Hospital Appointment Scheduling System

–

Document ID: ASP-CMS-SRS-02.doc

Author(s): Janani D P

Version: 0.2

Date: March 21, 2024

1. Abstract

The Hospital Appointment Scheduling System is designed to simplify the process of scheduling and managing appointments for patients in a hospital environment. This system enables users to easily book appointments, view available time slots, and receive reminders for scheduled appointments. Administrators have access to features such as managing time slots, viewing appointments, and sending reminders to patients. Users can login in the system and book appointments in the available time slots. The system ensures smooth coordination between healthcare providers and patients, leading to an efficient scheduling and communication.

2. Objective and Scope

The objective and the scope of the project is to build a Hospital Management System to improve and simplify the administrative procedures in a hospital setting for patients. The scope of the product includes the following basic features:

- Enable users to securely access the Hospital Patient Management System with personalized credentials.
- Facilitate the booking of patient appointments and time slot management.
- Simplify appointment allocation to healthcare providers within the HPMS.
- Reduce missed appointments by sending timely reminders to patients.

3. Project End Users

The end users of HPMS are hospital administrators and patients.

3.1Administrators

- Create a simple interface for admins to view, add, and manage patient appointments.
- Implement a feature to send basic appointment reminders to patients via email or SMS.

3.2Patients

- Develop a user-friendly booking system for patients to schedule appointments online.
- Allow patients to view their past and upcoming appointments within the system.

4. Features

4.1Login to the system

The system validates the entered information, ensuring that all fields are filled correctly and that the username and email are unique. Then, the system creates a new user account and stores the provided information in the database. Each and every user should be authenticated with a User Name and Password to login into the system. If the credentials are correct, the user is authenticated and granted access to their account/dashboard.

The admin enters their admin username and password on the login page. The system authenticates the credentials and grants access to the admin dashboard upon successful verification.

4.2Add/Update time slots

For administrators, the system enables easy management of appointment time slots, allowing for the addition, viewing, and updating of available time slots to accommodate patient needs.

4.3View time slots

On the user side, patients can conveniently view and select from updated time slots, providing flexibility in scheduling appointments within the hospital system.

4.4Appointment Booking

Patients select a preferred appointment date and time from the available slots and provide their contact information.

Once submitted, the system confirms the appointment and sends a confirmation message to the patient.

4.5Add/View/Delete Appointments

Admins can log into the system and view a list of existing appointments.

They can add new appointments by specifying the patient's details, appointment date, and time.

Users can also cancel their appointments after booking.

4.6Appointment Reminders

Admins can set up the reminder system by configuring the timing and method of reminders.

5. Functional and Non-Functional Requirements

5.1Functional Requirements

Login: User and admin can login into the system using the suitable credentials

Add time slots: Admins can add new time slots for appointment booking.

Update time slots: Admins can modify existing time slots, including date, start and end time.

View time slots: Users can view available time slots for appointment booking.

Appointment booking: Users can select an available time slot and book an appointment.

View appointment: Users and admin can view their scheduled appointments, including date, time, and healthcare provider.

Delete appointment: Users can cancel or delete their scheduled appointments.

Appointment reminder: Reminders are sent to users before scheduled appointments by admin side.

5.2Non-Functional Requirements

The system should support a large number of time slots without slowing down. The user interface for viewing time slots should work well on different devices. Patient information should be encrypted and stored securely. Canceling appointments should not lead to data loss. Users should confirm before deleting appointments to avoid mistakes. Access to appointment details should be restricted to authorized users.

6. Low Level Design

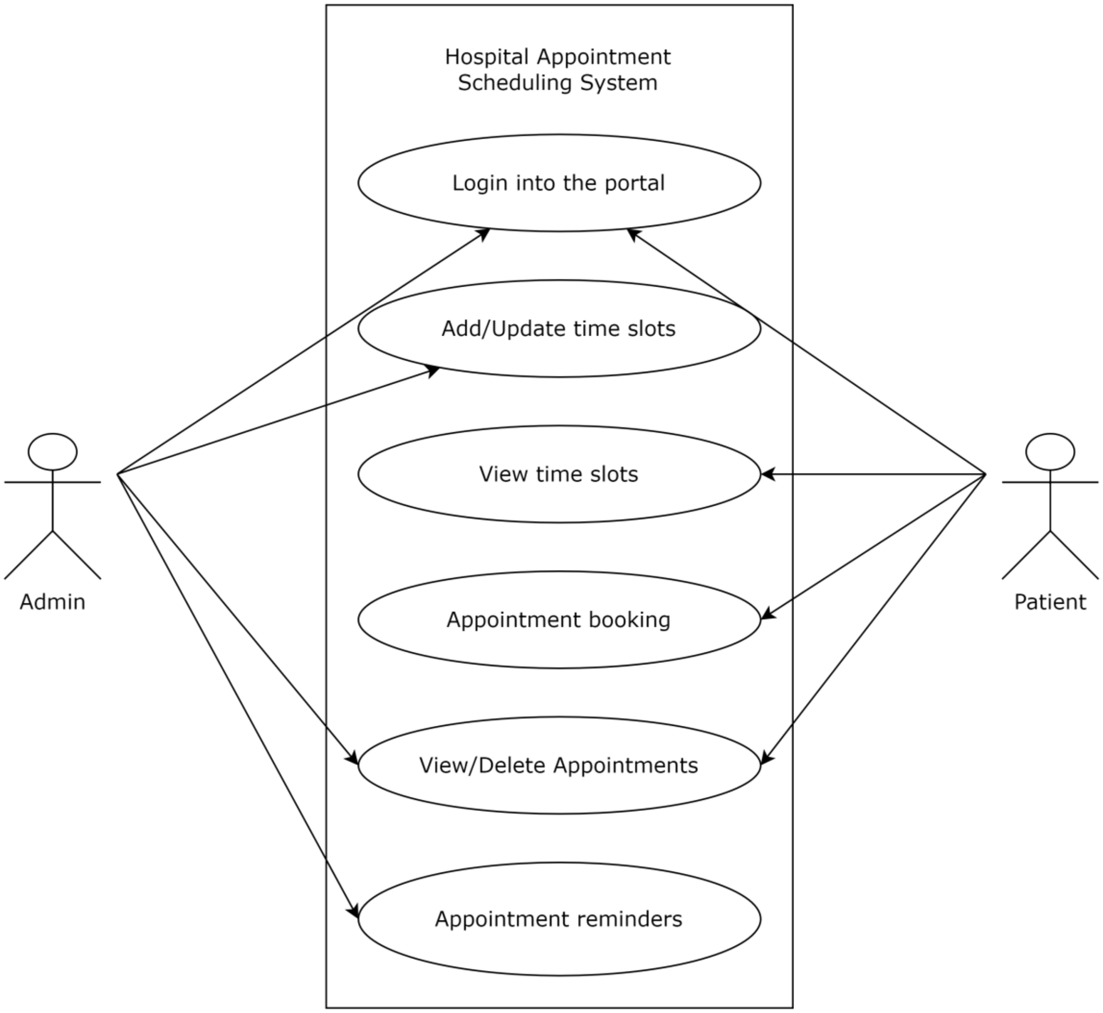
For the low-level design of the mentioned features, a database schema is established with a table for time slots, including fields for date, start time, end time, and availability. Backend logic will consist of API endpoints to handle operations such as adding, updating, and viewing time slots. These endpoints will validate inputs and interact with the database accordingly. The frontend interface will present available time slots to users in a user-friendly format. For appointment booking, another API endpoint will validate inputs, check availability, and store booking details in the database. The frontend will provide a booking interface for users to select time slots and provide necessary information. To view appointments, a backend endpoint will retrieve booked appointments for a specific user, while the frontend will display them in a readable format. Deleting appointments will involve backend logic to cancel appointments and remove corresponding records from the database, with the frontend providing a cancellation interface with a confirmation prompt. For appointment reminders, a background task scheduler will send reminders based on scheduled appointment times, with a notification service handling the actual delivery via email, SMS, or other channels according to user preferences.

7. High Level Design

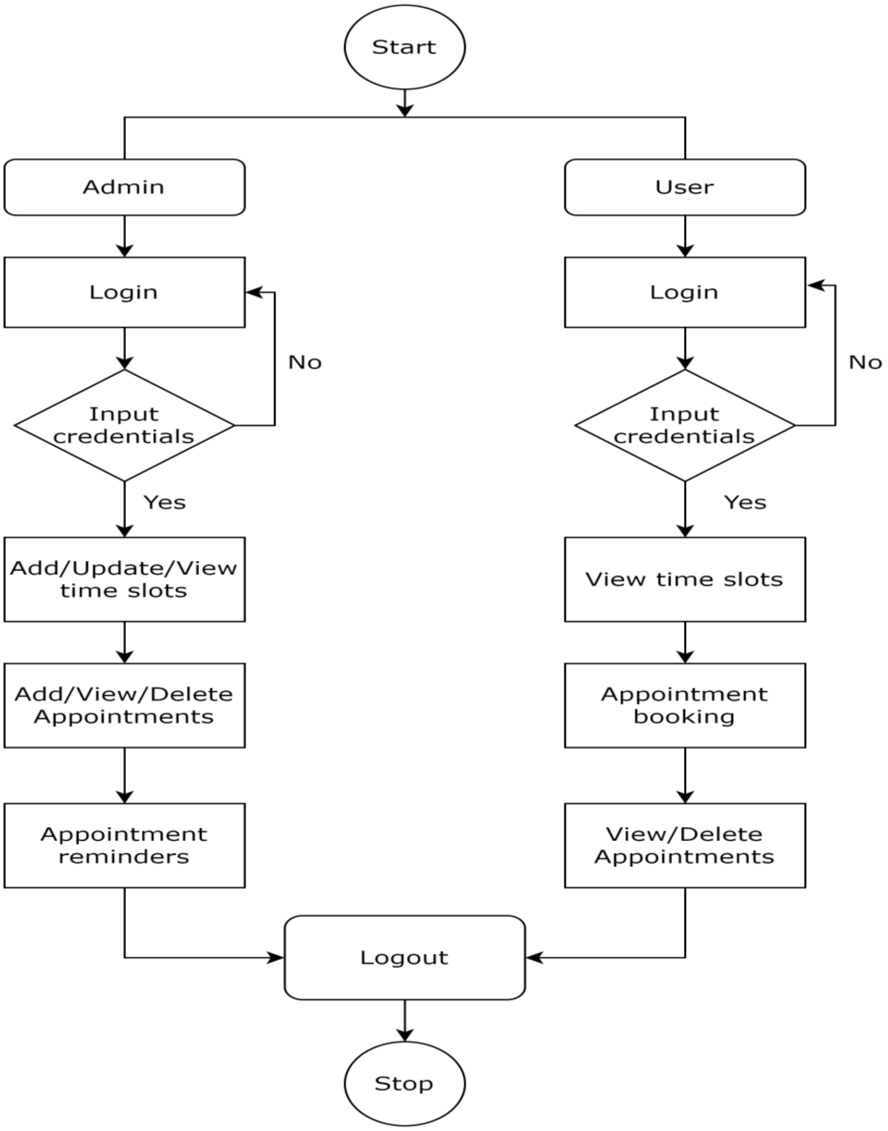
For the high-level design, a client-server architecture is established. The client-side will consist of a web application accessible to both admins and users. Admins will have additional privileges to manage time slots and appointments. The database will be relational, with tables for users, time slots, appointments, and reminder settings. Additionally, a background task scheduler to handle appointment reminders is incorporated. The system will follow RESTful principles for API design, ensuring scalability and maintainability. Security measures such as encryption, authentication, and authorization will be implemented to protect sensitive data. Finally, the system will be deployed on a cloud platform like AWS or Azure for scalability and availability.

8. Diagrams

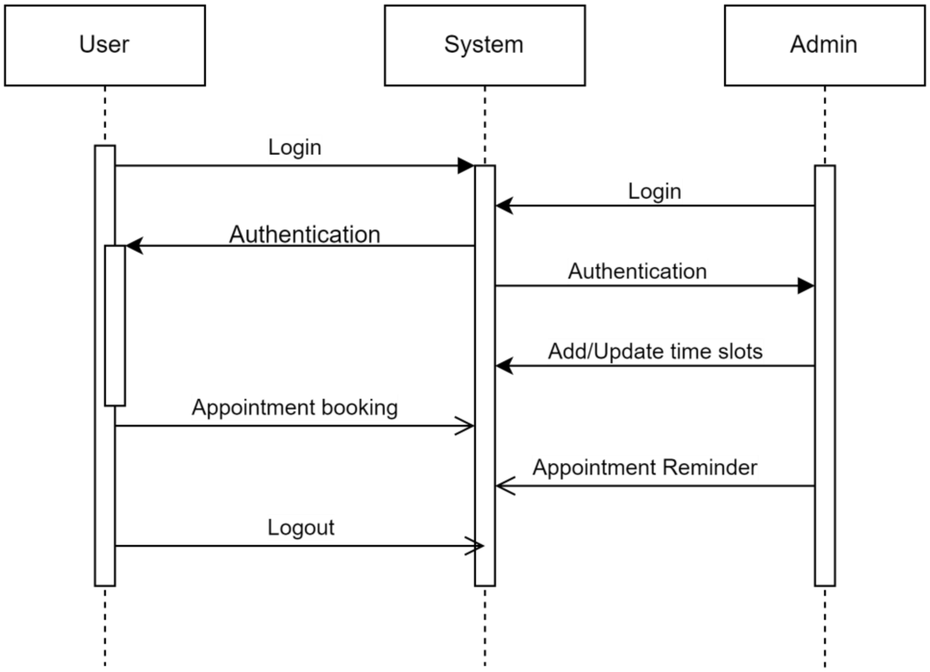
8.1Use Case Diagram



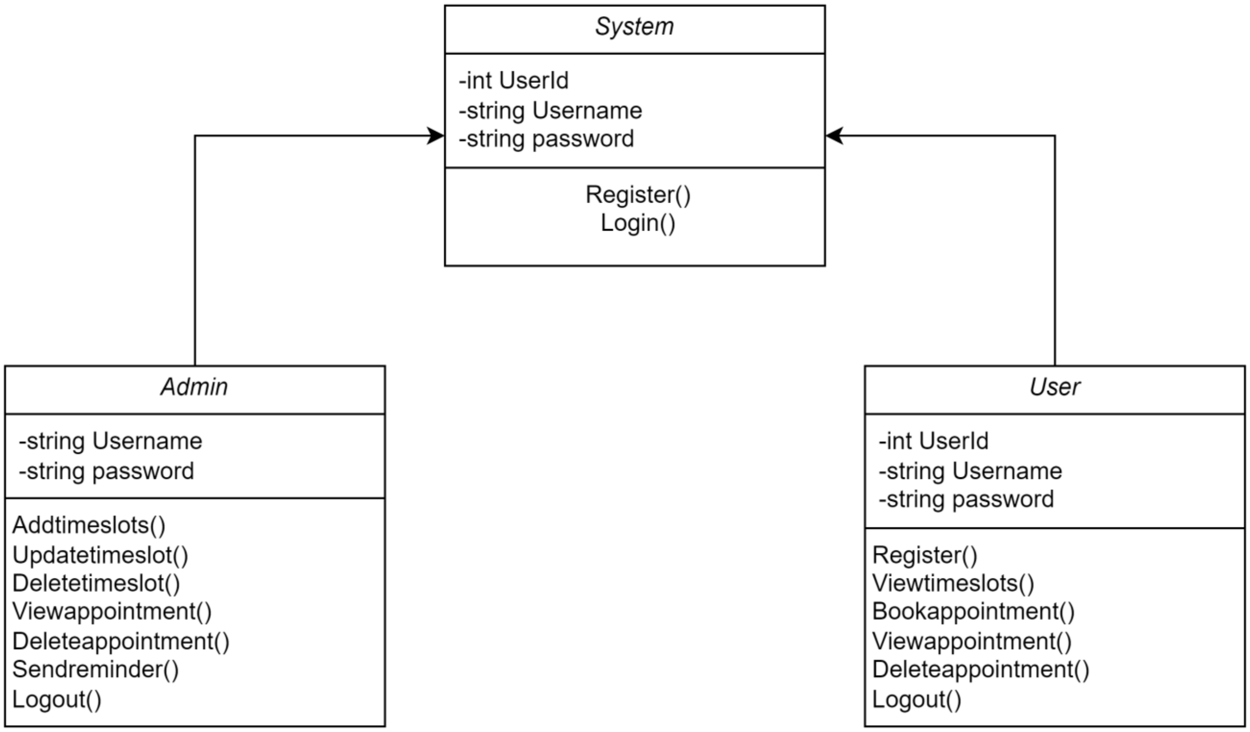
8.2Flow Diagram



8.3Sequence Diagram



8.4Class Diagram


System
-int UserId
-string Username
-string password
Register()
Login()
Admin
-string Username
-string password
Addtimeslots()
Updatetimeslot()
Deletetimeslot()
Viewappointment()
Deleteappointment()
Sendreminder()
Logout()
User
-int UserId
-string Username
-string password
Register()
Viewtimeslots()
Bookappointment()
Viewappointment()
Deleteappointment()
Logout()