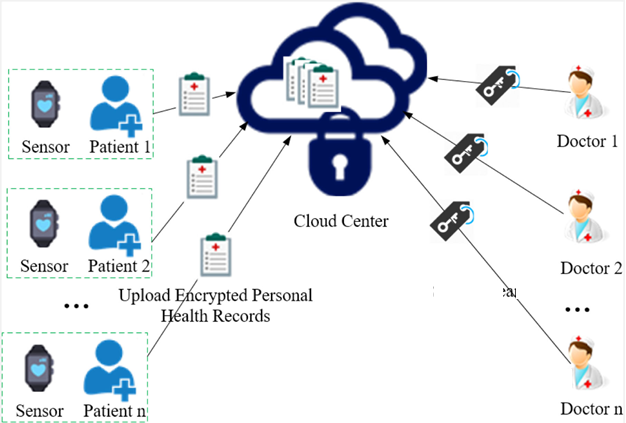
**SYSTEM DESIGN**

**SYSTEM ARCHITECTURE:**



**DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

**Patient:**

Patient

Check

Unauthorized user

Yes

No

Upload Blood Group, Temperature, BP, Pulse

View Uploaded Encrypted Medical record

View Original Medical record

End process

**Doctor (doctor-in-charge):**

Doctor

Yes

No

Check

Unauthorized user

View Assigned Patient details

Decrypt the patient report

Assign doctor-in-agent

Search for patient by ID, Name, Number

View Shared patient details

End process

**Doctor (doctor-in-agent):**

Doctor

Yes

No

Check

Unauthorized user

View Assigned Patient details

Decrypt the patient report

Search for patient

Decrypt the shared patient details

View Shared patient details

End process

**Cloud:**

No

Yes

Unauthorized user

Cloud Login

Check

Approve Patients & Doctors

View Patient & Doctors

Assign Doctors to the patient

End process

Approve the doctor request for authorization

**UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Patient

Doctor-in-agent

doctor-in-charge

Cloud

**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

Doctor-in-charge

Patient

Data User

Data Owner

Login

Login

Login

Login

View assigned Patient ()

Search Patient ()

Request for Patient Access ()

Assign doctor ()

Verify shared patient details ()

View Decrypted Patient Details ()

Upload PHR ()

View encrypted PHR()

View Original PHR ()

View Allowed Files ()

Request for File Access ()

Get Secret Key ()

Verify Secret Key ()

Download / Read File ()

Upload Files ()

Set File Access Policy ()

View Uploaded Files ()

View File Access Report ()

Cloud

Doctor-in-agent

Login

Login

View Cloud Files ()

View Patient Details ()

View Doctor Details ()

View Analysis ()

View Patient Details ()

View Doctor Details

Approve Doctor ()

Approve File Requests ()

**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

SERVICE

Doctor-in-agent

Cloud

Doctor

Patient

Upload PHR

View Patient assigned

View Uploaded PHR

Request for File Access

Search Patient

Approve File Request

View Shared Patient

Assign doctor

View Decrypted Patient Details

View Patient / Doctor Details

View Cloud Files

View Analysis

DATA BASE

**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Start

Doctor-in-agent

Cloud

View PHR requests, Approve PHR Request, View Patient Details,

View assigned Patient, Search patient, Request for doctor, View shared patient, Decrypt patient details, View the patient details.

View Cloud files, View patient details, View doctor details, View graph.

LOGIN

LOGIN

LOGIN

Doctor

Upload PHR, View uploaded encrypted PHR, View Decrypted PHR.

LOGIN

Patient