

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
- Bootstrap
- AJAX

➤ Back-end

- JSP
- 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.
- Bootstrap: For easy styling of webpages.
- AJAX: It enables a web application user to interact with a web page without the interruption of constant web page reloading

➤ 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
 - Enter verification code after the details submission (Only if you haven't logged in for 1 day)
- **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
 - Multi-factor authentication for Registration (send a 6 digit code for verification via email, only if not logged in for 1 day).
 - 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? (Provide a forgot password link to contact admin).
 - Enter verification code after the details submission (Only if you haven't logged in for 2 day)
- **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.
 - Multi-factor authentication for Registration (send a 6 digit code for verification via email, only if not logged in for 2 day).

- If credentials are correct then grant access, if not then deny access.
- If forgot password then send the request to Admin

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

Registration requirements for Patient/User:

➤ Front-End Requirements:

- User name
- First name
- Last name
- Email
- Birth date
- Phone number
- Password
- Confirm password
- Register button
- Already have an account? (Provide redirect link to login page).
- Enter verification code after the details submission (Only if you haven't logged in for 2 day)

➤ Back-End Requirements:

- Send data to the back end using HTTPS and the data like password must be hashed and salted
- Verify if the user name already exists
- Multi-factor authentication for Registration (send a 6 digit code for verification via email, only if not logged in for 2 day).
- If the code matches then the user account is created

Login requirements for Patient/User:

➤ Front-End Requirements:

- User name
- Password
- Login button
- Don't have an account? (Provide a registration link)
- Forgot password? (Provide a forgot password link to reset password via email)
- Enter verification code after the details submission (Only if you haven't logged in for 2 day)

➤ 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

- Multi-factor authentication for Registration (send a 6 digit code for verification via email, only if not logged in for 2 day)
- 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/Staff (Employee)
3. Patient/User (End user)

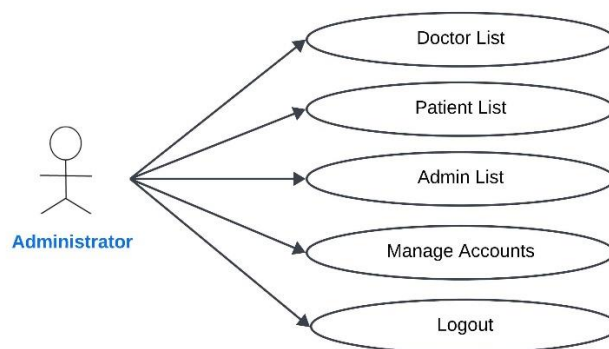
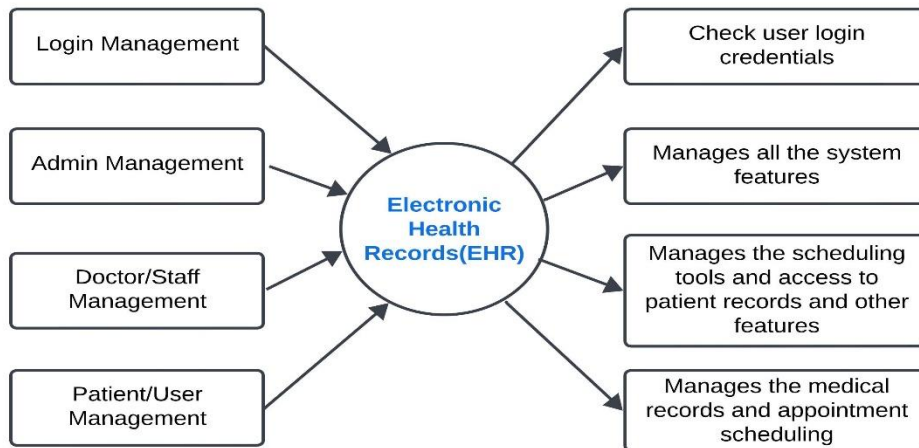
1. Requirements for Admin: (After Login)

➤ Front-End Requirements:

- Dashboard
 - Doctor List (Below this provide a “*view Doctors*” link to view all the Doctors)
 - Patient List (Below this provide a “*view Patients*” link to view all the Patients)
 - Admin List (Below this provide a “*view Admins*” link to view all the Admins)
 - Manage Accounts : (MENU)
 1. Admin Accounts: CRUD operations on Accounts
 2. Doctor Accounts: CRUD operations on Accounts
 3. Patient Accounts: CRUD operations on Accounts
 - Logout

➤ Back-End Requirements:

- CRUD operation on accounts
- CRUD operation on other data



Application Flow:

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

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

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

Admin > Login > Dashboard > Manage Accounts > Admin Accounts > Logout

Admin > Login > Dashboard > Manage Accounts > Doctor Accounts > Logout

Admin > Login > Dashboard > Manage Accounts > Patient Accounts > Logout

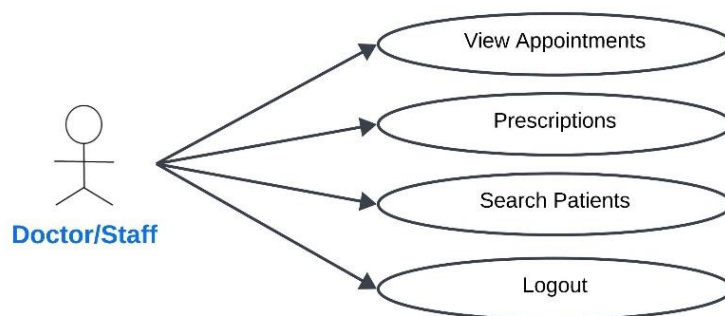
2. Requirements for Doctor/Staff: (After Login)

➤ Front-End Requirements:

- Dashboard
 - View Appointments (Below this provide a “*Appointment List*” to view all the appointments)
 - Prescriptions (Below this provide a “*Prescription List*” to view all the Prescriptions)
 - Search Patients (Search patients using Username or Phone Number)
 - Logout

➤ Back-End Requirements:

- Create, Update and Read accounts
- CRUD operation on other data



Application Flow:

Doctor/Staff > Login > Dashboard > Can view Appointment Lists > Logout

Doctor/Staff > Login > Dashboard > Can view all the Prescription Lists > Logout

Doctor/Staff > Login > Dashboard > Search Patients > Can view all the Prescription Lists > Can prescribe medicines to the patients there itself > Logout

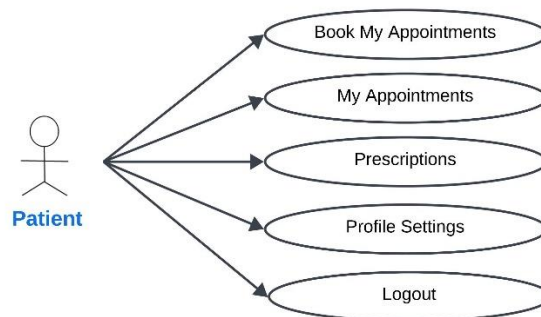
2. Requirements for Patient/User: (After Login)

➤ Front-End Requirements:

- Dashboard
 - Book My Appointments (Below this provide a “*Book Appointment*” link to book an appointment with Doctor)
 - My Appointments (Below this provide a “*View Appointment History*” link view the previous appointment history)
 - Prescriptions (Below this provide a “*View Prescription List*” link view the prescription lists from Doctor side)
 - Profile Settings (Can change E-Mail ID and Phone Number)
 - Logout

➤ Back-End Requirements:

- Create, Update, Read account details like profile
- CRUD operations on other data



Application Flow:

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

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

Patient > Login with credentials > Dashboard > Can view Prescriptions Lists> Logout

Patient > Login with credentials > Dashboard > Can update the Profile Settings like E-Mail Id and Phone Number > Logout

