HOSPITAL – PATIENT MANAGEMENT SYSTEM

This application is a platform for multiple users to find and manage resources. The key players in this application are Hospital Admin, Community Admin, System Admin, Doctor, Person, and Patient. Since the application is built without using database, the way we deal with the data here is by creating object and modifying it whenever required. We are using array list to store data in respective directories.

**Use Case 1: Hospital Admin**

Hospital Admin logins by entering the login credentials. The hospital admin can create new hospital, read, or view a list of hospitals, update details related to the hospital and delete hospital data as well. These details are stored in the hospital directory array list. The hospital admin can also create doctor by entering doctor details. These details are stored in the doctor directory. The hospital Admin can not only view doctors but also modify their data. Hospital admin can update and delete doctors as well.

**Use Case 2: Doctor**

Doctor logins by entering the login credentials. The doctor can view the vital signs details of the incoming patients and then decide to diagnose or leave patients based on their vital signs. The patient vital signs are mere patient data and hence are stored in the vital sign directory. Doctor has access to the encounter history table, which is just a record of patients and their visit history to the doctor’s clinic. Doctor can also view the patients’ records stored in the vital sign directory.

**Use Case 3: Community Admin**

Community Admin logins by entering the login credentials. Community admin can view and modify the information under a community and city. The Community admin has search functionality which can be used to retrieve a list of all doctors and hospitals based in a city. Community admin can view and modify information for House community and Hospital community. Based on the model diagram provided, it is safely assumed that every person lives in house which has an address and city attached to it. So, the same person directory can be used for house community as well. Community admin can also modify the address, location information of hospital and houses.

**Use Case 4: Person**

Since this application is primarily aimed towards people finding doctors and hospitals, a person can login without entering login credentials. The person fills in some basic information like name, address, etc. This corroborates the assumption that every person lives in a designated house. Similarly, the house directory n adperson directory is one and the same. The person is then encouraged to search hospitals and doctors by entering the intended city, they wish to look hospitals and doctors for. The person is then encouraged to make a visit to the choice of doctor, depending on the requirement.

**Use Case 5: Patient**

Doctors can create and view patients. The patient once visiting the doctor’s clinic is asked to fill in the vitals such as blood pressure, temperature, etc. This data is stored in the patient vitals directory, which is accessible to the doctor. A visit the doctor’s clinic is an encounter, and such encounter records are stored in the encounter history directory.

**Supplementary Use Case: System Admin**

The system admin has access to all the directories and can perform CRUD operations on all types of use cases. In this case, the system admin itself becomes a silent player in the application and works in the background as we run a demo of the application.