National COVID Management System

Solution Document

Version 1.0

Shammi Kolonne

National COVID Management System	Version :	1.0
Solution Document	Date :	19/04/2021
Solution Document - Final Draft		

Revision History

Date	Version	Description	Author
19/04/2021	0.1	Initial Draft	Shammi

Table of Contents

National COVID Management System	Version :	1.0
Solution Document	Date :	19/04/2021
Solution Document - Final Draft		

Solution Document

1. Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive architectural and design overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural and design decisions made regarding the system.

Throughout the document, the layout of the system will be described with the use of diagrammatic representations in order to give stakeholders an insight into the workings of the system being implemented.

1.2 Scope

This document shall provide a number of architectural views that will help different stakeholders understand the system in a way convenient to them.

The document will,
- Show the end-users the components of the system along with their relationships and interactions.
- Let the integrators view the processes and workflow rules of the system and how they communicate with each other.
- Provide programmers and software managers a building block view of the system.
- Show system engineers the execution environment of the system.
- Provide validation and illustration for all users to see the design is complete.

1.3 Definitions, Acronyms, and Abbreviations
- **Stakeholder -** A person such as an employee, customer, or citizen who is involved with an organization, society, etc. and therefore has responsibilities towards it and an interest in its success.[1]
- **Architecture View Model (4+1 view model)** − Architecture view model represents the functional and non-functional requirements of software application.[2]
- **MoH** - Ministry of Health

1.4 References

[1] "Cambridge English Dictionary: Meanings & Definitions."
https://dictionary.cambridge.org/dictionary/english/ (accessed Mar. 19, 2020).

[2] "TutorialsPoint: Architecture Models"
https://www.tutorialspoint.com/software_architecture_design/architecture_models.htm
(accessed Mar. 25, 2020).

[3] LucidChart - tool for drawing diagrams
https://www.lucidchart.com/?noHomepageRedirect=true (accessed Mar. 20, 2020)

1.5 Overview

The purpose and scope of this document were explained earlier in this document. The document will continue to explain the system to be implemented in an architectural and design perspective using an architectural view model while also getting into detail about other factors regarding the architecture of the system.

1. Section 2 will briefly describe how the architecture of the system will be presented in the following sections, explaining what views are necessary and what models will be used to explain the said views.
2. Section 3 will carry an explanation about the architectural goals and constraints of the system.
3. Rest of the sections will get into details about each view of the system.

2. Architectural Representation

The document will describe the Use-case, Logical, Process, Deployment and Implementation views of the system using,

- Use-case diagram and use-case realizations for the Use-case view
- Class diagram for Logical view
- Activity and sequence diagrams for Process view
- Deployment diagram for Deployment view
- Package diagram for Implementation view

following to the 4+1 Architecture View Model.
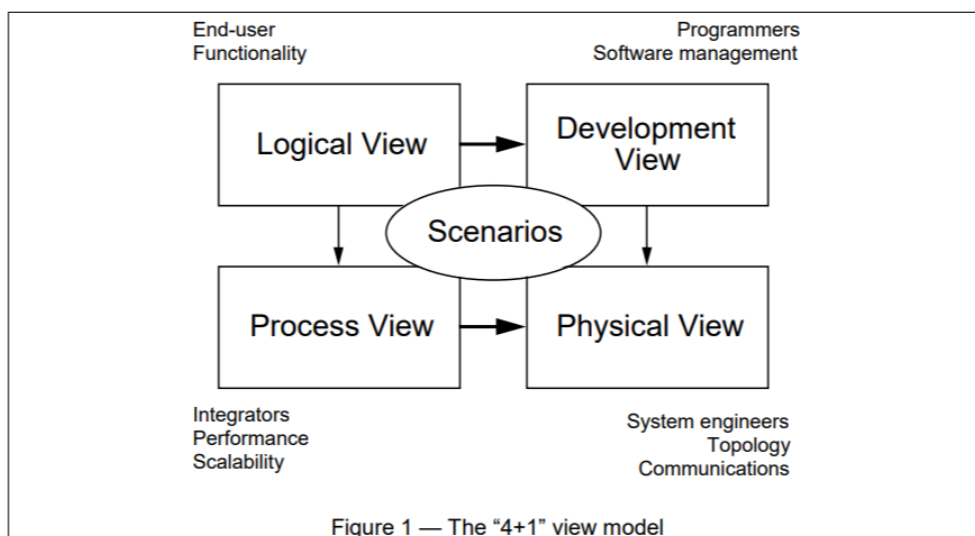
Figure 1 — The "4+1" view model

3. Architectural Goals and Constraints

3.1 Security and Privacy

Sensitive data such as patient records should be secure. Passwords should be encrypted before being saved.

3.2 Persistence

All the submissions would be stored in a SQL database. Further measures would also be taken to maintain ACID properties (atomicity, confidentiality, integrity, durability).

3.3 Reliability / Availability

The system would be rigorously tested before deployment to ensure that it works smoothly. Moreover, once deployed, the system would be available 24/7. The system is in charge of making important decisions when it comes to assigning beds to patients and finding the ideal location for the next hospital. Hence these parts should work with high accuracy so that all users can rely on the system at any time.

3.4 Performance

System performance cannot be commented during the preparation of this document. Rigorous tests would have to be undertaken in order to quantitatively assess the performance of the system.
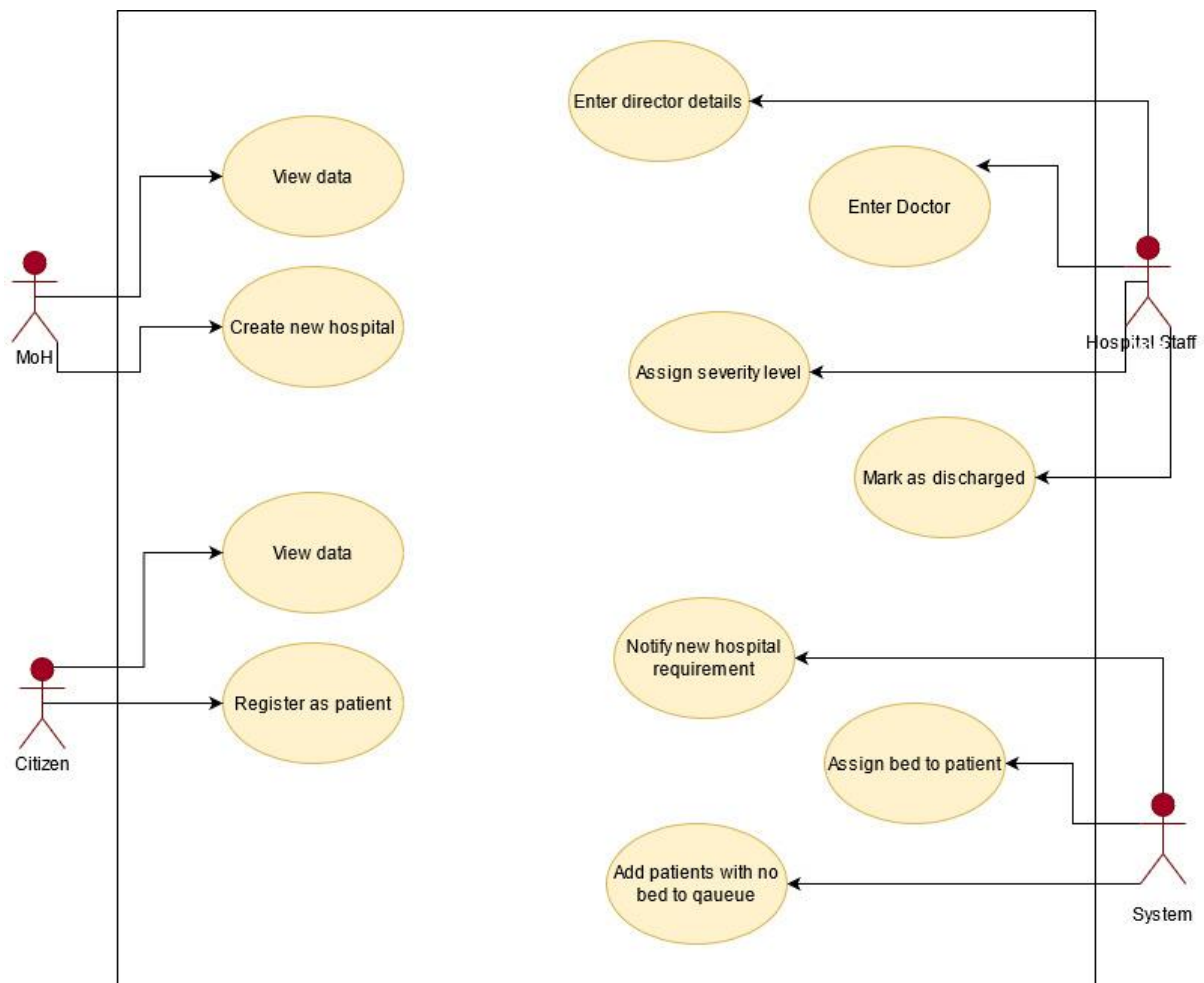
3.8 Development tools

The project incorporates many development tools at various stages.

- Programming : IntelliJ IDE, VScode
- Diagrams : Draw.IO, LucidChart
- Version Controlling: Git

4. Use-Case View

4.1 Use-Case Diagram

4.2 Use-Case Realizations

4.2.1 MoH related use-cases

4.2.1.1 Create New Hospital

Use case name	Create New Hospital
Actor	MoH
Description	When there are more than four patients waiting in the queue for a bed in a hospital, the system will notify the MoH that a new hospital is needed with the details of the location of the new hospital. The MoH, once the hospital is built, will add it to the system.
Preconditions	• Logged into the system. • The system should have requested a new hospital.
Main Flow	1) Click to create a new hospital. 2) Enter details about the new hospital. 3) New hospital created. 4) All waiting patients assigned to the new hospital. 5) Login credentials sent to hospital email.

4.2.2 Hospital Staff related use-cases

4.2.2.1 Add Director Details

Use case name	Add Director Details
Actor	Hospital Staff
Description	Once a new hospital is added to the system and login credentials are sent, the hospital staff can login and add the director of the hospital to the system.
Preconditions	• Logged into the system.

Main Flow	1) Click to add director details. 2) Enter director details. 3) System saves the data.

4.2.2.2 Add New Doctor

Use case name	Add New Doctor
Actor	Hospital Staff
Description	Add details of a doctor working in the hospital.
Preconditions	● Logged in.
Main Flow	1) Clicks to create a new doctor. 2) Enter details of the new doctor. 3) System saves the data.

4.2.2.3 Assign Severity Level to Patient

Use case name	Assign Severity Level to Patient
Actor	Hospital Staff
Description	Once a patient arrives at a hospital, a doctor assesses him and assigns him a severity level. The staff should update this details about the patient.
Preconditions	● Logged in. ● The patient has registered earlier. ● The doctor should be in the system.
Main Flow	1) Select the patient. 2) Select the severity level and click to choose. 3) Enter the assessed doctor. 4) System updates. 5) Patient status changed from registers to admitted.

4.2.2.4 Mark patient as Discharged

Use case name	Mark patient as Discharged
Actor	Hospital Staff
Description	Once a director of a hospital decides a patient can be discharged, this information should be updated in the system by the staff.
Preconditions	- Logged in. - The patient should be in the admitted stage.
Main Flow	1) Select the patient to be discharged. 2) Click the discharge button. 3) Status of patient changed to discharged. 4) The bed is vacant and will be assigned to a waiting patient.

4.2.3 Citizen related use-cases

4.2.2.4 Register as a Patient

Use case name	Register as a Patient
Actor	Citizen
Description	If a citizen is showing symptoms, he can register in the system as a patient. He will be given a serial number and a bed will be assigned if available. If not, he will be put to a waiting queue.
Preconditions	None
Main Flow	1. Click to register. 2. Provide details including coordinates. 3. System saves the new patient with a unique serial number.

|	4. System assigned an available bed to the patient. 5. If no bed is available, the patient will be added to the queue.

5. Logical View

5.1 Overview

This view will be defining the components of the system and showing all interactions and relationships between them. This view is important for the end-user who is interested in the services provided by the system which are converted into functional aspects of the system in this view.

5.2 Architecturally Significant Design Packages

Class Diagram
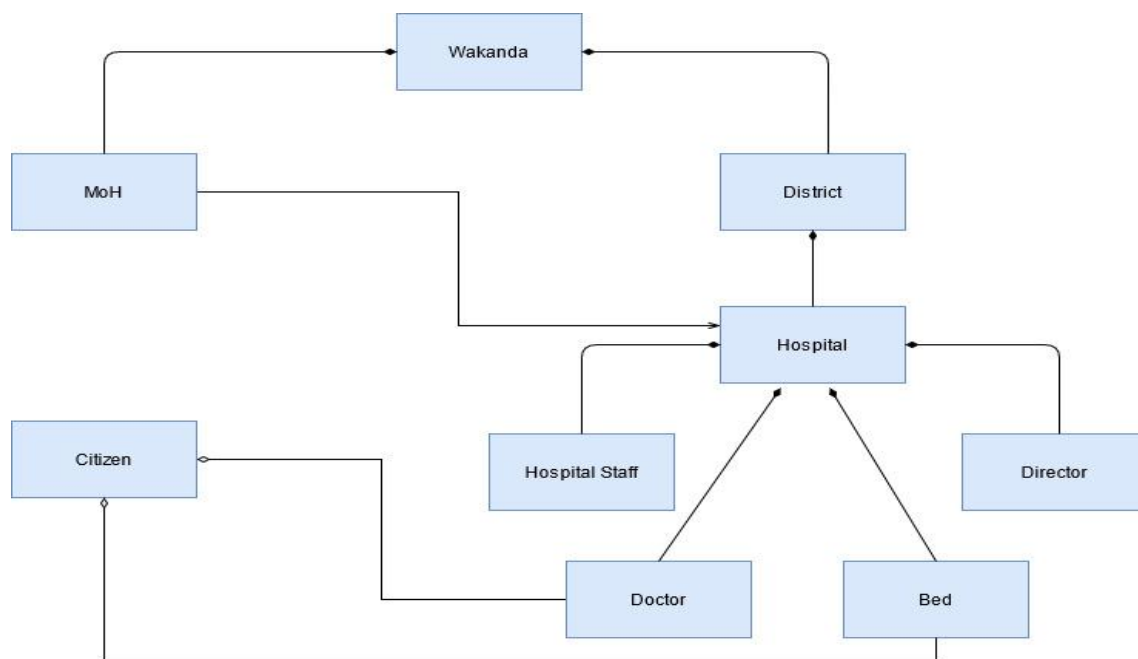
6. Process View

This view gives a clear idea about the processes and workflow rules of the system and how those processes communicate with each other.

6.1 Activity Diagrams

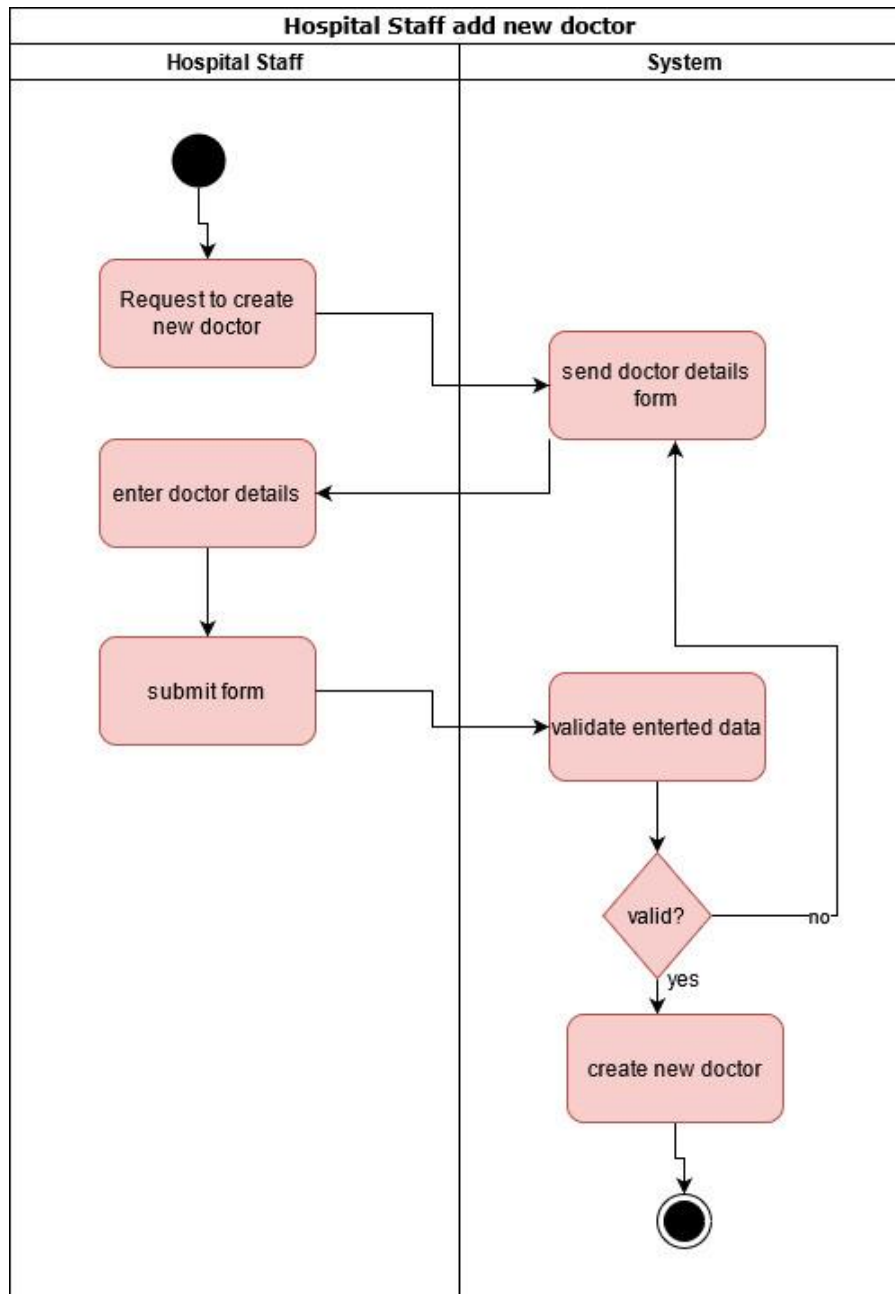### 6.1.1 Hospital staff assign severity level for a patient
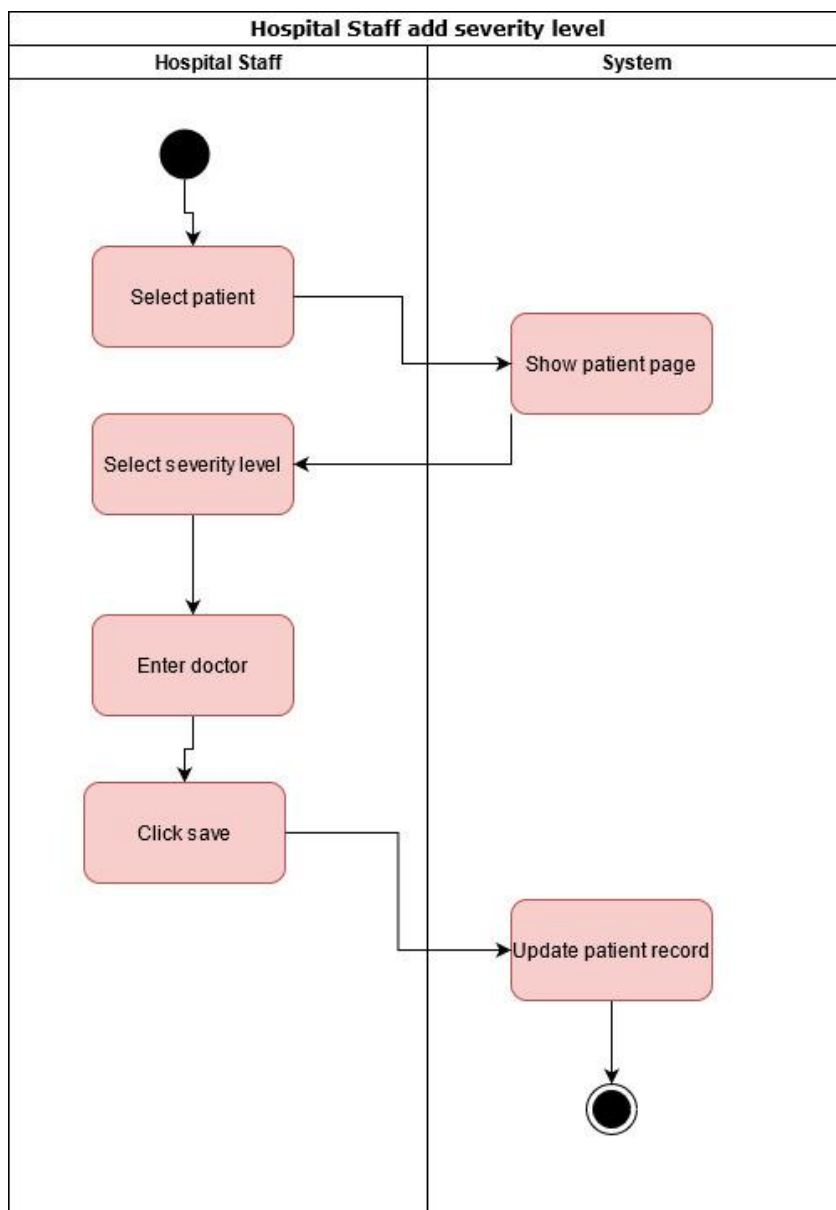
6.1.2 MoH adds new hospital to the system
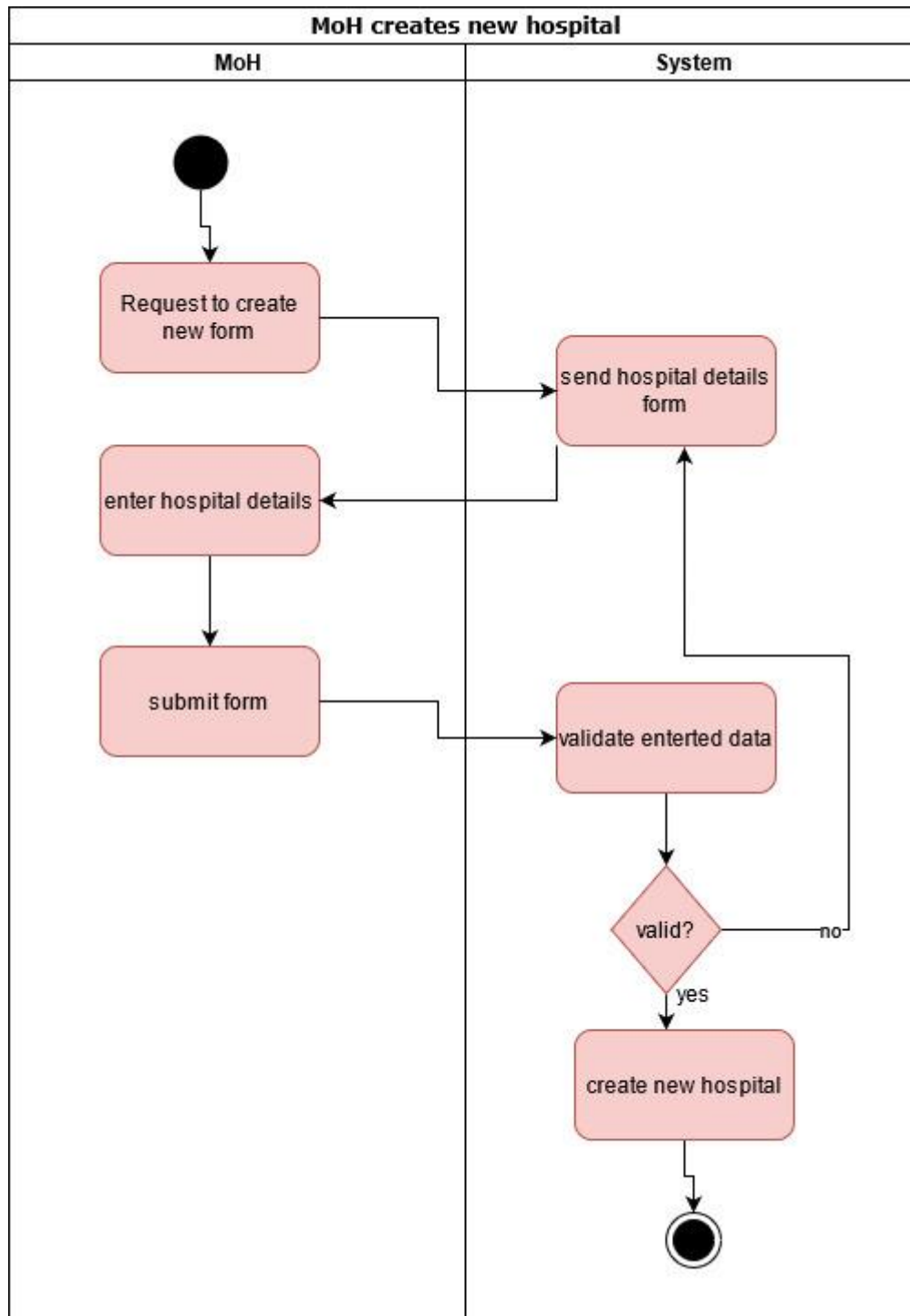
6.1.3 Hospital staff add new doctor

6.2 State diagrams

6.2.1 State diagram for patient
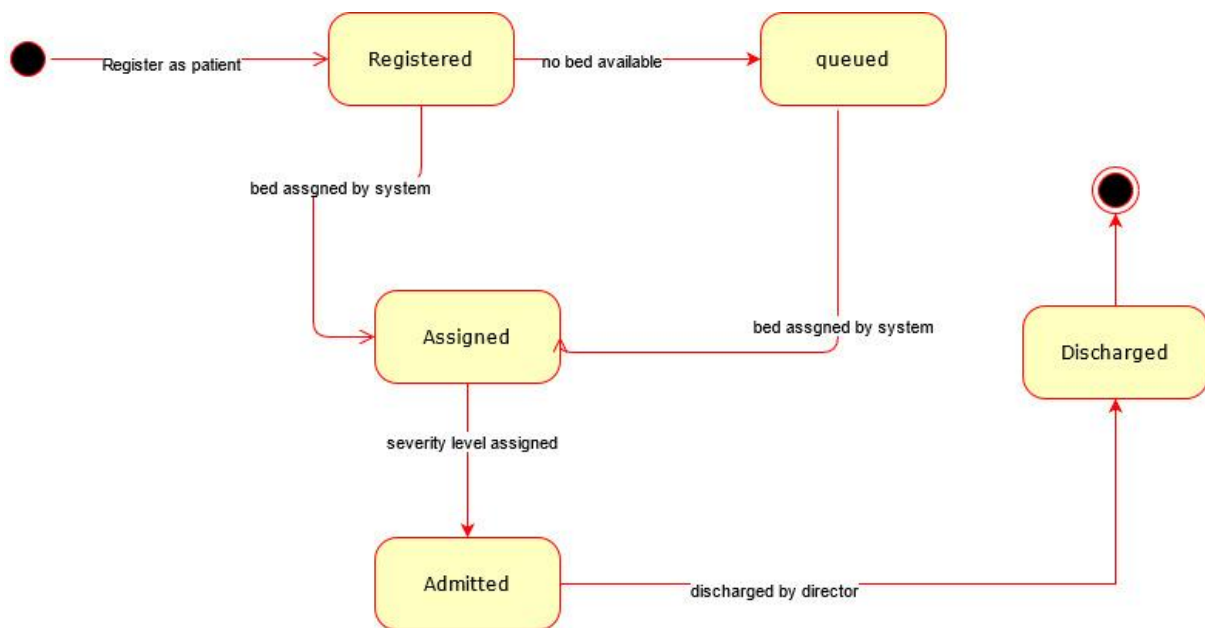
6.3 Sequence Diagrams

6.3.1 Patient registration and bed allocation
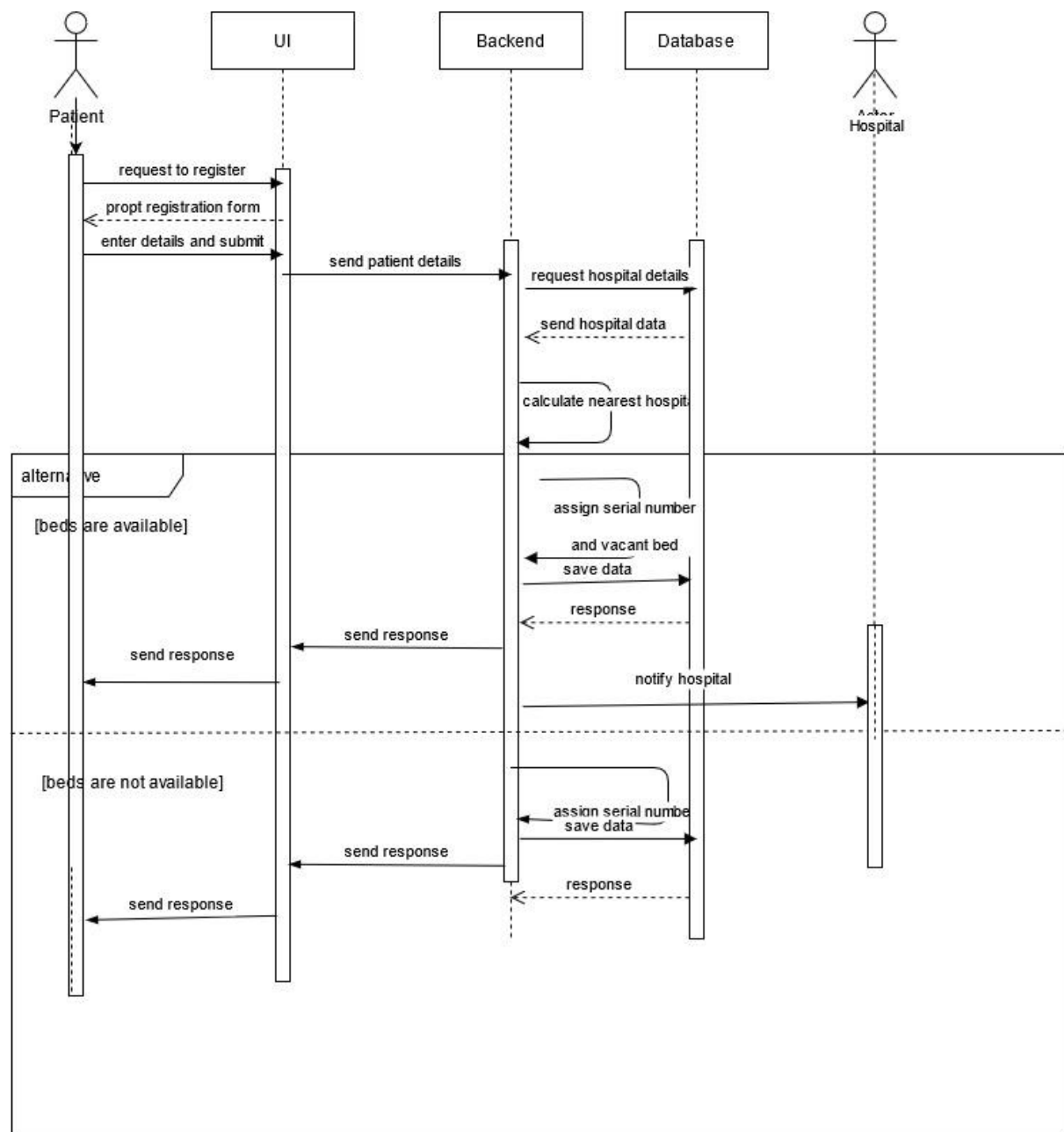
7. Deployment View

This view gives an insight into the execution environment of the system.

Deployment Diagram

The system will have a Sprint Boot backend and a React frontend. The database will be on a database server with MYSQL as the DBMS.
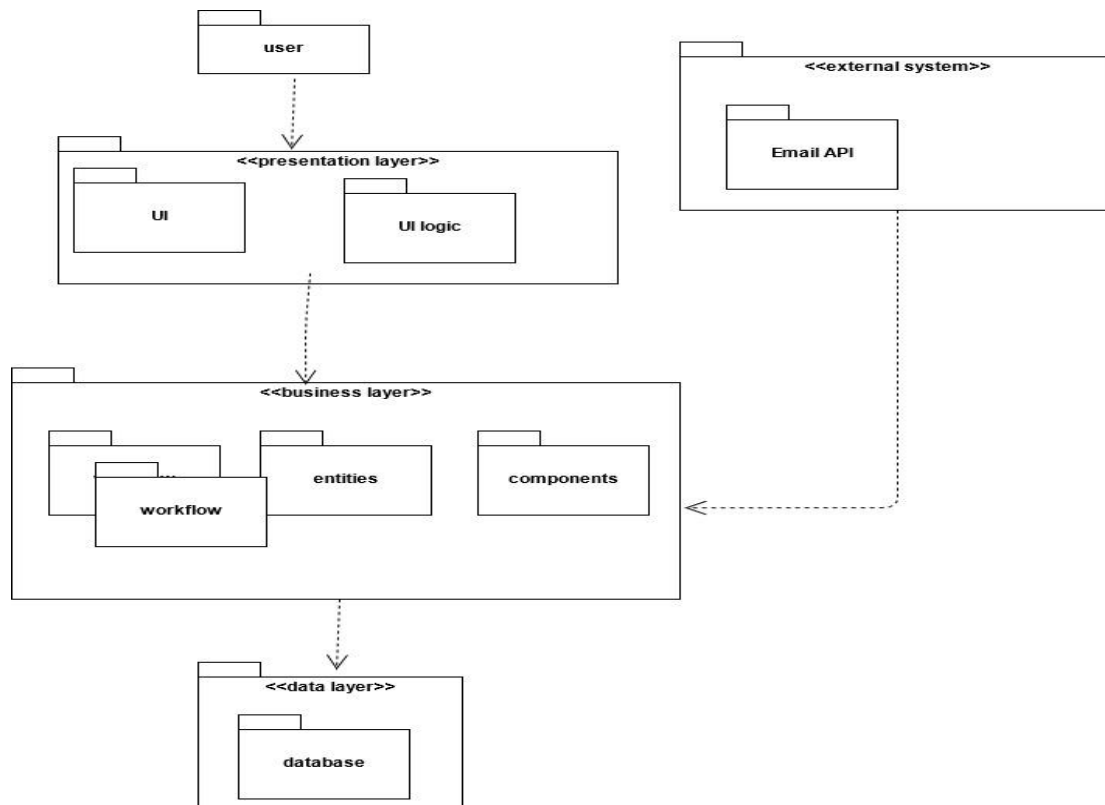
8. Implementation View

8.1 Overview

This view lets the programmers and software managers have a clear view of the hierarchy of the system, packages used and sub-systems utilized.

8.2 Layers

<u>Package Diagram</u>

9. Data View

Data about doctors, and hospitals and patient records are directly transferred to the database. The translation between the Design Model and the Data Model is trivial.