

USE_CASE: JP014: Develop a web-based tool for medical professionals to easily store and access patient records.

Introduction - EHR (Electronic Health Record):

An Electronic Health Record (EHR) is a digital version of a patient's medical history. Unlike traditional paper charts, EHR software provides a centralized platform to store and manage a patient's complete medical information in a secure and accessible format.

Software Requirements:

- **Front-end**
 - HTML
 - CSS
 - JavaScript
- **Middle-Layer**
 - JSP
- **Back-end**
 - Java
 - Apache HTTP server
 - Apache Tomcat Server
 - MySQL

Description about technologies used:

- **Front-end**
 - HTML: It is used to structure a web page and its content.
 - CSS: It is used to style and layout web pages.
 - JavaScript: It is used to make web pages interactive.
- **Back-end**
 - JSP: Simplifies creating dynamic web pages by combining HTML with Java code.
 - Java: It is used to create backend web application using servlets or JSP.
 - Apache HTTP Server: It is used to accept HTTP requests from visitors and send them back the requested information.
 - Apache Tomcat Server: It is used as a servlet container in the webserver, and is used to serve dynamic webpages.
 - MySQL: It is used to store data in tables that map to objects.

I. Functional Requirements: (Before Login)

Number of Actors: 3

Actors Name

1. Admin
2. Doctor/Staff (Employee)
3. Patient/User (End user)

1. Requirements for Admin: (Before Login)

Login requirements for Admin:

- **Front-End Requirements:**
 - User name
 - Password
 - Login button
 - Forgot Password?
- **Back-End Requirements:**
 - Send data to the back end using HTTPS and the data like password must be hashed and salted.
 - Verify user name and password
 - When the admin forgotten his/her password, then she/he clicks the Forgot Password? Link to reset his/her password
 - If credentials are correct then grant access, if not then deny access

2. Requirements for Doctor/Staff: (Before Login)

Login requirements for Doctor/Staff:

- **Front-End Requirements:**
 - User name
 - Password
 - Login button
 - Forgot password?
- **Back-End Requirements:**
 - Send data to the back end using HTTPS and the data like password must be hashed and salted
 - Verify user name and password.
 - When the doctor forgotten his/her password, then she/he clicks the Forgot Password? Link to reset his/her password
 - If credentials are correct then grant access, if not then deny access.

3. Requirements for Patient/User: (Before Login)

Registration requirements for Patient/User:

➤ **Front-End Requirements:**

- User name
- First name
- Last name
- Email Address
- Birth date
- Phone number
- Password
- Confirm password
- Register button
- Already have an account? (Provide redirect link to login page).

➤ **Back-End Requirements:**

- Send data to the back end using HTTPS and the data like password must be hashed and salted
- If the password and confirm password is matches then the user account is created

Login requirements for Patient/User:

➤ **Front-End Requirements:**

- Email Address
- Password
- Login button
- Don't have an account? (Provide a registration link)
- Forgot password? (Provide a forgot password link to reset password via email)

➤ **Back-End Requirements:**

- Send data to the back end using HTTPS and the data like password must be hashed and salted
- Verify username and password
- When the doctor forgotten his/her password, then she/he clicks the Forgot Password? Link to reset his/her password
- If credentials are correct then grant access, if not then deny access
- If forgot password then send the web page to reset password

II. Functional Requirements: (After Login)

Number of Actors: 3

Actors Name

1. Admin
2. Doctor(Employee)
3. Patient/User (End user)

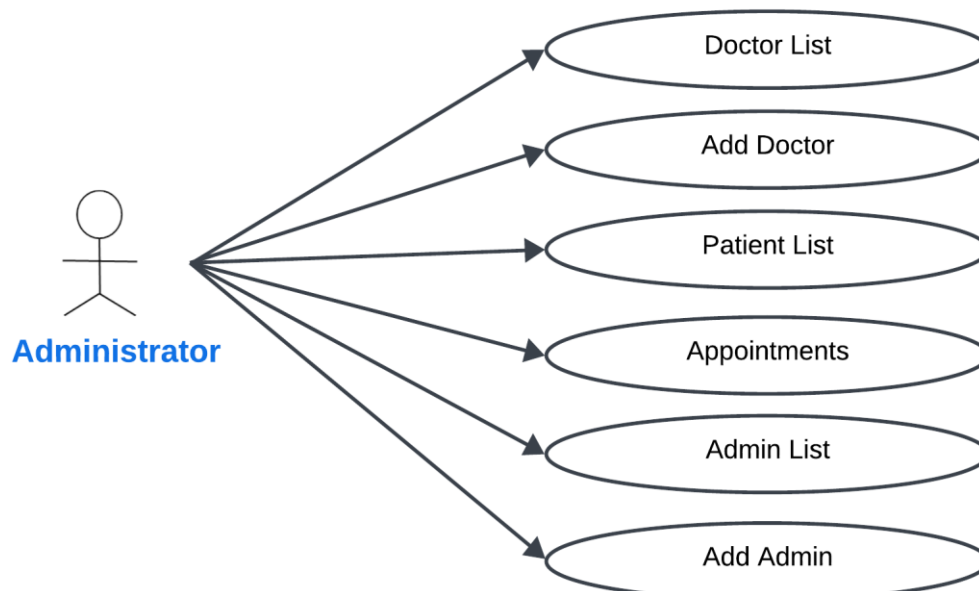
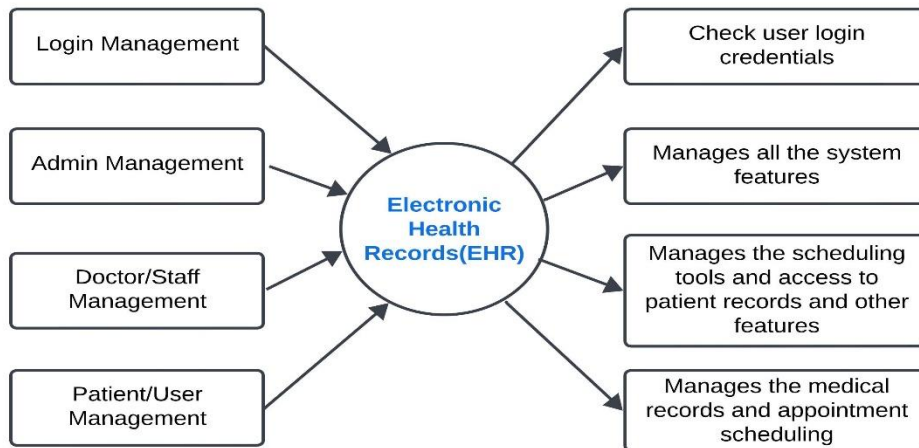
1. Requirements for Admin: (After Login)

➤ Front-End Requirements:

- Dashboard
 - Doctor List (Below this provide a “*view Doctors*” link to view all the Doctors)
 - Add Doctor(Below this provide a “*Add Doctor*” link to add a new Doctor)
 - Patient List (Below this provide a “*view Patients*” link to view all the Patients)
 - Appointments (Below this provide a “*Appointments*” link to view all the “Visited” and “Cancelled” Appointments)
 - Admin List (Below this provide a “*view Admins*” link to view all the Admins)
 - Add Admin(Below this provide a “*Add Admin*” link to add a new Admin)
 - Logout (Logs out from the Dashboard).

➤ Back-End Requirements:

- Admin can delete the Doctors, Patients, and also Admins by clicking the “Delete” button is present in each and every rows.



Application Flow:

Admin > Login > Dashboard > Can view all the Doctors List > Logout

Admin > Login > Dashboard > Can add a new Doctor > Logout

Admin > Login > Dashboard > Can view all the Patients List > Logout

Admin > Login > Dashboard > Can view all the Appointments List > Logout

Admin > Login > Dashboard > Can view all the Admins List > Logout

Admin > Login > Dashboard > Can add a new Admin > Logout

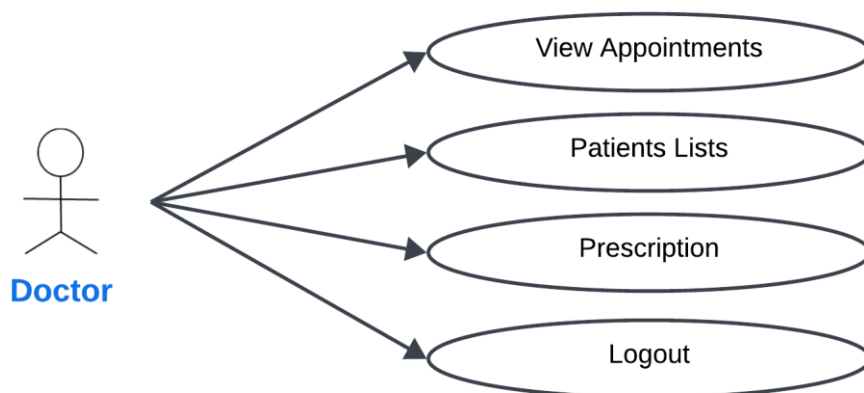
2. Requirements for Doctor/Staff: (After Login)

➤ Front-End Requirements:

- Dashboard
 - View Appointment (Below this provide a “*View Appointment*” link to view all the appointments from the patients side doctor can **Accept** or **Decline** appointments there itself)
 - Patients List (Below this provide a “Patients List” to view all the patients)
 - Prescription (Below this provide a “Prescription” link to prescribe a patient as per his/her Appointments)
 - Logout

➤ Back-End Requirements:

- Doctor can Accept or Decline the patients appointments
- And also doctor can prescribe a prescription for patient’s appointments.



Application Flow:

Doctor > Login > Dashboard > Can view all the Appointment Lists > Logout

Doctor > Login > Dashboard > Can view all the Patient's Lists > Logout

Doctor > Login > Dashboard > Can prescribe the Prescription > Logout

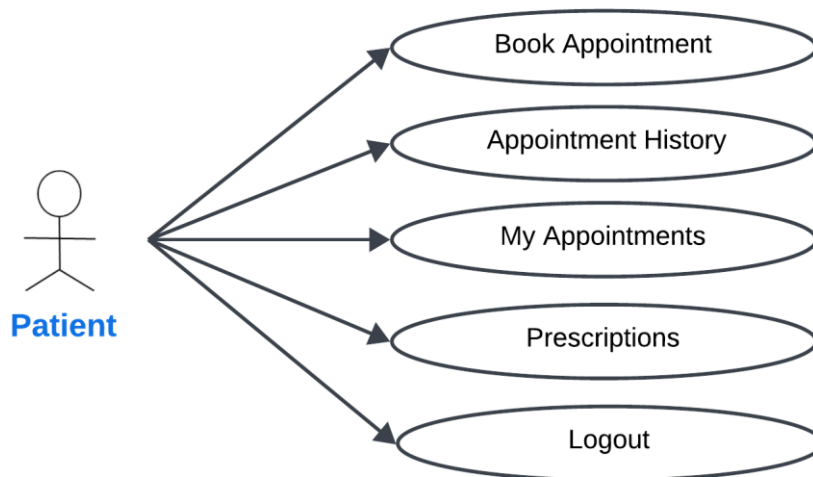
2. Requirements for Patient/User: (After Login)

➤ Front-End Requirements:

- Dashboard
 - Book Appointment (Below this provide a “*Book Appointment*” link to book an appointment)
 - Appointment History (Below this provide a “*Appointment History*” to view all the “Cancelled” and “Deleted” appointments)
 - My Appointments (Below this provide a “*My Appointments*” link to view all the “Approved” and “Awaiting” appointments)
 - Prescriptions (Below this provide a “Prescriptions” link to view all the prescriptions prescribed by the specific doctor)
 - Logout

➤ Back-End Requirements:

- Patient can view all the prescriptions prescribed by the doctor



Application Flow:

Patient > Login with credentials > Dashboard > Book Appointment > Logout

Patient > Login with credentials > Dashboard > Appointment History > Logout

Patient > Login with credentials > Dashboard > My Appointments> Logout

Patient > Login with credentials > Dashboard > Can view the Prescriptions prescribed by the doctor> Logout

