National COVID Management System

Software Architecture Document

Version 1.0

Revision History

Date	Version	Description	Author
18/04/21	1.0	Initial Software architecture document	K.A.S.H. Kumarasinghe

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

This document provides a comprehensive architectural 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 decisions which have been made on the system.

1.2 Scope

This document represents the overall architecture of the National COVID Management System (NCMS). It describes the various aspects of the mentioned system's design that are architecturally significant.

1.3 Definitions, Acronyms, and Abbreviations

SRS	Software Requirements Specification
NCMS	National COVID Management System
MoH	Ministry of Health
COVID-19	CoronaVirus Disease 2019 caused by SARS-CoV-2
REST	REpresentational State Transfer
API	Application Programming Interface
HTTP	HyperText Transfer Protocol
UML	Unified Modeling Language
UI	User Interface

1.4 References

[1] The "4+1" view model of software architecture, Philippe Kruchten, November 1995,

http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf

[2] Diagrams.net https://app.diagrams.net/ (Tool used to draw UML Diagrams)

1.5 Overview

This document provides an architectural overview of the NCMS using different architectural views such as logical, development, process and physical views. An overall view of the system can be visualized by referring to the content of each section of the document. Finally the document describes the data view, size and performance, and the quality of the NCMS.

2. Architectural Representation

The architecture of the NCMS will be described using the 4+1 view architecture mode.

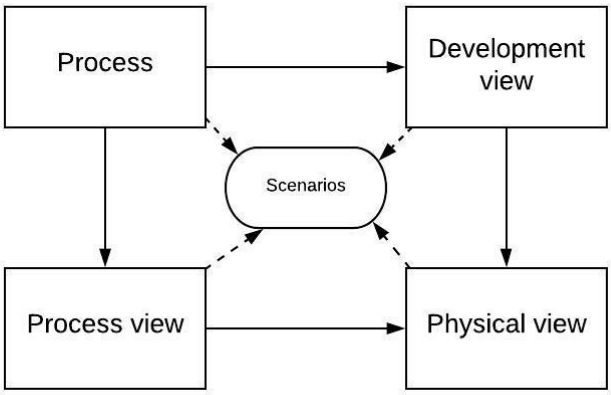

Figure 1 - 4+1 architecture view model

- **Scenarios**
  Scenarios which is also known as the Use case view captures the system's functionality which is associated with each user class. In the document, this view is represented using a UML use case diagram along with a detailed description for each use case. All the stakeholders can refer to this section of the document to get a comprehensive understanding of the functionality of the system.

- **Logical view**

Logical view describes the architecturally significant parts in the design of the system. In this document, this view is represented using a UML class diagram along with a description about each class. This view is mainly intended for system designers and developers.

- **Process view**
  Process view is concerned about the dynamic behaviour of the system. It focuses on the flow of system processes and on how the logical components interact. In this document, this view is represented using UML activity diagrams and UML sequence diagrams. This view is intended for system designers and developers.

- **Physical view**
  The physical view describes the physical network configurations and execution environments on which the system is deployed and run. This view is represented using UML deployment diagram. This view is mainly intended for deployment managers.

- **Implementation view**
  The implementation view illustrates a system from a programmer's perspective and it describes how the software components are structured in the system. In this document, this view is represented using the UML package diagram. This view is mainly intended for system developers.

3. Architectural Goals and Constraints

- **Privacy**
  The system will contain critical data such as passwords. Hashing methods should be used to build the privacy of the system

- **Portability**
  Application must be used on any browser, not restricted to any particular category of browser or device.

- **Scalability**
  System should be designed to handle a large number of concurrent usage. System should avoid the single point of failure.

4. Use-Case View

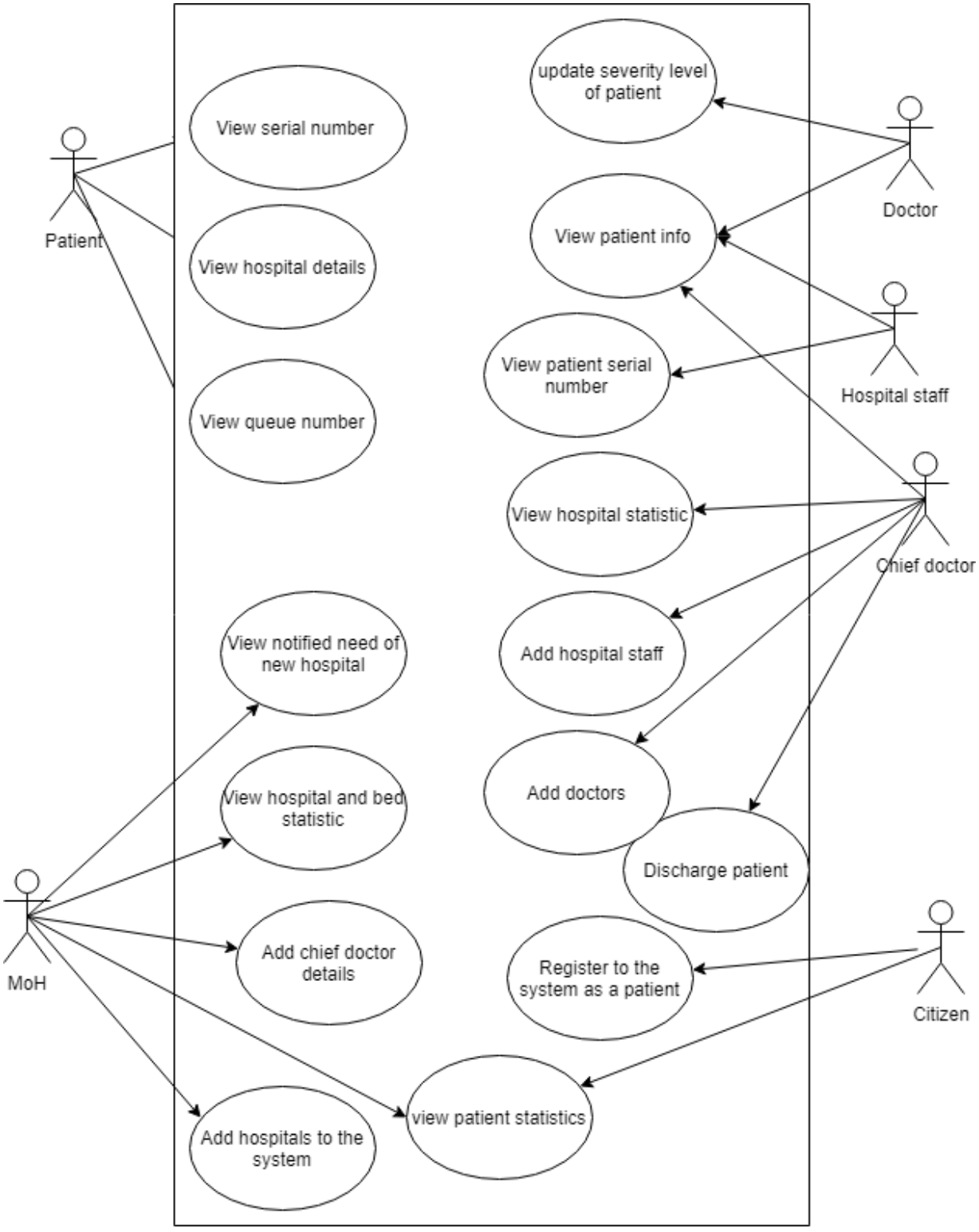

Figure 2 - Use case diagram

4.1 Use-Case Realizations

4.1.1 Use cases related to citizen

4.1.1.1 View patient statistics

Use case name	View patient statistics
Actor	Citizen
Description	Citizens should be able to view the patient statistics when they select the level (country, district, hospital) and the select the type as daily or overall.
Preconditions	No pre conditions should be satisfied. It is not needed for the user to be registered with the system to perform this use case.
Main flow	In the home page, the citizen selects from the level dropdown the level as country,district or hospital, the destination, and selects type as daily or overall.
Successful end/post condition	Total cases, total deaths and the total recovered based on the level and type that selected will be displayed on the screen.
Fail end/post condition	N/A
Extensions	N/A

4.1.2 Use cases related to MoH authority

4.1.2.1 View patient statistics

Use case name	View patient statistics

Actor	MoH authority
Description	MoH authorities should be able to view the patient statistics when they select the level (country, district, hospital) and the select the type as daily or overall.
Preconditions	No precondition
Main flow	In the home page, the MoH authority selects from the level dropdown the level as country,district or hospital, the destination, and selects type as daily or overall.
Successful end/post condition	Total cases, total deaths and the total recovered based on the level and type that selected will be displayed on the screen.
Fail end/post condition	N/A
Extensions	N/A

4.1.2.2 View hospital and bed statistic

Use case name	View hospital and bed statistic
Actor	MoH authority
Description	MoH authorities should be able to view the hospital and bed statistics when they select the hospital.
Preconditions	No precondition
Main flow	In the statistics page, the MoH authority selects from the hospital dropdown. .
Successful end/post condition	Overall and daily total cases, total deaths and total deaths of the selected hospital and the available bed details, queue details and the chief doctor’s details will be displayed.

Fail end/post condition	N/A
Extensions	N/A

4.1.2.3 Add hospital to the system

Use case name	Add hospital to the system
Actor	MoH authority
Description	MoH authorities should be able to add hospital to the system.
Preconditions	No precondition
Main flow	1. MoH authority navigate to the 'add hospital' section from the navigation bar 2. MoH authority inputs following data ○ Hospital name ○ Hospital coordinates ○ Hospital district ○ Hospital contact number ○ Chief doctor name ○ Chief doctor's contact number ○ Chief doctor's email ○ Chief doctor;s NIC
Successful end/post condition	Submitted hospital will be displayed on the hospital statistic section
Fail end/post condition	Submitted hospital will not be displayed on the hospital statistic section
Extensions	N/A

4.1.2.4 Add a chief doctor to the system

Use case name	Add a chief doctor to the system
Actor	MoH authority
Description	MoH authorities should be able to add chief doctors to the system.
Preconditions	The doctor's hospital must be exist in the system
Main flow	1. MoH authority navigate to the 'add doctor' section from the navigation bar 2. MoH authority inputs following data ○ Chief doctor name ○ Chief doctor's hospital ○ Chief doctor's contact number ○ Chief doctor's email ○ Chief doctor;s NIC
Successful end/post condition	Submitted chief doctor will be displayed under his/her hospital of the hospital statistic section
Fail end/post condition	Submitted chief doctor will not be displayed under his/her hospital of the hospital statistic section
Extensions	N/A

4.1.3 Use cases related to the chief doctor

4.1.3.1 Add a doctor to the system

Use case name	Add doctor to the system
Actor	Chief doctor

Description	Chief doctors should be able to add doctors to the system.
Preconditions	N/A
Main flow	1. Chief doctor navigate to the 'add doctor' section from the navigation bar 2. Chief doctor inputs following data ○ doctor's name ○ doctor's contact number ○ doctor's email ○ doctor;s NIC
Successful end/post condition	Success message will be displayed
Fail end/post condition	An error message will be displayed.
Extensions	N/A

4.1.3.2 Add a hospital staff member to the system

Use case name	Add a hospital staff member to the system
Actor	Chief doctor
Description	Chief doctors should be able to add hospital staff to the system.
Preconditions	N/A
Main flow	1. Chief doctor navigate to the 'add staff' section from the navigation bar 2. Chief doctor inputs following data ○ staff member's name ○ staff member's contact number ○ staff member's email ○ staff member's NIC
Successful end/post condition	Success message will be displayed

Fail end/post condition	An error message will be displayed.
Extensions	N/A

4.1.3.3 View hospital statistic

Use case name	View hospital statistic
Actor	Chief doctor
Description	Chief doctor should be able to view the hospital statistics of his/her hospital
Preconditions	No precondition
Main flow	Chief doctor navigates to the statistics page
Successful end/post condition	Overall and daily total cases, total deaths and total deaths of the hospital and the available bed details will be displayed.
Fail end/post condition	N/A
Extensions	N/A

4.1.3.4 Discharge a patient

Use case name	Discharge a patient
Actor	Chief doctor
Description	Chief doctor should be able to discharge the patient from the hospital
Preconditions	No precondition

Main flow	1. Chief doctor navigates to the patient page 2. Search the patient that want to discharge 3. Click discharge button
Successful end/post condition	Success message will display
Fail end/post condition	Success message will not display
Extensions	N/A

5. Logical View

5.1 Overview

The system will be designed on model-view-controller architecture.

- **Model**
  - an object which carries data.
  - can have logic to update the controller if its data changes.
- **View**
  - the visualization of the data that model contains.
- **Controller**
  - acts on both model and view.
  - controls the data flow into the model object and updates the view whenever data changes.
  - keeps view and model separate.

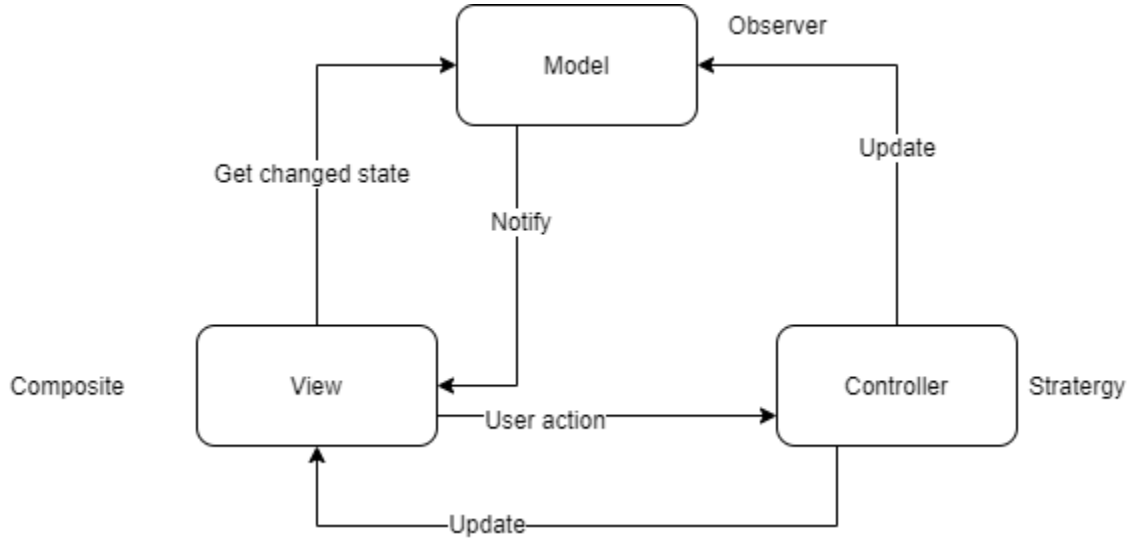

Figure 3 - MVC architecture

5.2 Architecturally Significant Design Packages

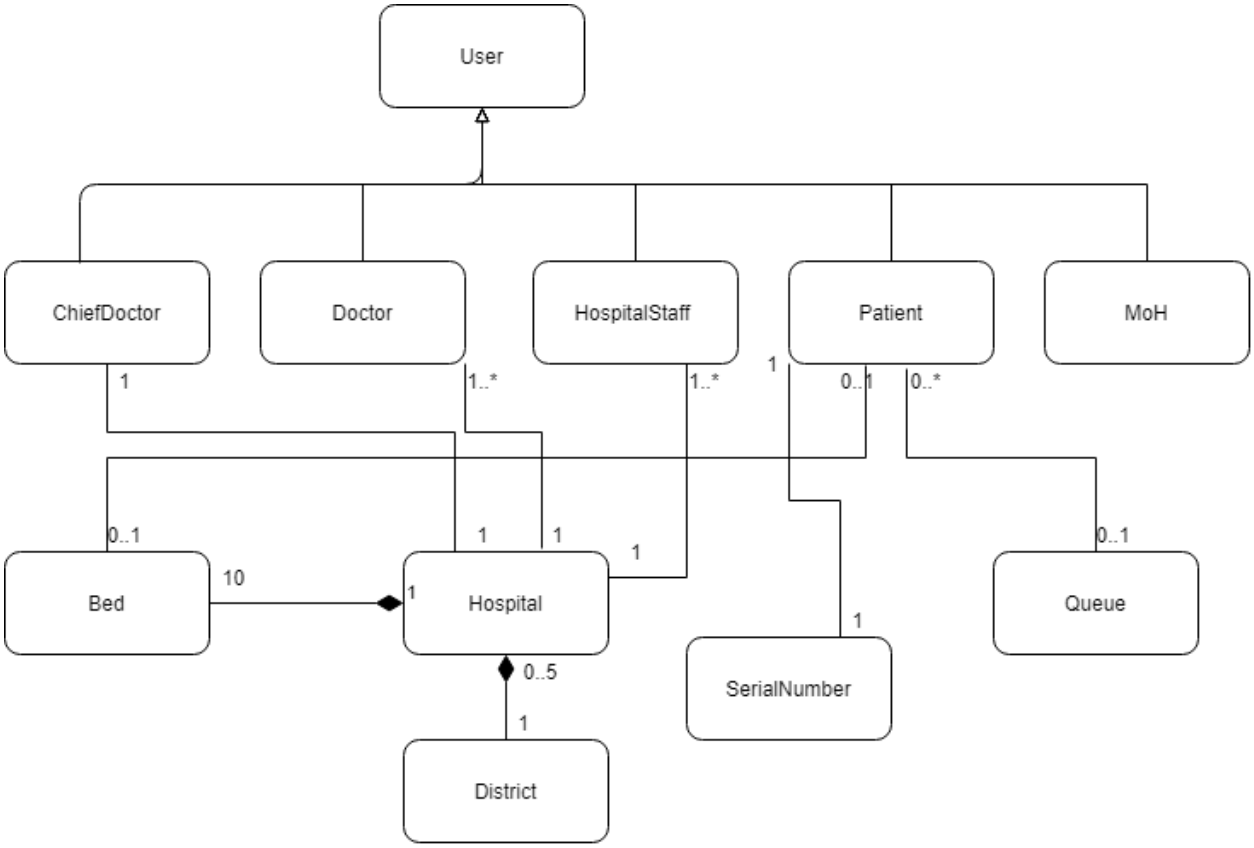

Figure 4 - class diagram

Description of the important classes of the NCMS.

- User
  User class is an abstract class, the superclass of the doctor, chief doctor, hospital staff, patient and MoH.The abstract class has attributes of firstName, second Name, email, NIC and the phone number.

- District
  District class has the attributes of name and the coordinates

- Hospital
  Hospital class has the attributes of name, coordinates and district details.

- Serial number
  Handles the serial number generating functionalities.

- Queue
  Contains queue of patients

6. Process View

6.1 Activity diagrams

6.1.1 Chief doctor add a doctor to the system

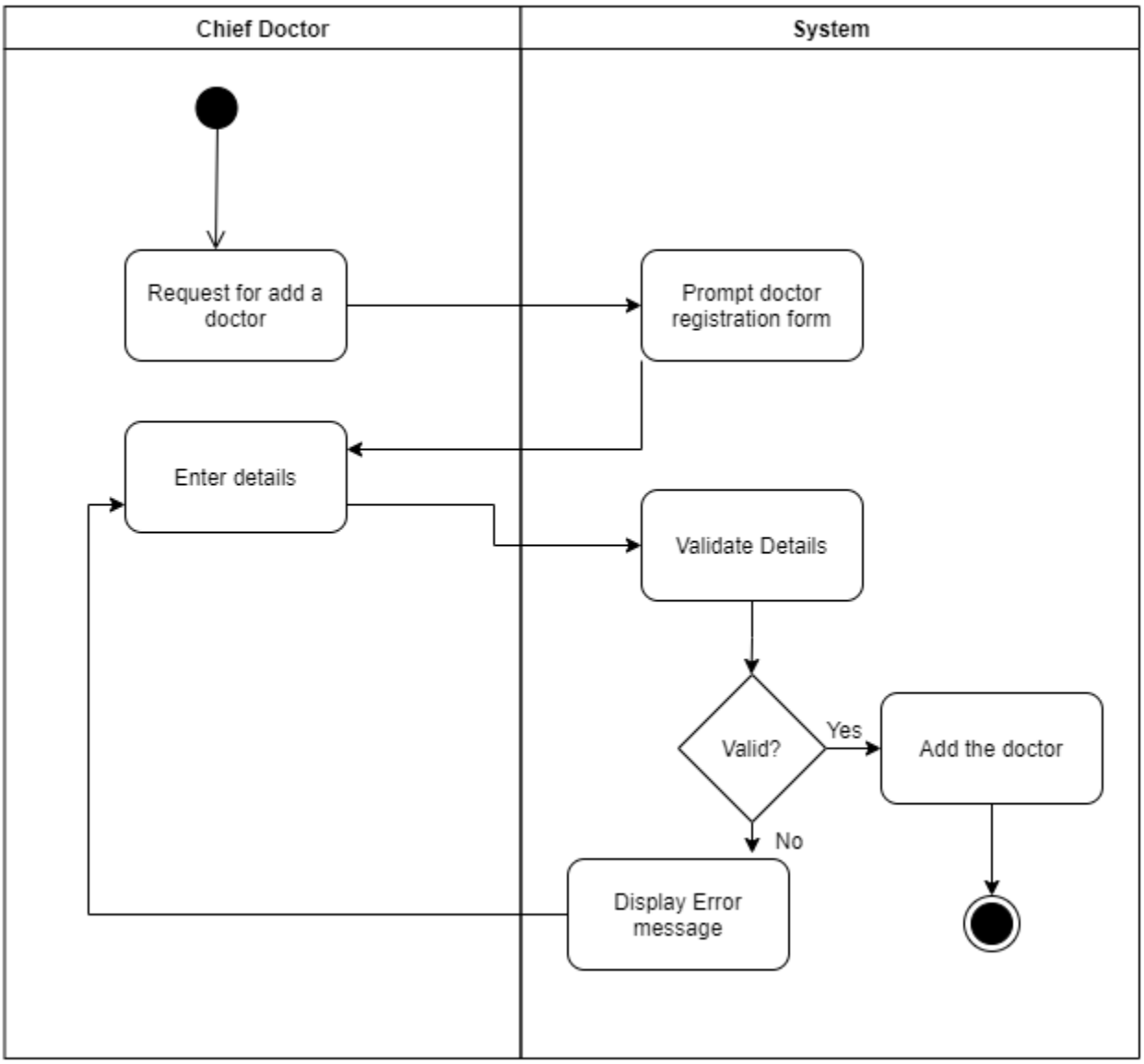

Figure 5 - activity diagram for a chief doctor add a doctor to the system

6.1.2 MoH authorities add a new hospital to the system

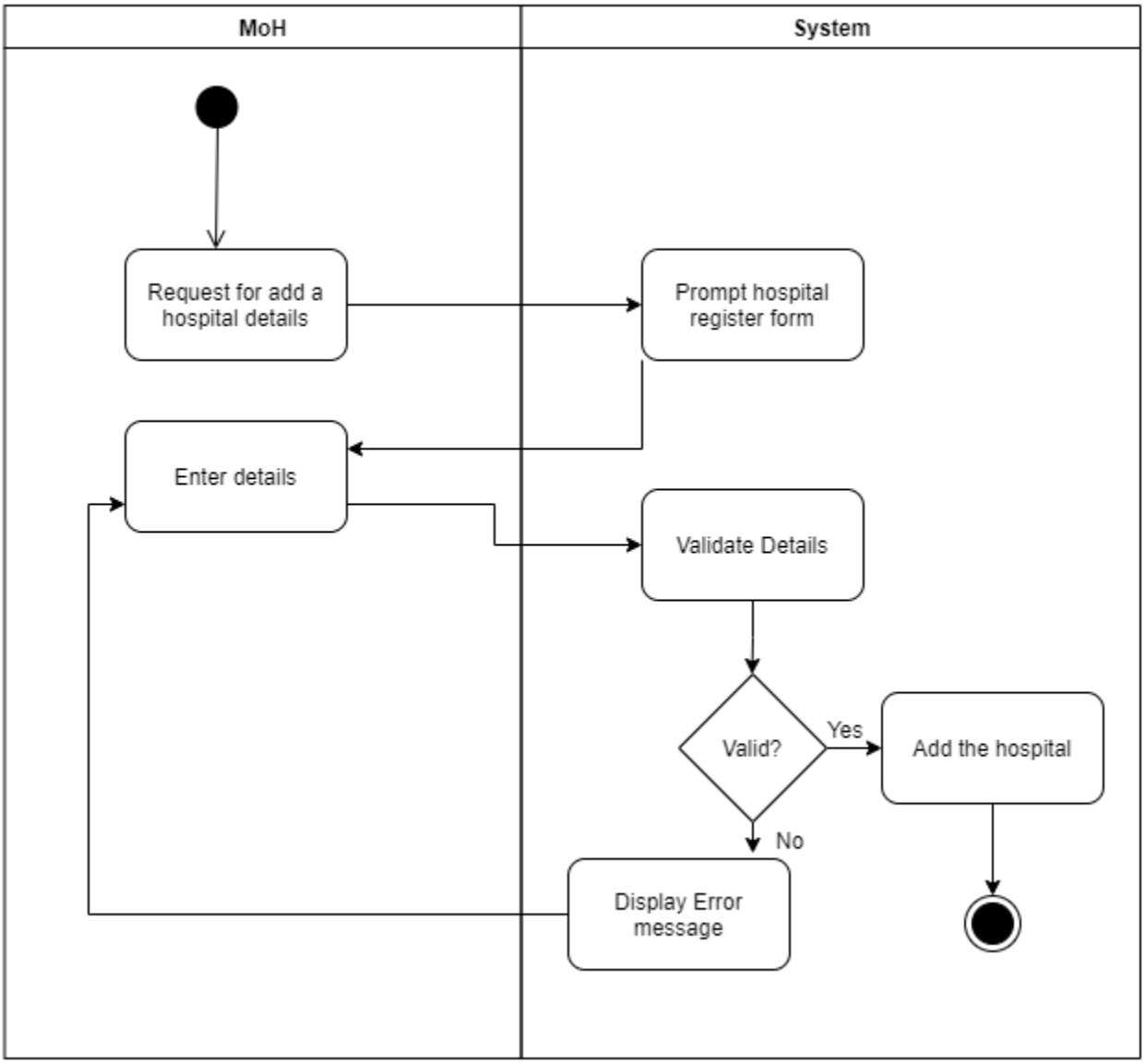

Figure 6 - Activity diagram for the MoH authorities add a new hospital to the system

6.1.3 Citizen register to the system as a patient

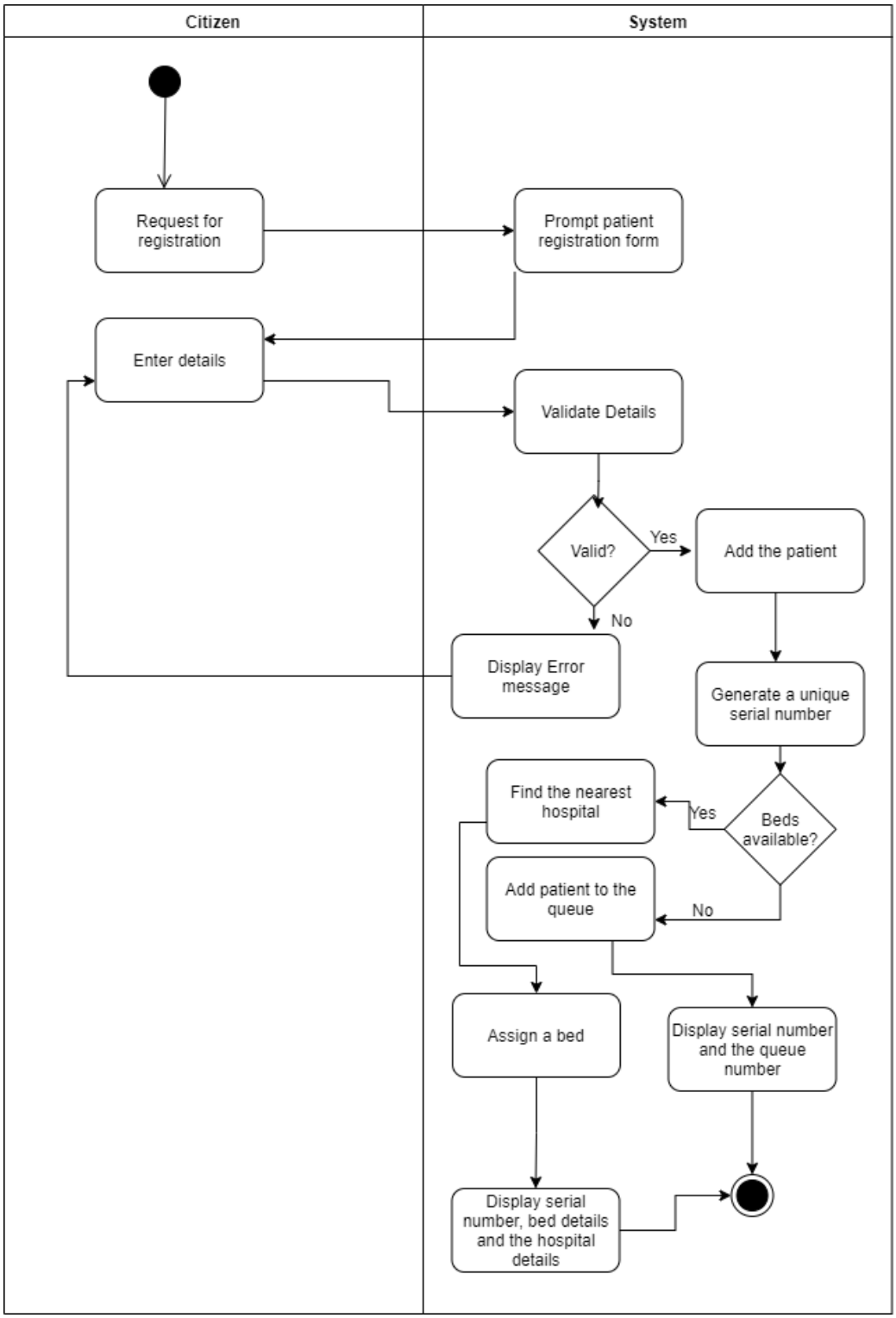

Figure 7 - Activity diagram for the citizen register to the system as a patient

6.1.4 Citizen view patient statistic

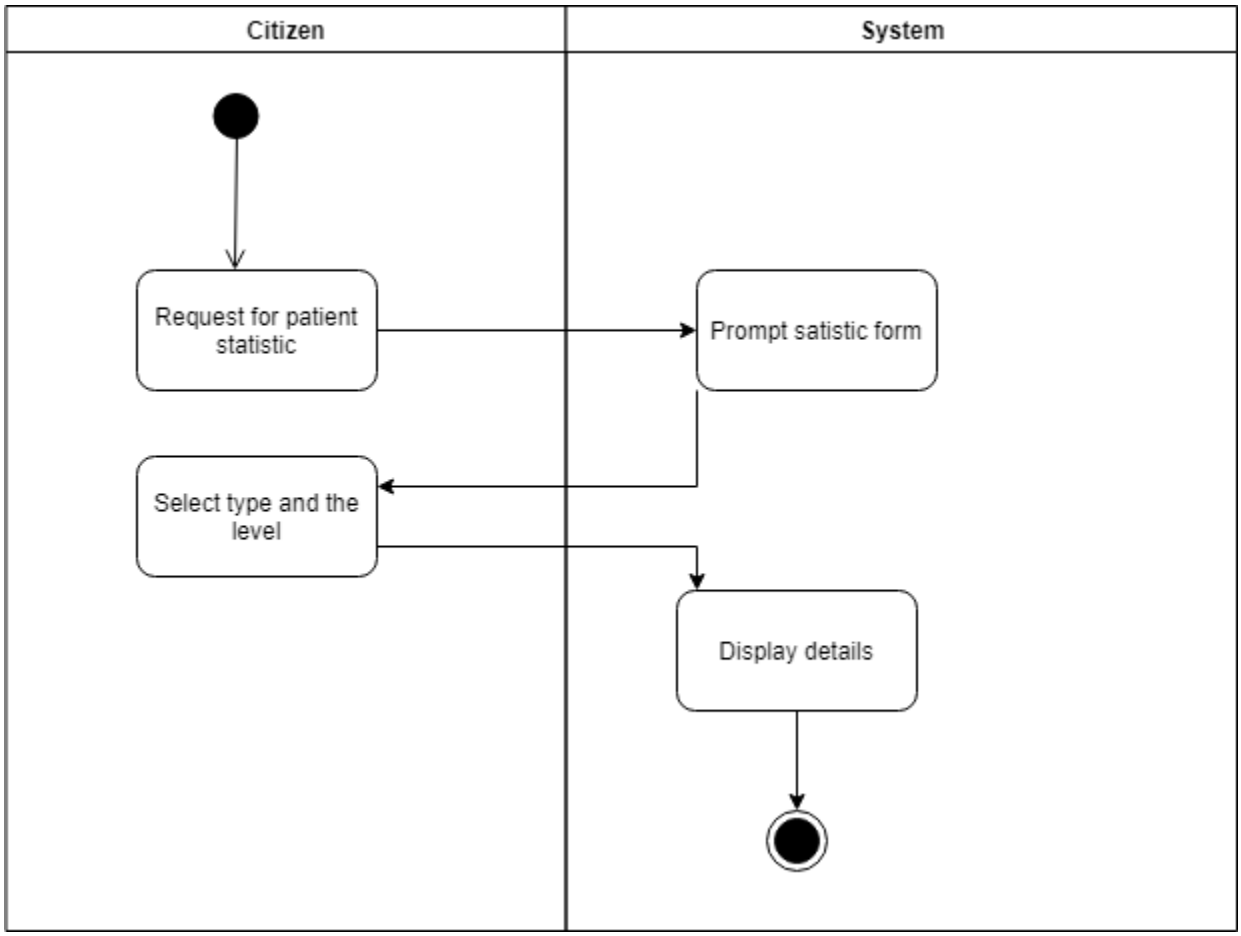

Figure 8 - Activity diagram for the citizen view patient statistic

6.2 Sequence diagram

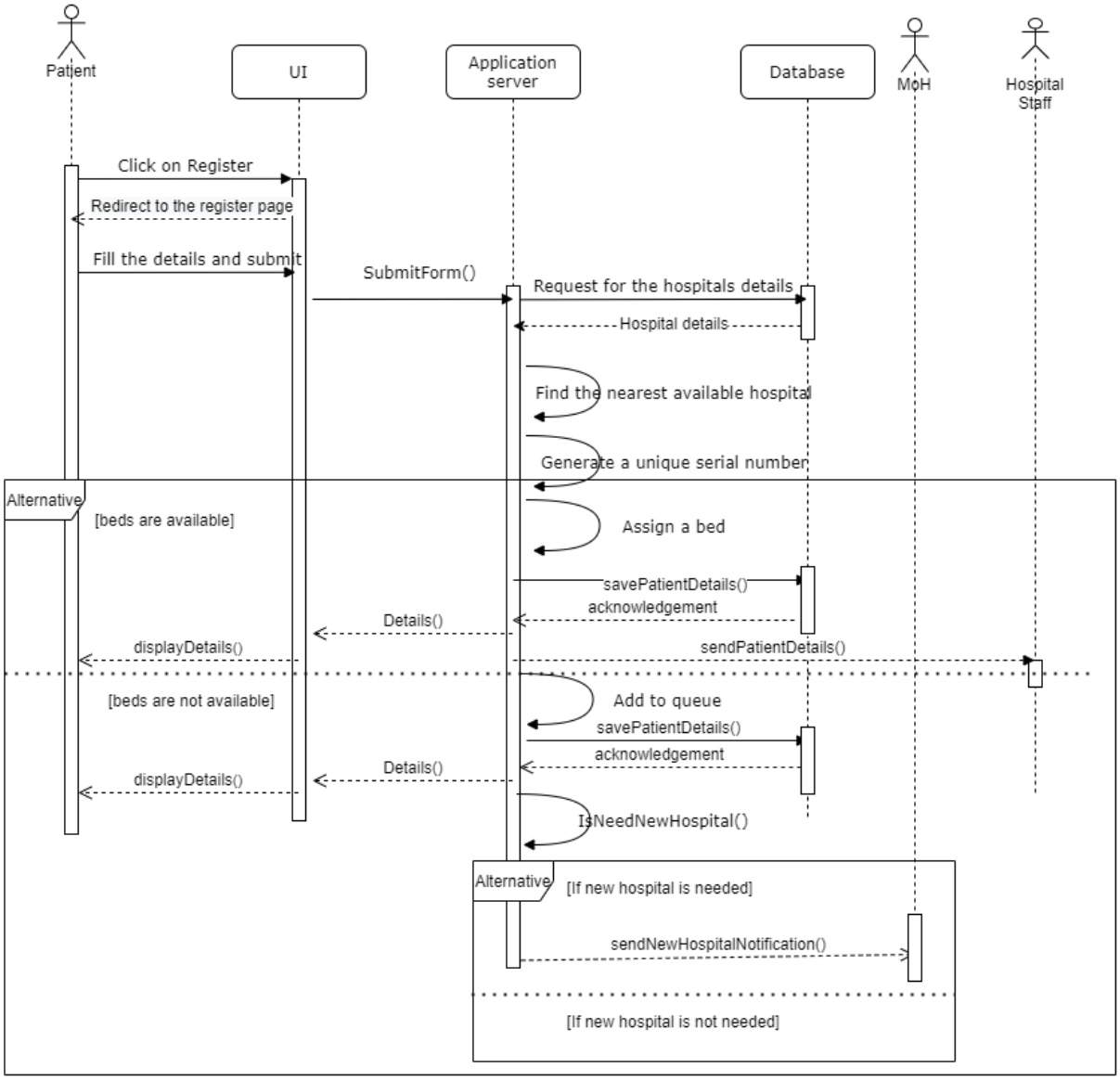

Figure 8 - Sequence diagram

7. Deployment View

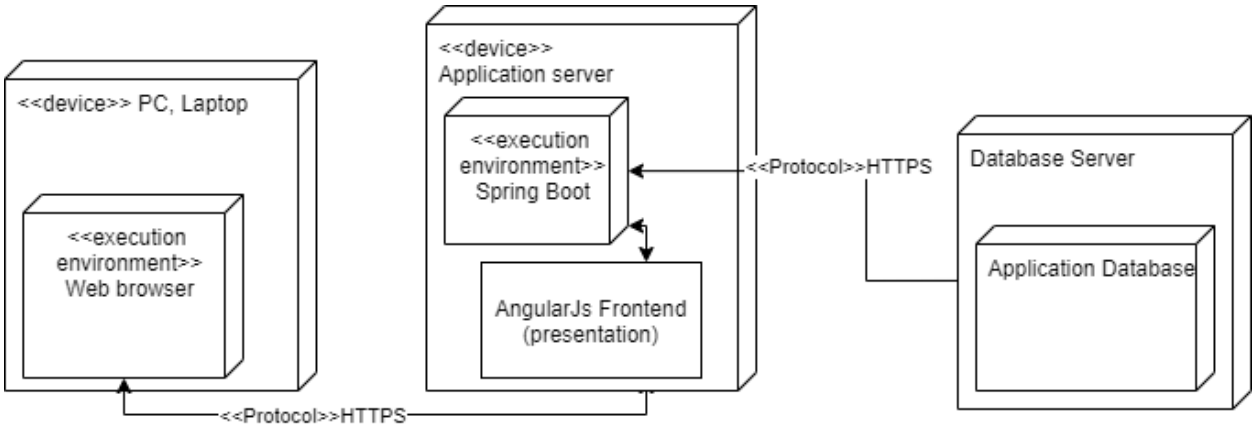

Figure 9 - Deployment diagram

The users can use the service by accessing the application server through the AngularJS Frontend of the web application using the Web browser of his personal computer or laptop. Application server communicates with the MySQL Database to do database operations.

8. Implementation View

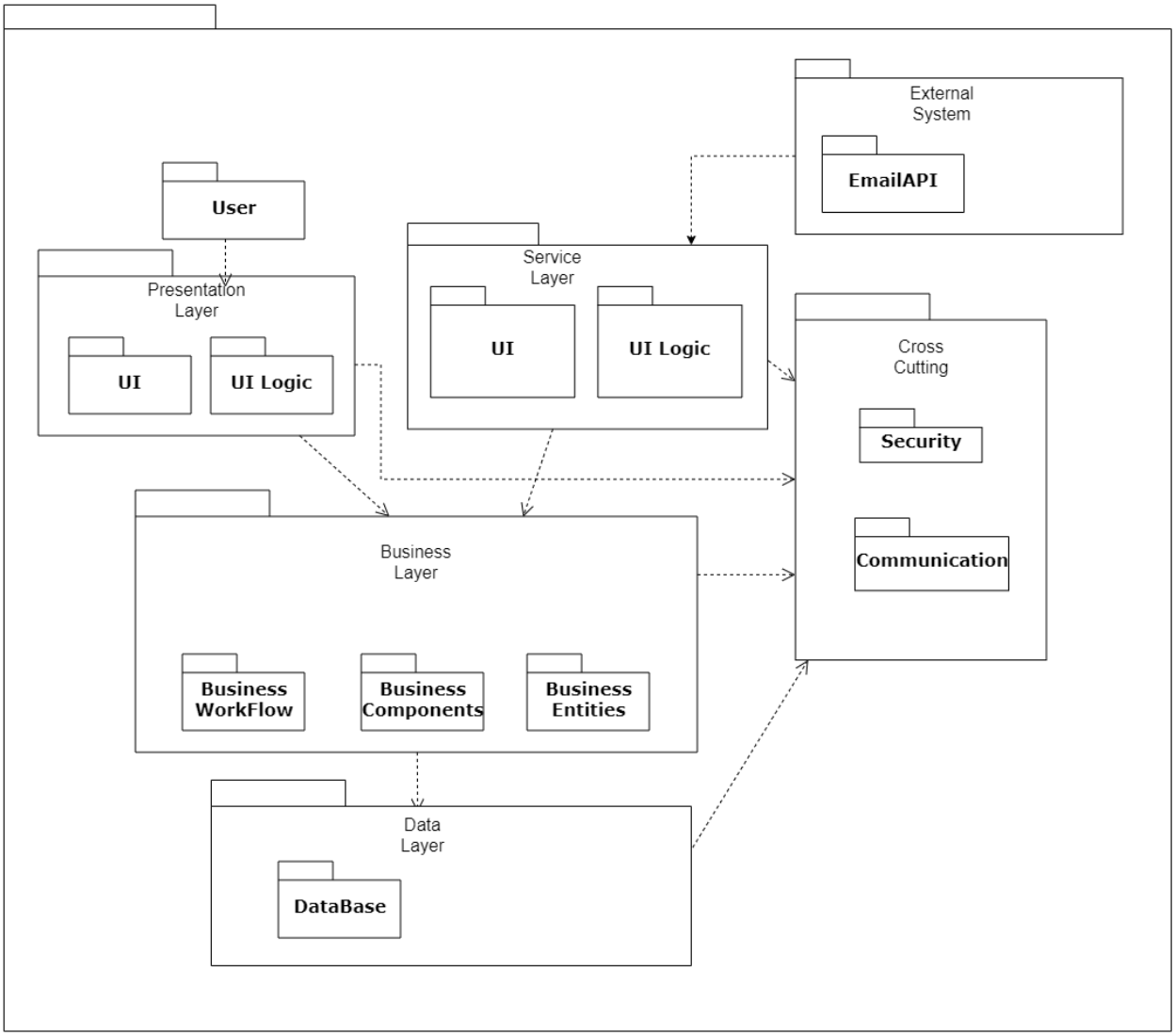

Figure 10 - Package diagram

9. Quality

9.1 Portability

The NCMS web application should be able to run with any browser on any device, not

restricted to any particular category of browser or device.

9.2 Scalability

System should be designed to handle a large number of concurrent usage. System should avoid the single point of failure.

9.3 Privacy

Since the system contains the patient details and the passwords security is important. For that encryption, hashing method will be used.

10. References

[1] The “4+1” view model of software architecture, Philippe Kruchten, November 1995, http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf

[2] Diagrams.net https://app.diagrams.net/ (Tool used to draw UML Diagrams)