


Database / Data Layer design

User

- Id
- Email
- Password
- RoleId

Role

- Id
- Name

→ System Admin
→ Community Admin
→ Doctor
→ Patient

System init : Roles setup();
System Admin Setup();

Person

- Id
- Name
- UserId
- HouseId aka residence

Doctor

- Id
- PersonId
- HospitalId
- Specialization

Patient

- Id
- PersonId

House

- Id
- Name

Appointment

- Id
- PatientId
- DoctorId
- Date

Encounter

- Id
- AppointmentId
- Vital BloodPressure
- Vital Temperature
- Vital Height
- Vital Weight

System Admin

Hospital

- Id
- CommunityId
- Name
- Address

Community

- Id
- Name
- CityId

City

- Id
- Name

Community Admin

System Admin

User Type :

Doctor

Patient

Community Admin

System Admin.

Doctor

Patient

See Appointments

Book Appointment

Cancel Appointment

View Appointment

Encounter

NotesId
DoctorId
AppointmentId
Blood Pressure
Height
Weight
Temperature
should be visible to Admin & to Doctor
Patient - should see all past encounters

Supported Operations

System Admin :

- CRUD on Cities
- CRUD on Communities
- CRUD on Hospitals.
- CRUD on Users
 - other System Admins
 - Community Admins
 - Patients
 - Doctors
- CRUD on Appointments
- CRUD on Encounters

Community Admin

- CRUD on its Community
- CRUD on Hospitals in Community.
- CRUD on Doctors of this Community
- CRUD on Patients who have appointments with doctors in community
- CRUD on Community Admins of this community
- CRUD on Appointments of this Community.
- CRUD on Encounters

Doctor

- Read Patient with Appointments
- RUD existing Appointments
- CRUD on his/her encounters

Patient:

- Read on Cities
- Read on Communities
- Read on Hospitals
- Read on Doctors
- CRUD on his/her Appointments
- Read on his/her encounters.

Application design:

Model

Datalayer

Store <User>
Store <Doctor>
:
:

All the stores

each Store:

- save
- retrieve
- filter
- filter with role
- getAll
- delete

Services

Userservices

Doctorservices

:
:

C

Controller

- Login functionality. (Email, password, roleID)
 - make sure actions are restricted based on roleID
- roleID permission check

System Admin Controller

Community Admin Controller

Doctor Controller

Patient Controller

Base Controller

- gives Role
- gives User State
- maintain User State

✓

View (UI)

System Admin

Community Admin

Doctor

Patient

hashmap:
 $\langle id, T \rangle$

id	Hospital	Ad
1	H1	A1
2	H2	A1
3	H3	A2

$\text{List} \leftarrow \text{nodes} \rightarrow \text{filter}(n \rightarrow n.\text{getAddress}().\text{equals}("A1"));$ $\rightarrow H_1, H_2$

$\text{List} \langle \text{hospital} \rangle \rightarrow \text{filterWithId} (\dots) \rightarrow \text{Pair} \langle 1, H1 \rangle$
 $\text{List} \langle \text{Pair} \langle \text{Integer}, \text{Hospital} \rangle \rangle \rightarrow \text{Pair} \langle 2, H2 \rangle$