USE CASE DIAGRAM DESCRIPTION

Doctors Listing and Appointment Application

**Introduction**

The Doctor Appointment System UML Diagram Document provides an overview of the system's architecture, functionalities, and interactions between the actors "Patient" and "Admin." This document aims to assist in understanding the system's design and facilitate communication among stakeholders involved in the development and management of the system.

**Use Case Diagram**

**Purpose:** The Use Case Diagram illustrates the interactions between the actors (Patient and Admin) and the system's functionalities.

**Key Elements:** The diagram includes the Patient actor, Admin actor, and their respective interactions with the system.

**Relationships:** The Patient actor initiates use cases such as "View Doctors List," "Search Doctor," "Book Appointment," "Cancel Appointment," "View Appointment List," "Add Rating," and "View Rating." The Admin actor is responsible for managing doctors, appointments, and ratings.

**Diagram Notation:** The Use Case Diagram adheres to standard UML notation, using actors, use cases, and associations.

**Use Case Descriptions**

**1.View Doctors List**

Purpose: Allows the patient and admin to view the list of available doctors.

Actors: Patient, Admin

Preconditions: None

Flow of Events: The patient or admin selects the "View Doctors List" option, and the system displays the list of doctors with their information.

Postconditions: The list of doctors is successfully displayed.

**2.Search Doctor**

Purpose: Allows the patient to search for a specific doctor.

Actors: Patient

Preconditions: None

Flow of Events: The patient provides search criteria (e.g., name, specialty), and the system displays the matching doctors.

Postconditions: The search results are successfully displayed.

**3.Book Appointment**

Purpose: Allows the patient to book an appointment with a doctor.

Actors: Patient

Preconditions: The patient must be registered in the system.

Flow of Events: The patient selects a doctor and preferred appointment date/time. The system checks the doctor's availability and confirms the appointment.

Postconditions: The appointment is successfully booked.

**4.Cancel Appointment**

Purpose: Allows the patient to cancel a previously booked appointment.

Actors: Patient

Preconditions: The patient must have a booked appointment.

Flow of Events: The patient selects the appointment to be cancelled. The system confirms the cancellation and updates the appointment status.

Postconditions: The appointment is successfully cancelled.

**5.View Appointment List**

Purpose: Allows the admin to view the upcoming appointments.

Actors: Admin

Preconditions: The patient must have one or more upcoming appointments.

Flow of Events: The system retrieves and displays the patient's upcoming appointments.

Postconditions: The appointments are successfully displayed.

**6. Add Rating**

Purpose: Allows the patient to add a rating for a doctor.

Actors: Patient

Preconditions: The patient must have a completed appointment with the doctor.

Flow of Events: The patient selects the doctor and provides a rating and feedback. The system stores the rating.

Postconditions: The rating is successfully added.

**7.View Rating**

Purpose: Allows the patient to view the ratings and feedback for a doctor.

Actors: Patient

Postconditions: The ratings and feedback for the selected doctor are successfully displayed to the patient, allowing them to make informed decisions based on the experiences of other patients.

Flow of Events: The ratings and feedback are presented to the patient, providing an overview of the doctor's performance and patient satisfaction.

**8.Add Doctor**

Purpose: Allows the admin to add a new doctor to the system.

Actors: Admin

Preconditions: The admin must be authenticated and have administrative privileges.

Flow of Events: The admin provides the necessary information for the new doctor, such as name, specialty, and contact details. The system adds the doctor to the system's database.

Postconditions: The new doctor is successfully added to the system.

**9.Update Doctor**

Purpose: Allows the admin to update the information of a doctor.

Actors: Admin

Preconditions: The admin must be authenticated and have administrative privileges.

Flow of Events: The admin selects a doctor and modifies their information, such as name, specialty, or contact details. The system updates the doctor's information in the database.

Postconditions: The doctor's information is successfully updated.

**10.Delete Data**

Purpose: Allows the admin to delete data from the system, including doctors, appointments, or ratings.

Actors: Admin

Preconditions: The admin must be authenticated and have administrative privileges.

Flow of Events: The admin selects the data to be deleted (e.g., doctor, appointment, rating). The system removes the selected data from the system's database.

Postconditions: The selected data is successfully deleted from the system.