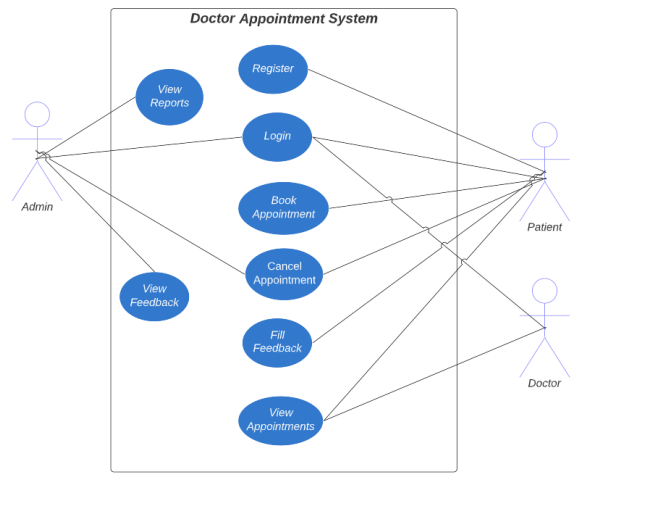
• User Authentication: The process of verifying the identity of a user.

**Phase -2**

ER Diagram:

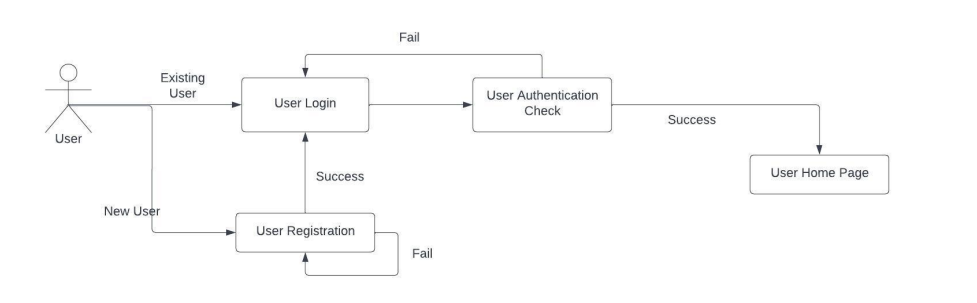


USE CASE DIAGRAM:

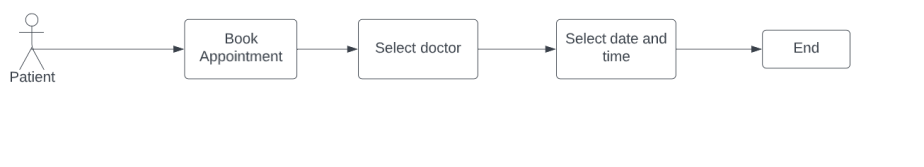


MODULES :

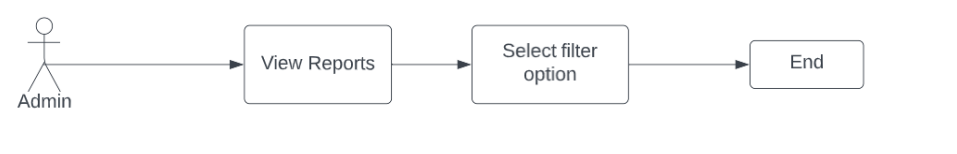
● Login: Login system for all types of roles (admin, doctor, patient).



● Appointment booking: Users can book appointments for their required date and time.



● Admin View Reports: Admin can view patient and doctor data based on certain filters.



**Structure of the project :**

**Entities :**

1.Patient Entity

2. Doctor Entity

3. Admin Entity

4. Appointment Enitty

**Controllers:**

1.Doctor Controllers

1. Add Appointment
2. Registeration
3. Login

2. Patient Controllers

* Login Patient
* Add Appointment
* Register Patient

3.Admin Controller:

* View All patients
* View All Doctors
* View All Appointments
* Login Admin

**Services:**

1.Admin Service

* getAllDoctor()
* getSpecializations()
* getAllPatient()
* Login()
* getAllAppointment()

2.Patient Service

* LoginPatient()
* RegisterPatient()
* GetAvailable Appointments() By filtering specialization
* Book Available Appointments()

3.Dcotor Service

* Register Doctor()
* Login Doctor()
* Add Appointment()