NOTES

1. UI/UX - alignments and panel css (colors)
2. Data Types - use relevant data types - think of the data attributes for Person, Doctor, House, etc classes -Do not use String for all fields. Use INT, DATE, LONG, Boolean as required.
3. Validations - as stated - will be tested on NULL and data types ONLY - eg: Age of type INT should not accept String or Null.
4. Error handling - No errors on your terminal - use try, catch wherever necessary | Forms should not be submitted with Null values.
5. Demo of the application should be smooth - not stop due to error
6. All 15 classes need to be implemented - you can extend it if you have to, but do not delete a class
7. INHERITANCE should be applied on relevant classes
8. Panels - At least 5 panels required: System Administrator, Doctor, Patient, Community Admin and Hospital Admin
9. CRUD can be implemented in the same panel - e.g: A Hospital Administrator can perform all CRUD operations on Doctors in the same panel
10. README file is important
11. Commit history - more than 5 logical commits are required
12. UNIQUE ID - for a person, doctor, encounter wherever valid is necessary
13. NO DATABASE required - USE ARRAYLIST or suitable DATA STRUCTURES for the DIRECTORIES
14. person class - at least 5 attributes. For the other classes(DOCTOR, HOSPITAL, HOUSE, VITAL, ENCOUNTER) you can think of at least 3 primitive attributes apart from the non-primitive attributes that connect the classes.
15. Sequence diagram - 1 Flow from System admin perspective 1 - Doctor flow 1 - Patient flow
16. Please submit your class diagram with right notations, arrows and attributes. Push the diagrams to GitHub, add it to the project folder before your final push.See how you can add a link to it in your README or attach as an image.