

INDEX

TOPICS

Page No's

- **Certificates**
- **Acknowledgement**
- **Abstract**
- **Figures/Tables**

CHAPTER-1: INTRODUCTION

CHAPTER-2: LITERATURE SURVEY

CHAPTER-3: SYSTEM ANALYSIS

3.1 Existing System

3.2 Proposed System

CHAPTER-4: SYSTEM REQUIREMENTS

4.1 Hardware Requirements

4.2 Software Requirements

CHAPTER-5: SYSTEM STUDY

5.1 Feasibility Study

5.2 Feasibility Analysis

CHAPTER-6: SYSTEM ARCHITECTURE

6.1 Data Flow Diagram

6.2 UML Diagrams

CHAPTER-7: INPUT AND OUTPUT DESIGN

7.1 Input Design

7.2 Output Design

16 CHAPTER-8: IMPLEMENTATION

8.1 MODULES

8.1.1 Module Description

CHAPTER-9: SOFTWARE ENVIRONMENT

9.1 JAVA

9.2 Source Code

CHAPTER-10: RESULTS/DISCUSSIONS

10.1 System Test

10.2 Output Screens

CHAPTER-11: CONCLUSION

CHAPTER-12: REFERENCES/BIBLIOGRAPHY

LIST OF FIGURES

S.NO	TABLES/FIGURES	PAGE NO'S
6	6.1 System Architecture	
	6.2 UML Diagrams	
	6.2.1 Use Case Diagram	
	6.2.2 Class Diagram	
	6.2.3 Sequence Diagram	
	6.2.4 Collaboration Diagram	
	6.2.5 Activity Diagram	
	6.2.6 Component Diagram	
	6.2.7 Deployment Diagram	
	6.2.8 ER Diagram	
	6.2.9 Data Dictionary	
9	9.2 JAVA Source code with compilation	
10	10.2 Screenshots	

PROTECTING PERSONAL HEALTHCARE RECORDS WITH BLOCKCHAIN TECHNOLOGY

ABSTRACT

For decades artificial intelligence (AI) has been used for various applications in the healthcare industry. Machine learning and artificial intelligence algorithms allow us to diagnose and customize medical care and follow-up plans to get better results, and during the covid19 pandemic, it was found that AI models have been using to predict the Covid-19 symptoms, understanding how it spreads, speeding up research and treatment using medical data. However, it is very challenging to make a robust AI model and use it in a real-time and real-world environment since most organizations do not want to share their data with other third parties due to privacy concerns, furthermore, it is difficult to build a generalized prediction model because of the fragmented nature of the patient data across the healthcare system. To solve the above problems, this paper presents a solution based on blockchain and AI technologies. The blockchain will securely protect the data access and AI-based federated learning for building a robust model for global and real-time usage. Personal healthcare records (PHRs) contain sensitive information that is highly valuable to individuals and healthcare providers. However, traditional methods of storing and sharing PHRs face significant security and privacy challenges, such as data breaches, unauthorized access, and lack of patient control.

Objective: This paper explores the potential of blockchain technology to enhance the security and privacy of PHRs.

Methods: We review existing literature on blockchain technology and its applications in healthcare. We analyze the key features of blockchain, such as immutability, transparency, and decentralization, and discuss how these features can be leveraged to address the challenges of PHR management.

CHAPTER-1

INTRODUCTION

In the digital age, healthcare systems are increasingly shifting toward electronic health records (EHRs) to enhance patient care, streamline operations, and enable data-driven decision-making. However, this digital transformation also introduces serious concerns around data security, privacy, and unauthorized access to sensitive personal health information. Traditional centralized storage models are often vulnerable to data breaches, manipulation, and single points of failure. Blockchain technology offers a promising solution to these challenges by providing a decentralized, tamper-resistant, and transparent platform for managing personal healthcare records. Through its distributed ledger system, blockchain ensures that medical data remains secure, immutable, and accessible only to authorized parties. Smart contracts further enhance control by enabling automated and conditional access to data based on patient consent. By integrating blockchain into healthcare systems, patients gain greater ownership over their records, and healthcare providers can ensure data integrity, confidentiality, and trustworthiness—key components of a secure and efficient digital health ecosystem.

CHAPTER-2

LITERATURE SURVEY

TITLE: Security and Privacy for Healthcare Blockchains

AUTHORS: Rui Zhang, Rui Xue, Ling Liu

ABSTRACT: This paper provides a comprehensive analysis of security and privacy requirements for medical data sharing using blockchain. It discusses various technical solutions, including anonymous signatures, attribute-based encryption, and zero-knowledge proofs, to address security and privacy concerns in healthcare blockchains.

TITLE: A Secure Blockchain Framework for Healthcare Records Management Systems

AUTHORS: Mahmoud Ahmad Al-Khasawneh, Muhammad Faheem, Ala Abdulsalam Alarood, Safa Habibullah, Abdulrahman Alzahrani

ABSTRACT: This study presents a design science-based framework for a secure blockchain system to manage healthcare records. The framework comprises components like blockchain networks, smart contracts, privacy key management, data encryption, and integration with healthcare IT systems to ensure security and privacy in managing healthcare information.

TITLE: A Privacy-Preserving Healthcare Framework Using Hyperledger Fabric

AUTHORS: Charalampos Stamatellis, Pavlos Papadopoulos, Nikolaos Pitropakis, Sokratis Katsikas, William J Buchanan

ABSTRACT: The paper introduces PREHEALTH, a privacy-preserving EHR management solution utilizing Hyperledger Fabric's permissioned blockchain framework and Identity Mixer (Idemix). The solution ensures anonymity and unlinkability in storing patient records, demonstrating efficiency and feasibility for real-world deployment.

CHAPTER-3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The artificial intelligence-based technique already providing promising result in detecting various critical disease using medical images. Mainly the role of the AI based technique in diagnosis of various diseases using medical images is to act as a decision support system for the doctors so that the decision taking process will be faster and at the same time maintaining the precision and accuracy. One of the best techniques is deep learning that has been successful in detecting various diseases using medical images. Sekeroglu and Ozsahin proposed an approach that can detect Covid-19 based on chest X-ray using Convolutional Neural Network. The proposed method performed the classification of images in three groups such as healthy, pneumonia, and Covid-19 with an accuracy of 98.50% with a small imbalanced dataset [10]. Jain et al., proposed a method, that is used to classify two groups such as healthy group and Covid-19 group using chest X-rays. They have used 6432 samples and used various deep learning techniques for classification of these groups.

They have tried different deep learning techniques and found that Xception model provides high accuracy of 97.97%. This model was found to be effective for detecting covid-19 patients [11]. Ozturk et al., proposed an approach that uses binary classification for differentiating two group such as Covid-19 vs No-Findings and Multiclass classification for differentiating three groups such as Covid-19 vs No-Findings vs Pneumonia. They have tried YOLO object detection system using the darknet model. It was found that binary classification provides an accuracy of 98.08% and multiclass classification provides an accuracy of 87.02%. This model would able to help the radiologists for initial screening [12].

Linn and Koo identify quite simple yet very robust use of blockchain for the data storage purpose and leveraging it to the healthcare sector for storing the data of the patients. The proposed system tends to motivate the storage of the complete data starting from the data of birth to the wearable data to the lipid profile data to the MRI data everything on the patients Blockchain [14]. Liangetal mentioned

that user can share data with even the healthcare providers and insurance companies for either getting the services or insurance quotes respectively. The system also employs to be a user centric system where the user has got complete rights regarding the data to be shared or not. This kind of systems can be very helpful in terms of medical research and also preservation of personal medical records which are a real matter of privacy can be very much achieved with Blockchain as an architecture [15].

FL based technique would able to help the healthcare providers because it can improve the accuracy and robustness of the AI model and that helps to make the model more generalizable so that it could be used in real time environment. At the same time this approach would able to save time and cost. The most important part is that we would able to get all the benefits from FL without leakage of information.

DISADVANTAGES

- Our solution is not a differential privacy mechanism that modifies patient data to confuse the hospital records.
- An existing system is not implements Blockchain- Federated Learning Architecture.

3.2 PROPOSED SYSTEM

In the beginning hospitals, universities, pharma work in the isolated place and run the model isolated to each other but later all the trained model are shared to the federated server where the models learn among each other without leaking the private information. Usually all the devices connected to the centralized federated server used to send the data to the server and inside the server all the model averaged and made to a single model. This process keeps on repeating until a high- quality model has been generated. This kind of architecture provides lot of benefits as follows:

- 1) Since the learning process is collaborative in nature the models become smarter and smarter which helps in producing good result at the later stage with unknown data.
- 2) Since the prediction is happening locally in the new model, it lowered down the latency
- 3) In the present approach it is not necessary to be available near the data to get more insight about the data, insight could be obtained from anywhere.
- 4) The most important point is that the privacy remained through the process because all the data are isolated to that particular device, only the trained models are shared to the server.

The blockchain service allows the users (researchers, clinicians, institutions, etc) to access the data by providing a small transaction fee. The non-repudiation and patient's private data are guaranteed. By using the blockchain system, we can know who is accessing the data, when the data was requested, for what reasons. The data access mechanism is described as follow:

- 1) The user creates a user account on the platform,
- 2) The user requests access to a certain resource/asset,
- 3) The smart contract checks if the resource is available in the ledger. If the requested data is available, the smart contract reserves it and notifies the user. The user checks the requirements to access the data and signs the contract. Once the validation is done the user will be charged according to the smart contract's rules. Accordingly, a usage token is issued to the user.

ADVANTAGES

- Privacy is most important, especially in the case of medical imaging data. As the AI based model is getting traction these days especially in the field of medical imaging, protecting the private information is necessary before used in the real time environment.
- The healthcare records are maintaining using blockchain which are very safe and secure.

CHAPTER-4

SYSTEM REQUIREMENTS

4.1 FUNCTIONAL REQUIREMENTS

- Hospitals.
- Healthcare Cloud
- Patient
- Doctor

4.2 NON-FUNCTIONAL REQUIREMENTS

4.2.1 HARDWARE REQUIREMENTS

MINIMUM (Required for Execution)		MY SYSTEM (Development)
System	Pentium IV 2.2 GHz	i3 Processor 5 th Gen
Hard Disk	20 Gb	512 Gb
Ram	1 Gb	4 Gb

4.2.2 SOFTWARE REQUIREMENTS

Operating System	Windows 10/11
Development Software	JAVA 8.1
Programming Language	Java
Front End Technologies	HTML5, CSS3, Java Script
Database Language	SQL

Database (RDBMS)	MySQL
Database Software	MySQL Server
Web Server or Deployment Server	Apache tomcat
Design/Modelling	Rational Rose

CHAPTER-5

SYSTEM STUDY

5.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ **ECONOMICAL FEASIBILITY**
- ◆ **TECHNICAL FEASIBILITY**
- ◆ **SOCIAL FEASIBILITY**

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5.2 FEASIBILITY ANALYSIS

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

- Operational Feasibility
- Economic Feasibility
- Technical Feasibility

OPERATIONAL FEASIBILITY

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

ECONOMIC FEASIBILITY

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

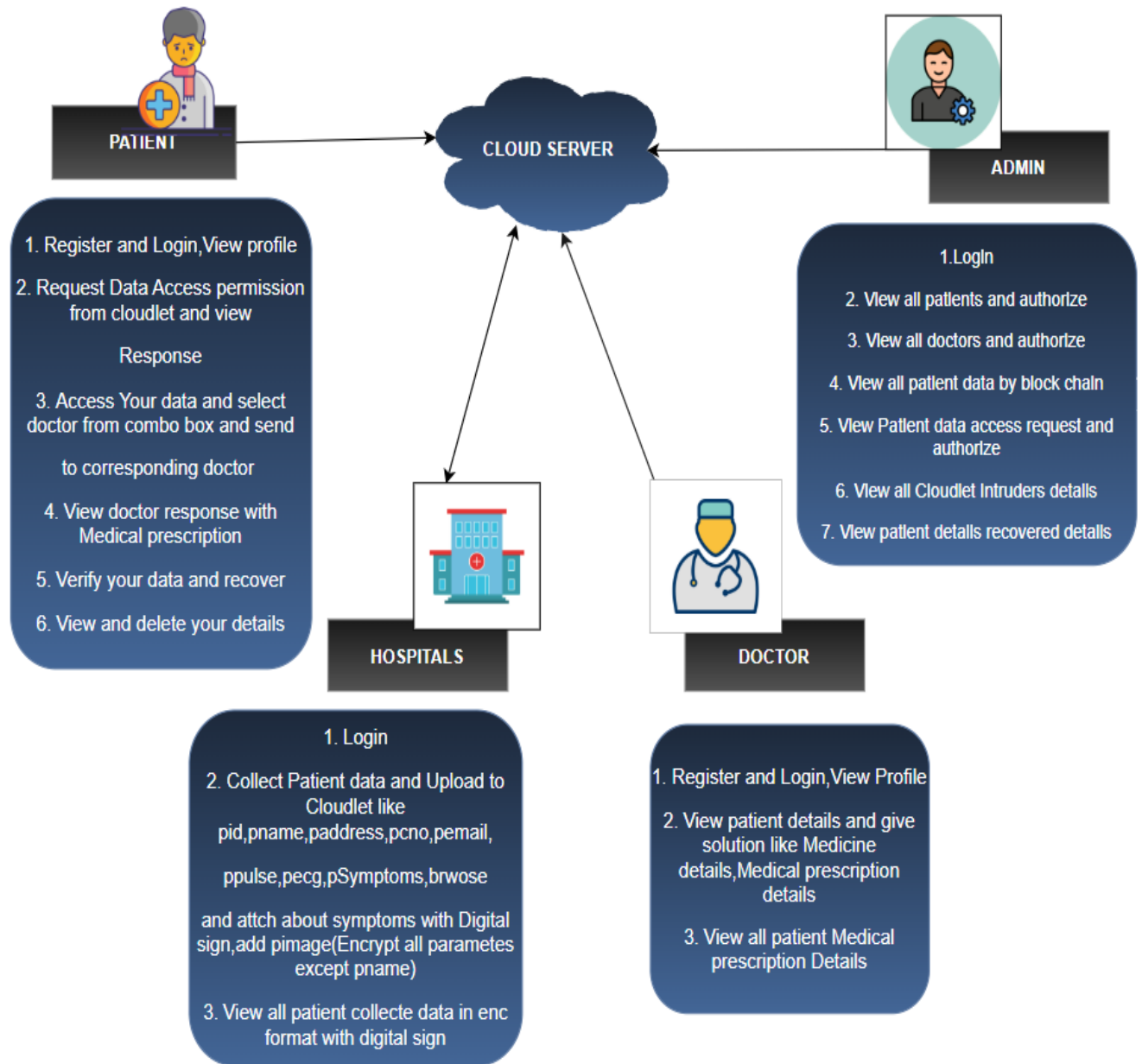
TECHNICAL FEASIBILITY

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform Independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

CHAPTER-6

SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE



6.2 UML DIADRAM

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling 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.

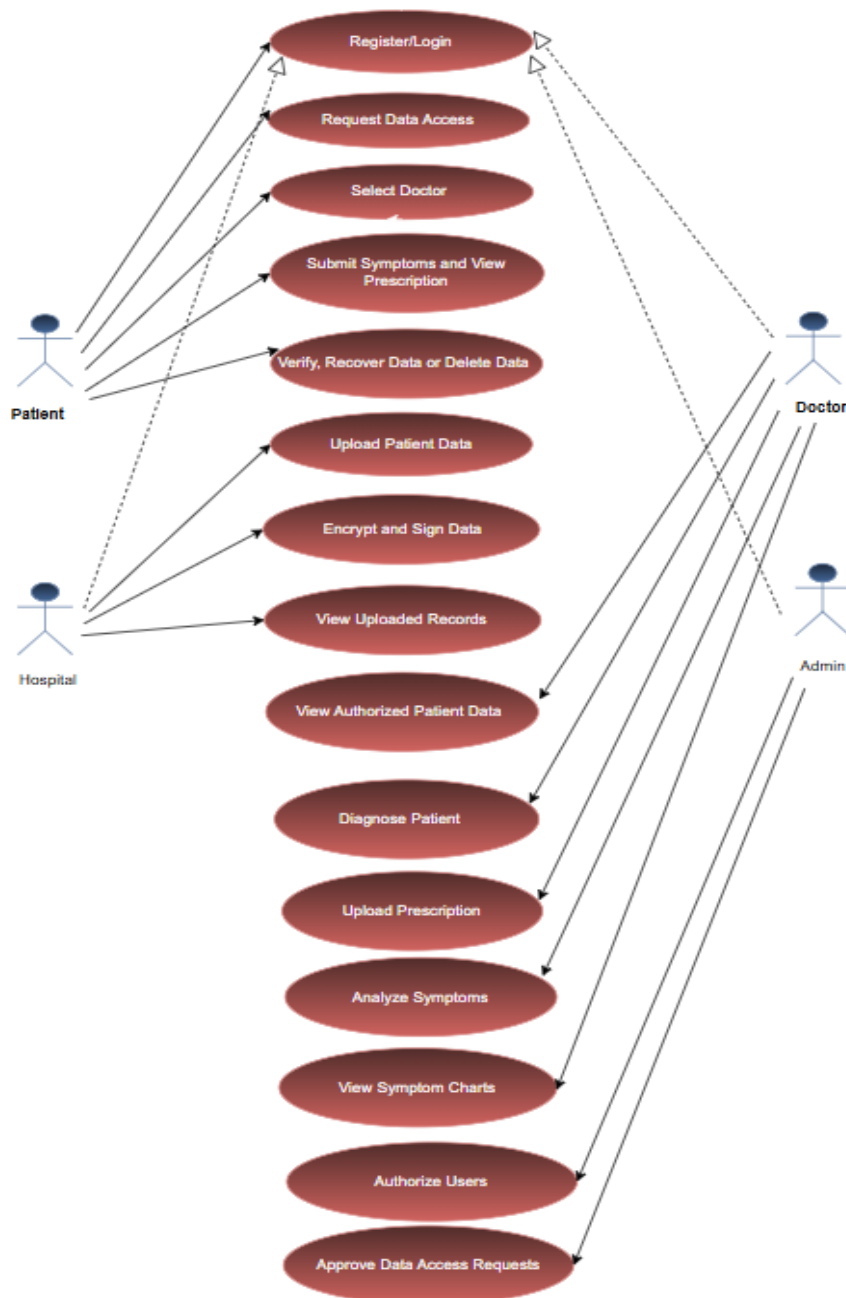
GOALS:

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

1. Provide users a ready-to-use, expressive visual modelling 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 modelling 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.

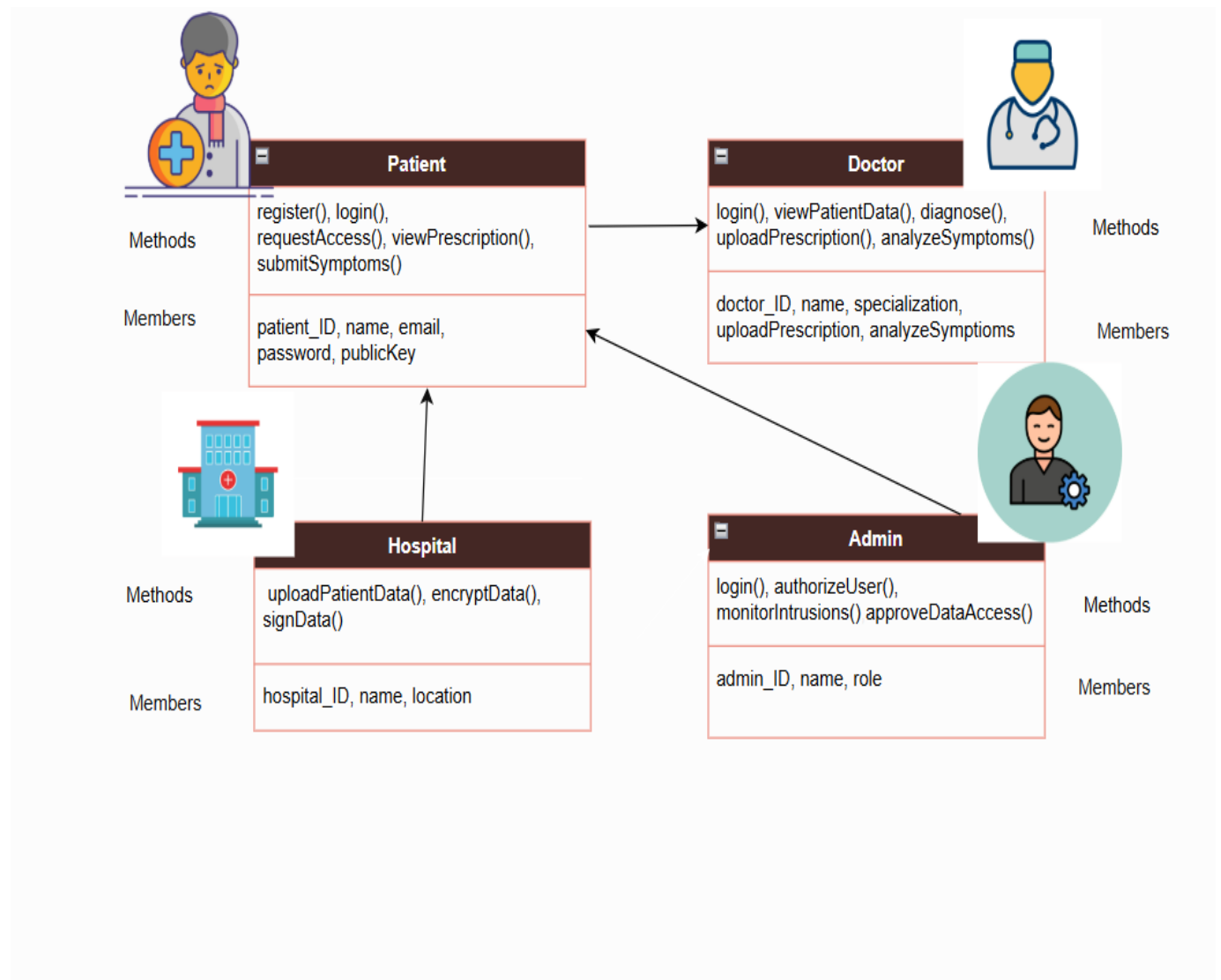
6.2.1 USE CASE DIADRAM

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural 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.



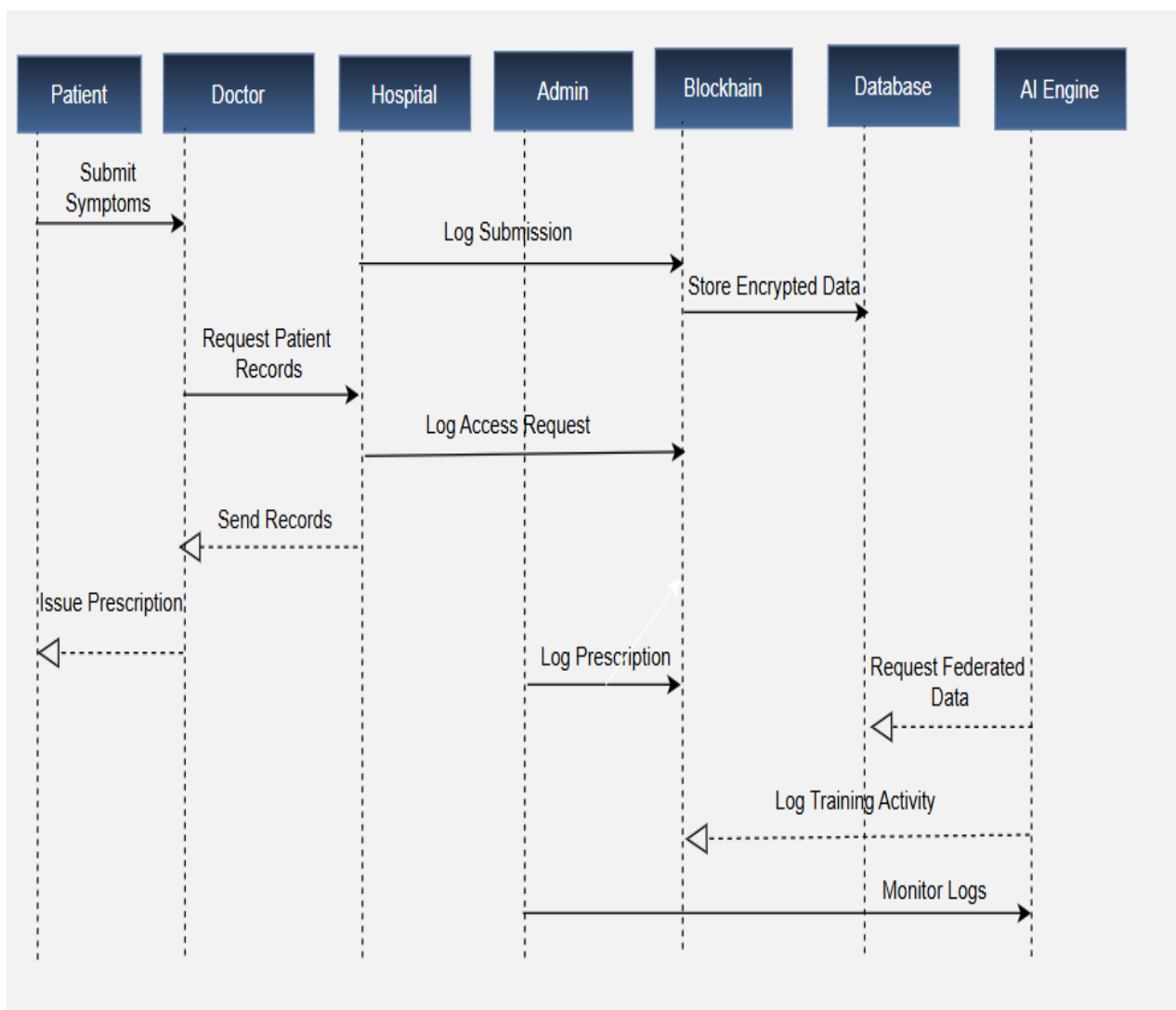
6.2.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling 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.



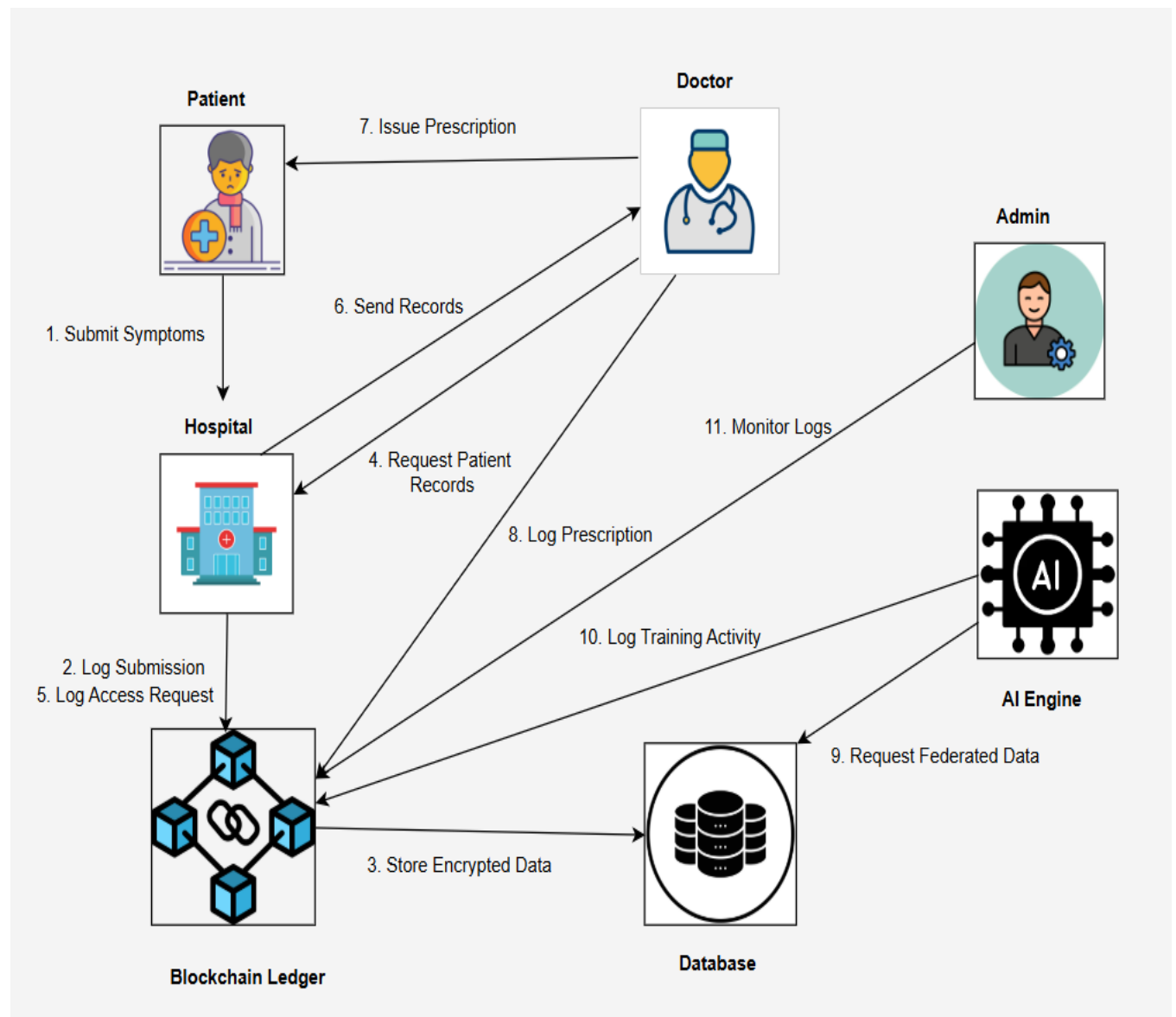
6.2.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modelling 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.



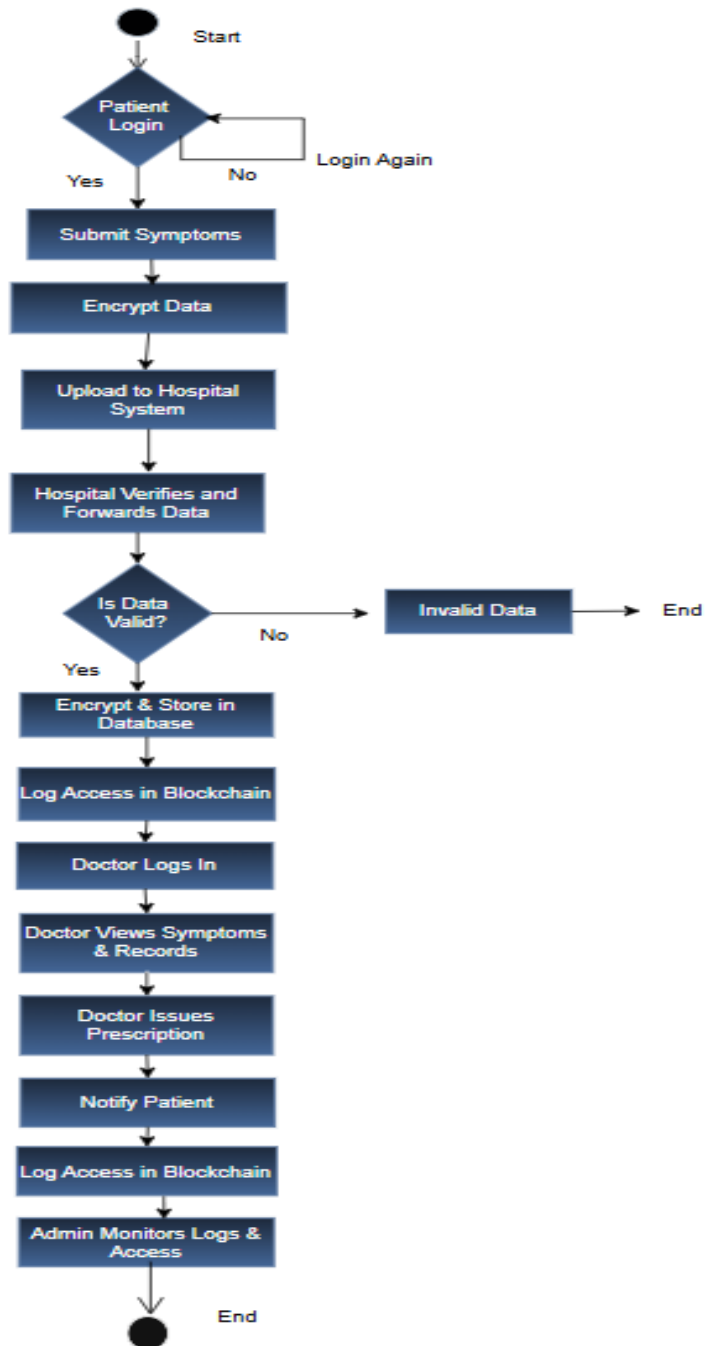
6.2.4 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use case and define role of the object.



6.2.5 ACTIVITY DIAGRAM

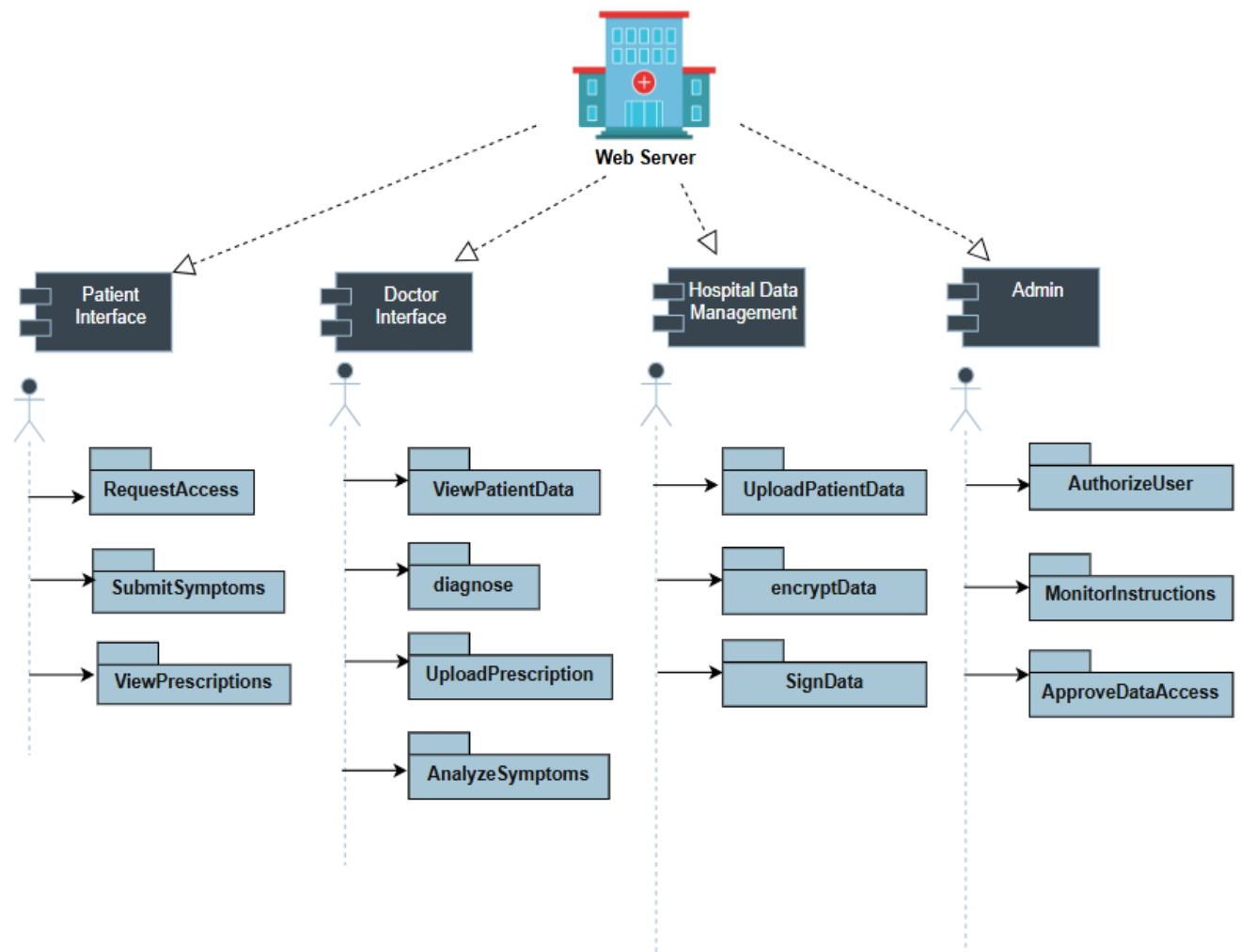
Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of component in a system.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



6.2.6 COMPONENT DIAGRAM

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and

executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behaviours is explained by the provided and required interfaces.

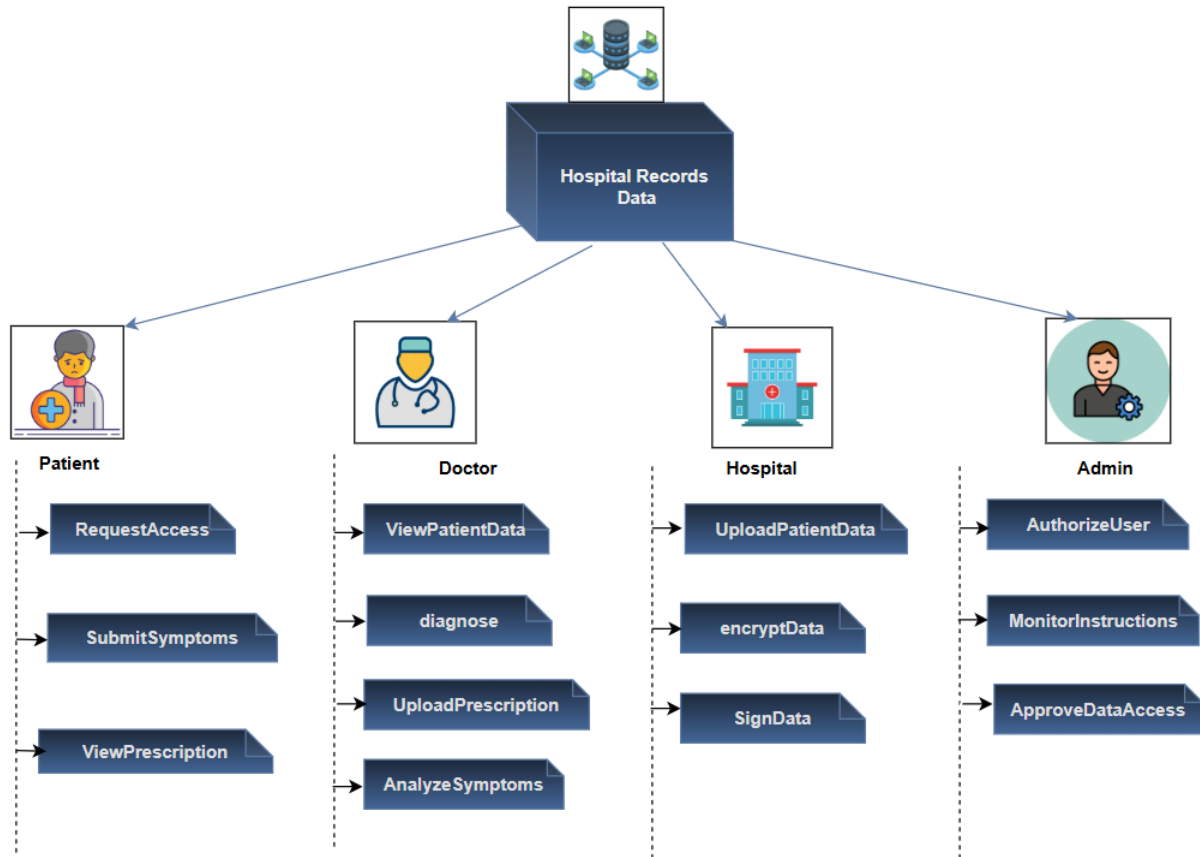


6.2.7 DEPLOYMENT DIAGRAM

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware.

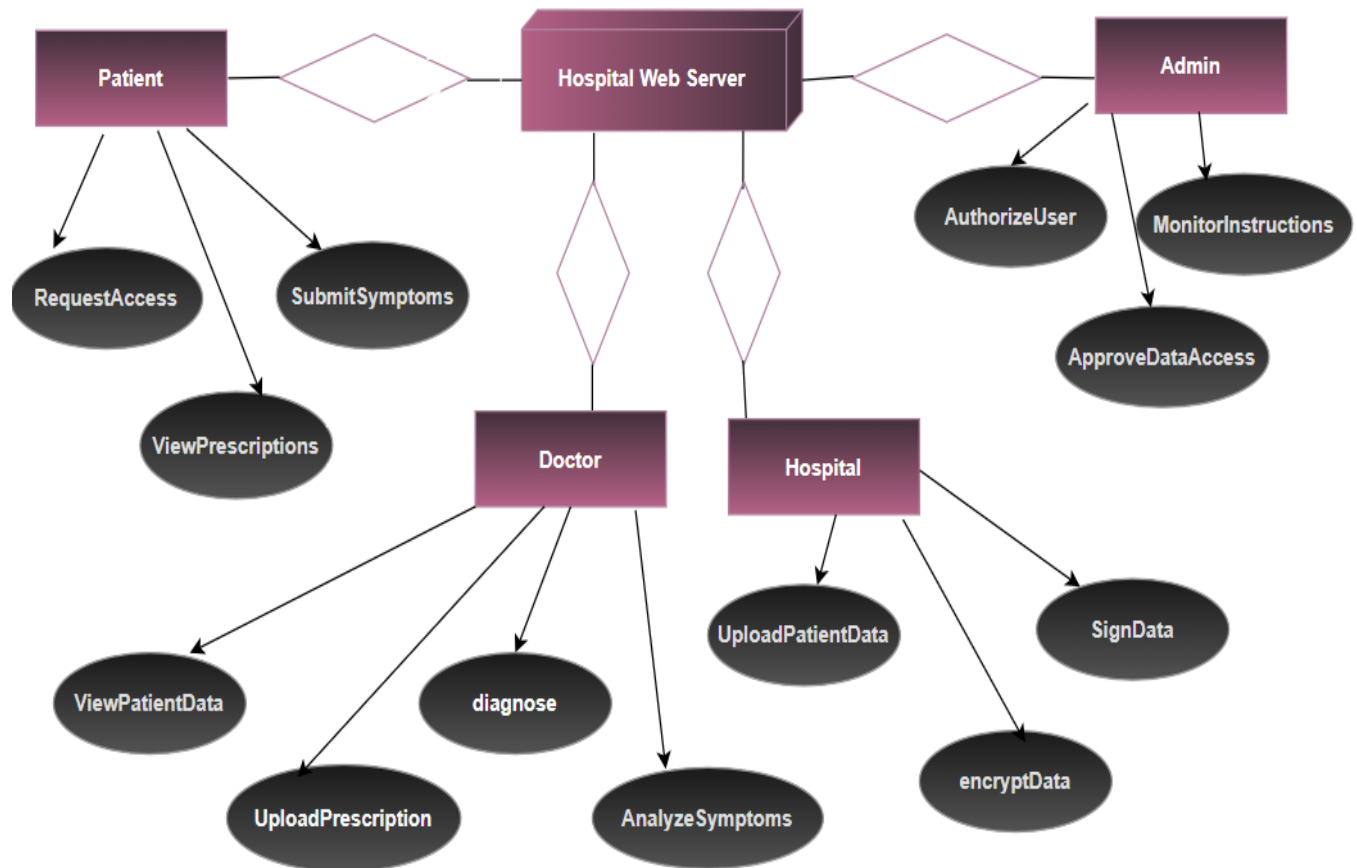
Deployment diagrams help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.



6.2.8 ER DIAGRAM

The Entity Relational Model is a model for identifying entities to be represented in the database and representation of how those entities are related. The ER data model specifies enterprise schema that represents the overall logical structure of a database graphically.

The Entity Relationship Diagram explains the relationship among the entities present in the database. ER models are used to model real-world objects like a person, a car, or a company and the relation between these real-world objects. In short, the ER Diagram is the structural format of the database.



6.2.9 DATA DICTIONARY

HOSPITAL DETAILS

Column name	Data type	constraints	description
-------------	-----------	-------------	-------------

id	Int(11)	Primary key	Hospitals have separate id numbers
name	Varchar(200)	Null	We have to enter hospital name
email	Varchar(200)	Null	We have to enter required email address of hospital
hospitaladdress	Varchar(200)	Null	We have to enter the hospital address
hospitalarea	Varchar(200)	Null	We have to mention near by area

DOCTOR LOGIN

Column name	Data type	constraints	description
User name	Varchar(100)	Not null	Doctor enters his username
password	Int	Primary key	Doctor enters his password for login

PATIENT REGISTER

Column name	Data type	Constraints	description
id	Int(11)	Primary key	Patient id
name	Varchar(200)	Null	patient name
email	Varchar(200)	Null	patient email
mobile	Varchar(200)	Null	patient mobile number
address	Varchar(200)	Null	patient address where he was belongs
User name	Varchar(200)	Null	patient requires username
password	Varchar(200)	Null	patient requires the user password

CHAPTER-7

INPUT/OUTPUT DESIGN

7.1 INPUT DESIGN

Input Design plays a vital role in the life cycle of software development; it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations.

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error in the input are controlled by the input design. The application has been developed in user-friendly manner. The forms have been designed in such a way during the processing the cursor is placed in the position where must be entered. The user is also provided with in an option to select an appropriate input from various alternatives related to the field in certain cases.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

7.2 OUTPUT DESIGN

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client

on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client. User authentication procedures are maintained at the initial stages itself. A new user may be created by the administrator himself or a user can himself register as a new user but the task of assigning projects and validating a new user rest with the administrator only.

The application starts running when it is executed for the first time. The server has to be started and then the internet explorer is used as the browser. The project will run on the local area network so the server machine will serve as the administrator while the other connected systems can act as the clients. The developed system is highly user friendly and can be easily understood by anyone using it even for the first time.

CHAPTER-8

IMPLEMENTATION

8.1 MODULES

- Hospitals.
- Healthcare Cloud
- Patient
- Doctor

8.1.1 MODULE DESCRIPTION

- **Hospitals**

In this module, the Hospitals Collect Patient data and Upload to Cloudlet like pid,pname,paddress,pcno,pemail,ppulse,pecg,pSymptoms,brwose and attach about symptoms with Digital sign,add pimage(Encrypt all parametes except pname) and View all patient collecte data in enc format with digital sign.

- **Healthcare Cloud**

The **Cloud** server manages which is to provide data storage service for the **wearable devices** and also View all patients and authorize and View all doctors and authorize ,Vliew all patient Cloudlet data with enc format ,View Patient data access request and authorize ,View all Cloudlet Intruders details and View patient details recovered details ,View No.Of same symptoms in Chart(Symptom name vs No. Of Patients),View No.Of Patients refered same doctor in Chart(Doctor name vs No.Of Patients).

- **Patient**

In this module, the patient Register and Login, View profile ,Request Data Access permission from cloudlet and view Response, Access Your data and select doctor from combo box and send to corresponding doctor and View doctor response with Medical prescription, Verify your data and recover and View and delete your details.

- **Doctor**

The doctor is the one who will perform the following operations such as Register and Login, View Profile, View patient details and give solution like Medicine details, Medical prescription details View all patient Medical prescription Details.

CHAPTER-9

SOFTWARE ENVIRONMENT

9.1 JAVA

Java Technology

Java technology is both a programming language and a platform.

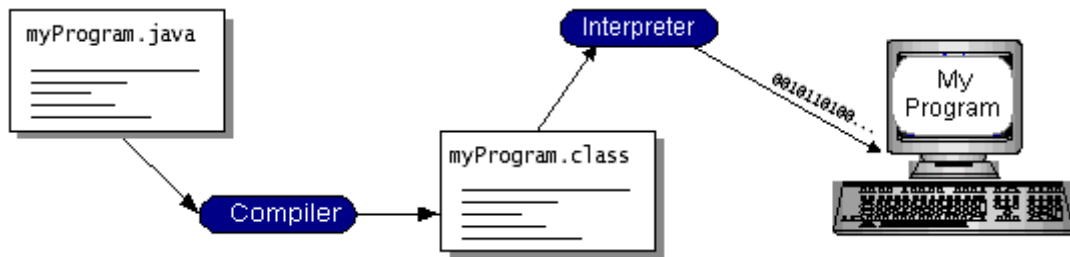
The Java Programming Language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

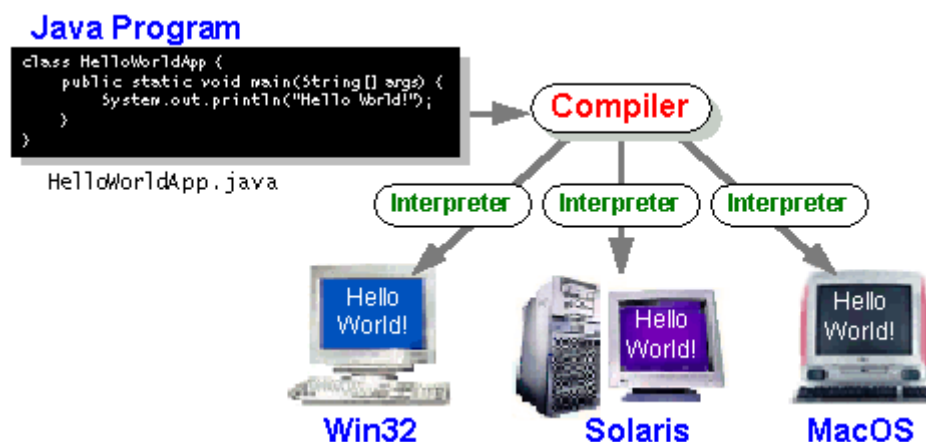
- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make "write once, run anywhere" possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.



The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

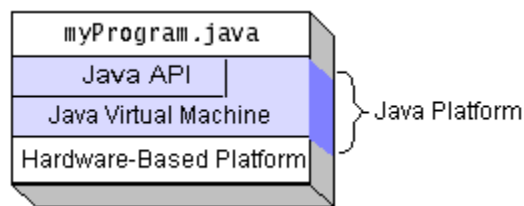
The Java platform has two components:

- The Java Virtual Machine (Java VM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

What Can Java Technology Do?

The most common types of programs written in the Java programming language are applets and applications. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language is also a powerful software platform. Using the generous API, you can write many types of programs.

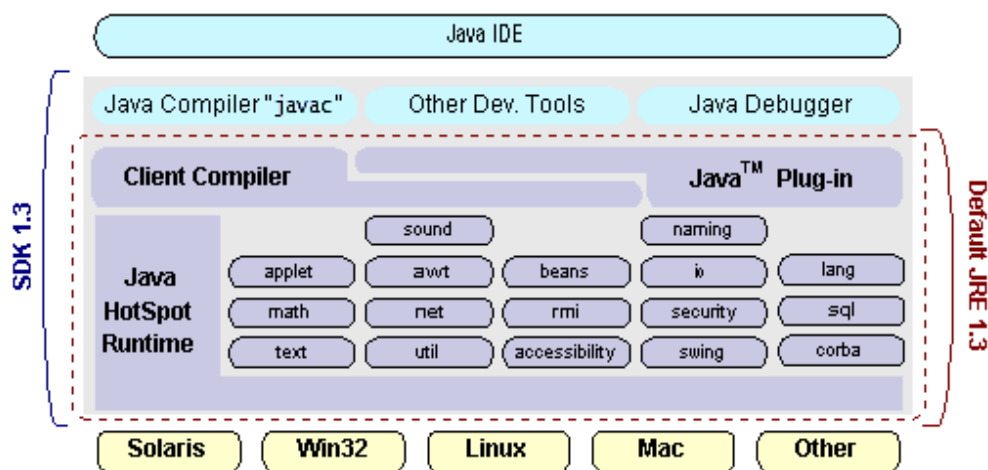
An application is a standalone program that runs directly on the Java platform. A special kind of application known as a server serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a servlet. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

- **The essentials:** Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.
- **Applets:** The set of conventions used by applets.
- **Networking:** URLs, TCP (Transmission Control Protocol), UDP (User Datagram Protocol) sockets, and IP (Internet Protocol) addresses.
- **Internationalization:** Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.
- **Security:** Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.
- **Software components:** Known as JavaBeansTM, can plug into existing component architectures.

- **Object serialization:** Allows lightweight persistence and communication via Remote Method Invocation (RMI).
- **Java Database Connectivity (JDBC™):** Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



How Will Java Technology Change My Life?

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

- **Get started quickly:** Although the Java programming language is a powerful object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.
- **Write less code:** Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.

- **Write better code:** The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.
- **Develop programs more quickly:** Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.
- **Avoid platform dependencies with 100% Pure Java:** You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java™ Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.
- **Write once, run anywhere:** Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.
- **Distribute software more easily:** You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded "on the fly," without recompiling the entire program.

ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a de facto standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of “plug-in” database connectivity modules, or drivers. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC’s framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

1. SQL Level API

The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to “generate” JDBC code and to hide many of JDBC’s complexities from the end user.

2. SQL Conformance

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the connectivity module to handle non-standard functionality in a manner that is suitable for its users.

3. JDBC must be implemental on top of common database interfaces

The JDBC SQL API must “sit” on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

4. Provide a Java interface that is consistent with the rest of the Java system

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

5. Keep it simple

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

6. Use strong, static typing wherever possible

Strong typing allows for more error checking to be done at compile time; also, less error appear at runtime.

7. Keep the common cases simple

Because more often than not, the usual SQL calls used by the programmer are simple SELECT’s, INSERT’s, DELETE’s and UPDATE’s, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Finally we decided to proceed the implementation using Java Networking.

And for dynamically updating the cache table we go for MS Access database.

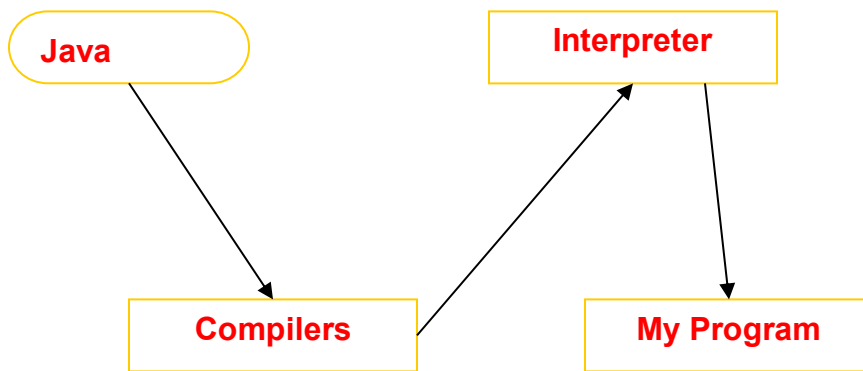
Java has two things: a programming language and a platform.

Java is a high-level programming language that is all of the following

Simple	Architecture-neutral
Object-oriented	Portable
Distributed	High-performance
Interpreted	multithreaded
Robust	Dynamic
Secure	

Java is also unusual in that each Java program is both compiled and interpreted. With a compiler you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.



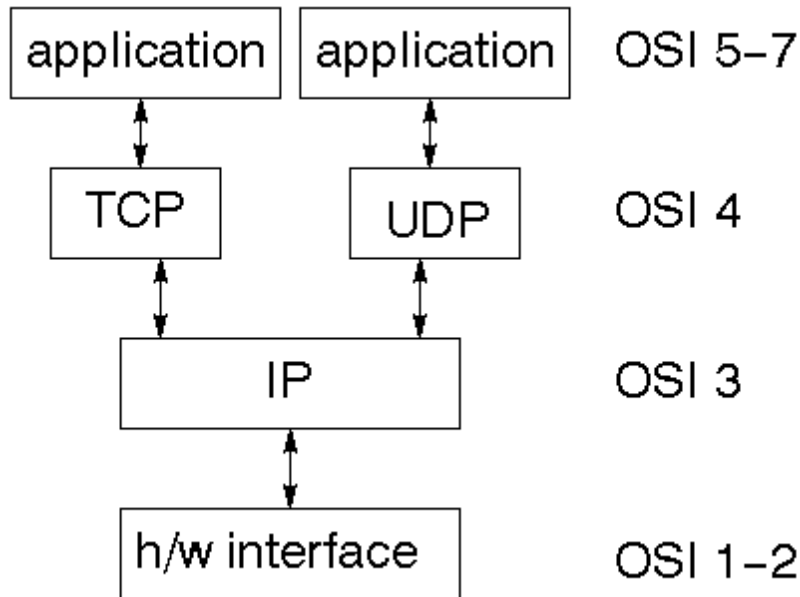
You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of the Java VM. The Java VM can also be implemented in hardware.

Java byte codes help make “write once, run anywhere” possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

Networking

TCP/IP stack

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

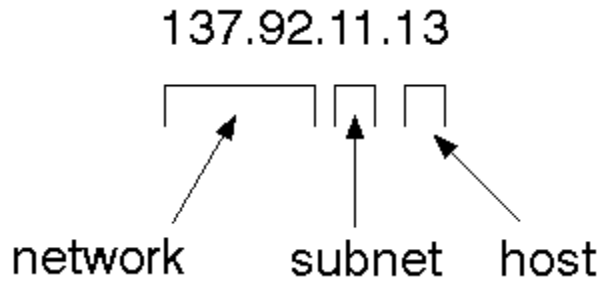
Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

Total address



The 32 bit address is usually written as 4 integers separated by dots.

Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call `socket`. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
int socket(int family, int type, int protocol);
```

Here "family" will be `AF_INET` for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

JFree Chart

JFreeChart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. JFreeChart's extensive feature set includes:

A consistent and well-documented API, supporting a wide range of chart types;

A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

JFreeChart is "open source" or, more specifically, [free software](#). It is distributed under the terms of the [GNU Lesser General Public Licence](#) (LGPL), which permits use in proprietary applications.

1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XYPlot class in JFreeChart;

Testing, documenting, testing some more, documenting some more.

2. Time Series Chart Interactivity

Implement a new (to JFreeChart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of JFreeChart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

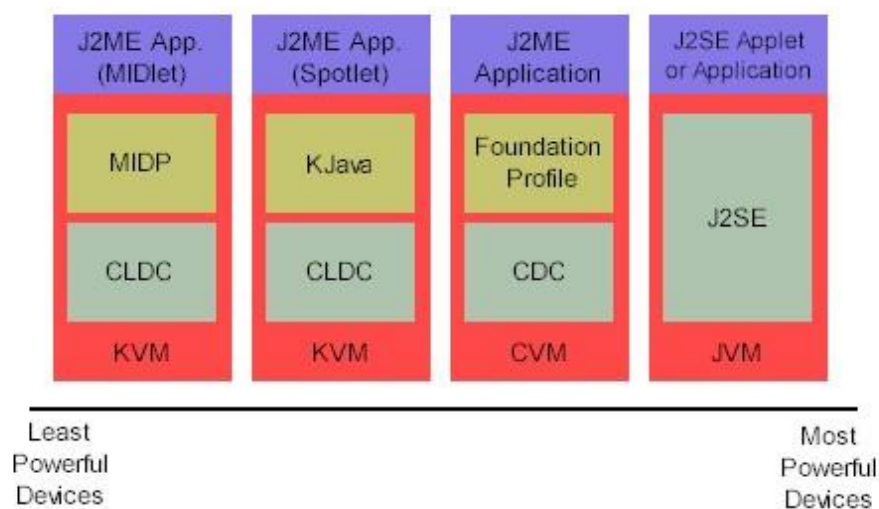
4. Property Editors

The property editor mechanism in JFreeChart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

J2ME (Java 2 Micro edition):-

Sun Microsystems defines J2ME as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems." Announced in June 1999 at the JavaOne Developer Conference, J2ME brings the cross-platform functionality of the Java language to smaller devices, allowing mobile wireless devices to share applications. With J2ME, Sun has adapted the Java platform for consumer products that incorporate or are based on small computing devices.

1. General J2ME architecture



J2ME uses configurations and profiles to customize the Java Runtime Environment (JRE). As a complete JRE, J2ME is comprised of a configuration, which determines the JVM used, and a profile, which defines the application by adding domain-specific classes. The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. We'll discuss configurations in detail in the The profile defines the application; specifically, it adds domain-specific classes to the J2ME configuration to define certain uses for devices. We'll cover profiles in depth in the The following graphic depicts the relationship between the different virtual machines, configurations, and profiles. It also draws a parallel with the J2SE API and its Java virtual machine. While the J2SE virtual machine is generally referred to as a JVM, the J2ME virtual machines, KVM and CVM, are subsets of JVM. Both KVM and CVM can be

thought of as a kind of Java virtual machine -- it's just that they are shrunken versions of the J2SE JVM and are specific to J2ME.

2.Developing J2ME applications

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role preverification plays in this process.

3.Design considerations for small devices

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "gotchas" before developing the application can be a painful process. Here are some design strategies to consider:

- * Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.
- * Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed Java Archive (jar) files.
- * Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

4.Configurations overview

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

* **Connected Limited Device Configuration (CLDC)** is used specifically with the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

* **Connected Device Configuration (CDC)** is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

5.J2ME profiles

What is a J2ME profile?

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain-specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: KJava and MIDP. Both KJava and MIDP are associated with CLDC and smaller devices. Profiles are built on top of configurations. Because profiles are specific to the size of the device (amount of memory) on which an application runs, certain profiles are associated with certain configurations.

A skeleton profile upon which you can create your own profile, the Foundation Profile, is available for CDC.

Profile 1: KJava

KJava is Sun's proprietary profile and contains the KJava API. The KJava profile is built on top of the CLDC configuration. The KJava virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. KJava contains a Sun-specific API that runs on the Palm OS. The KJava API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However, because it is not a standard J2ME package, its main package is `com.sun.kjava`.

We'll learn more about the KJava API later in this tutorial when we develop some sample applications.

Profile 2: MIDP

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like KJava, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a common, industry-standard profile for mobile devices that is not dependent on a specific vendor. It is a complete and supported foundation for mobile application

development. MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

- * java.lang

- * java.io

- * java.util

- * javax.microedition.io

- * javax.microedition.lcdui

- * javax.microedition.midlet

- * javax.microedition.rms

9.2 SOURCE CODE

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Home Page</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-aller.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

<script type="text/javascript" src="js/coin-slider.min.js"></script>

<style type="text/css">

<!--

.style1 {font-size: 40px}

.style2 {

        color: #FF0000;

        font-weight: bold;

}

.style3 {color: #0000FF}

-->
```

```
</style>

</head>

<body>

<div class="main">

  <div class="header">

    <div class="header_resize">

      <div class="menu_nav">

        <ul>

          <li class="active"><a href="index.html"><span>Home Page</span></a></li>

          <li><a href="c_login.jsp"><span>Healthcare Cloud</span></a></li>

          <li><a href="d_login.jsp"><span>Doctor</span></a></li>

          <li><a href="p_login.jsp"><span>Patient</span></a></li>

          <li><a href="w_login.jsp"><span>Hospitals</span></a></li>

          <li><a href="i_login.jsp"><span>Intruder</span></a></li>

        </ul>

      </div>

      <div class="logo">

        <h1><a href="index.html" class="style1">Protecting Personal Healthcare Record Using
Blockchain and Federated Learning Technologies</a></h1>

      </div>

      <div class="clr"></div>

      <div class="slider">
```

```
<div id="coin-slider"> <a href="#"> </a> <a href="#"> </a> <a href="#"> </a> </div>
```

```
</div>
```

```
<div class="clr"></div>
```

```
</div>
```

```
</div>
```

```
<div class="content">
```

```
<div class="content_resize">
```

```
<div class="mainbar">
```

```
<div class="article">
```

```
<h2><span>Protecting Personal Healthcare Record Using Blockchain and Federated
Learning Technologies</span></h2>
```

```
<p class="infopost style2">Artificial intelligence, blockchain, federated learning,
privacy, pandemic</p>
```

```
<div class="clr"></div>
```

```
<div class="img"></div>
```

```
<div class="post_content">
```

```
<p align="justify" class="style2"><strong>For decades artificial intelligence (AI) has
been used for various applications in the healthcare industry. Machine learning and artificial
intelligence algorithms allow us to diagnose and customize medical care and follow-up plans to
get better results, and during the covid19 pandemic, it was found that AI models have been
using to predict the Covid-19 symptoms, understanding how it spreads, speeding up research
and treatment using medical data. However, it is very challenging to make a robust AI model
```

and use it in a real-time and real-world environment since most organizations do not want to share their data with other third parties due to privacy concerns, furthermore, it is difficult to build a generalized prediction model because of the fragmented nature of the patient data across the healthcare system. To solve the above problems, this paper presents a solution based on blockchain and AI technologies. The blockchain will securely protect the data access and AI-based federated learning for building a robust model for global and real-time usage.

</div>

<div class="clr"></div>

</div>

</div>

<div class="sidebar">

<div class="searchform">

<form id="formsearch" name="formsearch" method="post" action="#">

<input name="editbox_search" class="editbox_search" id="editbox_search" maxlength="80" value="Search our ste:" type="text" />

<input name="button_search" src="images/search.gif" class="button_search" type="image" />

</form>

</div>

<div class="clr"></div>

<div class="gadget">

<h2 class="star">Sidebar Menu</h2>

<div class="clr"></div>

<ul class="sb_menu">

Home Page

Cloudlet

Doctor

Patient

Hospitals

Intruder

</div>

<div class="gadget">

<h2 class="star">Concepts</h2>

<div class="clr"></div>

<ul class="ex_menu">

Artificial intelligence,

blockchain,

federated learning,

privacy, pandemic

</div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="fbg"></div>

<div class="footer">

<div class="footer_resize">

<div style="clear:both;"></div>

</div>

</div>

</div>

<div align=center></div>

</body>

</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>Patient Details </title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<link href="css/style.css" rel="stylesheet" type="text/css" />

<link rel="stylesheet" type="text/css" href="css/coin-slider.css" />

<script type="text/javascript" src="js/cufon-yui.js"></script>

<script type="text/javascript" src="js/cufon-aller.js"></script>

<script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>

<script type="text/javascript" src="js/script.js"></script>

```
<script type="text/javascript" src="js/coin-slider.min.js"></script>
```

```
<style type="text/css">
```

```
<!--
```

```
.style1 {font-size: 40px}
```

```
.style6 {font-size: 18px}
```

```
.style7 {font-family: "Times New Roman", Times, serif}
```

```
.style8 {font-size: 20px}
```

```
.style9 {color: #0000FF}
```

```
.style10 {font-family: "Times New Roman", Times, serif; font-size: 20px; }
```

```
.style13 {color: #000000}
```

```
.style14 {font-family: "Times New Roman", Times, serif; font-size: 20px; color: #000000; }
```

```
.style15 {font-family: "Times New Roman", Times, serif; font-size: 20px; color: #FF0000; }
```

```
-->
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="main">
```

```
<div class="header">
```

```
<div class="header_resize">
```

```
<div class="menu_nav">
```

```
<ul>
```

```
<li><a href="index.html"><span>Home Page</span></a></li>
```

Cloudlet

<li class="active">Doctor

Patient

Hospitals

Intruder

</div>

<div class="logo">

<h1>Protecting Personal Healthcare Record Using
Blockchain and Federated Learning Technologies</h1>

</div>

<div class="clr"></div>

<div class="slider">

<div id="coin-slider"> </div>

</div>

<div class="clr"></div>

</div>

</div>

<div class="content">

<div class="content_resize">

<div class="mainbar">

<div class="article">

<h2 align="center"> View Patient Details and Give Solution </h2>

<p> </p>

<%@page import="java.io.BufferedInputStream"%>

<%@page import="java.security.DigestInputStream"%>

<%@page import="java.io.FileInputStream"%>

<%@page import="java.io.PrintStream"%>

<%@page import="java.io.FileOutputStream"%>

<%@page import="java.math.BigInteger"%>

<%@page

import="java.security.Key,java.security.KeyPair,java.security.KeyPairGenerator,javax.crypto.Cipher"%>

<%@page

import="java.util.*,java.security.Key,java.util.Random,javax.crypto.Cipher,javax.crypto.spec.SecretKeySpec,org.bouncycastle.util.encoders.Base64"%>

<%@page import="java.security.MessageDigest"%>

<%@page import="java.sql.Statement"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.text.SimpleDateFormat"%>

<%@page import="java.util.Date"%>

```
<%@page
```

```
import="com.oreilly.servlet.*,java.sql.*,java.lang.*,java.text.SimpleDateFormat,java.util.*,java.i  
o.*,javax.servlet.*,javax.servlet.http.*" %>
```

```
<%@ page import="java.sql.*"%>
```

```
<%@ include file="connect.jsp"%>
```

```
<%
```

```
try
```

```
{
```

```
String
```

```
s1="",s2="",s3="",s4="",s5="",s6="",s7="",s8="",s9="",s10="",s11="",s12="",s13="",s17="";
```

```
String decs3="",decs4="",decs5="",decs6="",decs7="",decs8="",decs9="",decs17="";
```

```
int i=0,j=1,k=0;
```

```
//String keys = "";
```

```
//String pname = request.getParameter("pat");
```

```
int id = Integer.parseInt(request.getParameter("pid"));
```

```
String ss4="";
```

```
String query="select * from doctor_files where id="+id+" ";
```

```
Statement st=connection.createStatement();
```

```
ResultSet rs=st.executeQuery(query);
```

```
if ( rs.next() )
```

```
{
```

```
    i=rs.getInt(1);
```

```
        s2=rs.getString(2);//name
```

```
        s3=rs.getString(3);//name
```

```
        s4=rs.getString(4);//mail
```

```
        s5=rs.getString(5);
```

```
        s6=rs.getString(6);//add
```

```
        s7=rs.getString(7);//dob
```

```
        s8=rs.getString(8);
```

```
        s9=rs.getString(9);
```

```
        s10=rs.getString(10);
```

```
        s11=rs.getString(11);
```

```
        s12=rs.getString(12);//
```

```
        s17=rs.getString(17);//
```

```
String keys="q2e34rrfgfggg2a";
```

```
byte[] keyValue = keys.getBytes();
```

```
Key key = new SecretKeySpec(keyValue, "AES");
```

```
Cipher c = Cipher.getInstance("AES");
```

```
c.init(Cipher.ENCRYPT_MODE, key);
```

```
decs3 = new String(Base64.decode(s3.getBytes()));
```

```
decs4 = new String(Base64.decode(s4.getBytes()));
```

```
decs5 = new String(Base64.decode(s5.getBytes()));  
decs6 = new String(Base64.decode(s6.getBytes()));  
decs7 = new String(Base64.decode(s7.getBytes()));  
decs8 = new String(Base64.decode(s8.getBytes()));  
decs9 = new String(Base64.decode(s9.getBytes()));  
decs17 = new String(Base64.decode(s17.getBytes()));
```

```
%>
```

```
<table width="620" border="1" align="center" cellpadding="10">
```

```
<tr>
```

```
<td width="286" bgcolor="#99CCCC"><div align="left" class="style3 style7 style11  
style12 style8 style9"><span class="style4 style12">Patient Image :- </span></div></td>
```

```
<td width="356" bgcolor="#CCCC00"><input name="image" type="image"  
style="width:100px; height:90px;" src="d_p_Pic.jsp?id=<%=i%>" /></td>
```

```
</tr>
```

```
<tr>
```

<td width="245" bgcolor="#99CCCC"><div align="left" class="style3 style7 style11 style12 style8 style9">Patient Name :- </div></td>

<td width="329" bgcolor="#CCCC00"><%=s2%></td>
</tr>

<tr>

<td width="245" bgcolor="#99CCCC"><div align="left" class="style3 style7 style11 style12 style8 style9">Contact Number :- </div></td>

<td width="329" bgcolor="#CCCC00"><%=decs3%></td>
</tr>

<tr>

<td width="245" bgcolor="#99CCCC"><div align="left" class="style3 style7 style11 style12 style8 style9">E-mail :- </div></td>

<td width="329" bgcolor="#CCCC00"><%=decs4%></td>
</tr>

<tr>

<td width="245" bgcolor="#99CCCC"><div align="left" class="style3 style7 style11 style12 style8 style9">Patient Address :- </div></td>

<td width="329" bgcolor="#CCCC00"><%=decs5%></td>
</tr>

<tr>

<div align="left" class="style3 style7 style11 style12 style8 style9">Pulses :-</div>	<%=decs6%>
---	---

<tr>

<div align="left" class="style3 style7 style11 style12 style8 style9">ECG :-</div>	<%=decs7%>
--	---

<tr>

<div align="left" class="style3 style7 style11 style12 style8 style9">Symptoms :-</div>	<%=decs17%>
---	--

<tr>

<div align="left" class="style3 style7 style11 style12 style8 style9">Symptoms File Name :-</div>	<%=decs8%>
---	---

<tr>

<td width="245" bgcolor="#99CCCC"><div align="left" class="style3 style7 style11 style12 style8 style9">File Content :- </div></td>

<td bgcolor="#CCCC00">

<textarea name="p9" id="textarea" cols="52" rows="15" readonly="readonly"><%=decs9%></textarea>

</td>

</tr>

<tr>

<td width="245" bgcolor="#99CCCC"><div align="left" class="style3 style7 style11 style12 style8 style9">Digital Sign (MAC) :- </div></td>

<td width="329" bgcolor="#CCCC00"><%=s10%></td>

</tr>

<tr>

<td height="52" colspan="2" align="center" bgcolor="#666666"><a href="d_pat_solve2.jsp?pid=<%=i%>" class="style8"> Give Solution </td>

</tr>

</table>

<%

connection.close();

}

```
        catch(Exception e)

        {

            out.println(e.getMessage());

        }

%>

<p>&nbsp;</p>

        <p align="right"><a href="d_pat_solve.jsp" class="style6">Back</a></p>

</div>

</div>

<div class="sidebar">

    <div class="gadget">

        <h2 class="star"><span>Doctor</span> Menu</h2>

        <div class="clr"></div>

        <ul class="sb_menu">

            <li><a href="d_main.jsp"> Doctor Main </a></li>

            <li><a href="d_login.jsp"> Log Out </a></li>

        </ul>

    </div>

</div>

<div class="clr"></div>

</div>

</div>
```

```
<div class="fbg">
```

```
<div class="fbg_resize">
```

```
<div class="col c1">
```

```
<h2><span>Image</span> Gallery</h2>
```

```
<a href="#"></a>
<a href="#"></a> <a
href="#"></a> <a
href="#"></a> <a
href="#"></a> <a
href="#"></a> </div>
```

```
<div class="clr"></div>
```

```
</div>
```

```
</div>
```

```
<div class="footer">
```

```
<div class="footer_resize">
```

```
<div style="clear:both;"></div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div align=center></div>
```

```
</body>
```

```
</html>
```

CHAPTER-10

RESULTS/DISCUSSIONS

10.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and

consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

10.1.1 SYSTEM TESTING

TESTING METHODOLOGIES

The following are the Testing Methodologies:

- Unit testing.
- Integration testing.
- User acceptance testing.
- Output testing.
- Validation testing.

Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1)Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

OTHER TESTING METHODOLOGIES

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

The package “Virtual Private Network” has satisfied all the requirements specified as per software requirement specification and was accepted.

USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

TESTING STRATEGY :

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation .A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

In unit testing different modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this. The future holds a lot to offer to the development and refinement of this project.

10.2 OUTPUT SCREENSHOTS


Home Page



Healthcare Cloud Login



Healthcare Doctor Registration Page



Sidebar Menu

Home Page
Cloudlet
Doctor
Patient
Hospitals
Intruder

Doctor Registration

Name (required)

Password (required)

Email id (required)

Mobile Number (required)

Your Address

Date of Birth (required)

Select Gender (required)

Enter Pincode (required)

Enter Location (required)


Select Profile Picture (required)

No file chosen

Healthcare Doctor Login Page

Home Page
Cloudlet
Doctor
Patient
Hospitals
Intruder

Protecting Personal Healthcare Record Using
Blockchain and Federated Learning
Technologies



Sidebar Menu

Home Page
Cloudlet
Doctor
Patient
Hospitals
Intruder

Doctor Login


Name

Password

New User? [Register Here](#)

Patient Registration Page

Protecting Personal Healthcare Record Using Blockchain and Federated Learning Technologies



Navbar

- Home Page
- Cloudlet
- Doctor
- Patient
- Hospital
- Intruder

Sidebar Menu

Patient Registration

Name (required)

Password (required)

Email id (required)

Mobile Number (required)

Your Address

Date of Birth (required)

Select Gender (required)

Select

Enter Pincode (required)

Enter Location (required)

Select Profile Picture (required)

Choose File No file chosen

Patient Login Page

Protecting Personal Healthcare Record Using Blockchain and Federated Learning Technologies



Navbar

- Home Page
- Cloudlet
- Doctor
- Patient
- Hospital
- Intruder

Sidebar Menu

Patient Login

Name

Password

LoginReset

New User? [Register Here](#)

Image Gallery

Hospital Login Page

[Home Page](#) [Cloudlet](#) [Doctor](#) [Patient](#) [Hospitals](#) [Intruder](#)

Protecting Personal Healthcare Record Using Blockchain and Federated Learning Technologies



Sidebar Menu

Home Page
Cloudlet
Doctor
Patient
Hospitals
Intruder

Hospitals Login

Select Block

-- Select --

User Name

Password

Login Reset

Image Gallery



Intruders Page

[Home Page](#) [Cloudlet](#) [Doctor](#) [Patient](#) [Hospitals](#) [Intruder](#)

Protecting Personal Healthcare Record Using Blockchain and Federated Learning Technologies



Sidebar Menu

Home Page
Cloudlet
Doctor
Patient
Hospitals
Intruder

Enter Patient Name

Patient Name :-

Continue

[Back](#)

Image Gallery



Healthcare Cloud Main Page

[Home Page](#) [Healthcare Cloud](#) [Doctor](#) [Patient](#) [Hospitals](#) [Intruder](#)

Protecting Personal Healthcare Record Using Blockchain and Federated Learning Technologies



HC Menu

- [View All Patients and Authorize](#)
- [View All Doctors and Authorize](#)
- [View All Patients Data](#)
- [View Patient Data Access Request and Authorize](#)
- [View All Intruders Details](#)
- [View Personal Healthcare Records by Blockchain](#)
- [View Patient Recovered Details](#)
- [View No.Of Same Symptoms In Chart](#)
- [View No.Of Patients Referred Same Doctor In Chart](#)

Welcome to Healthcare Cloud Main

All Patients Cloudlet Data

[Home Page](#) [Cloudlet](#) [Doctor](#) [Patient](#) [Hospitals](#) [Intruder](#)

Protecting Personal Healthcare Record Using Blockchain and Federated Learning Technologies



Cloudlet Menu

- [Cloudlet Main](#)
- [Log Out](#)

View All Patients Cloudlet Data

Sl.No.	Report Id	Patient Name	Report Collected Date	View Details
1	1	Omkar	11/11/2022 12:14:03	Click Here
2	2	Omkar	11/11/2022 12:24:08	Click Here
3	3	Raju	11/11/2022 12:33:51	Click Here
4	4	Raju	11/11/2022 12:38:06	Click Here
5	5	Maujanath	11/11/2022 16:13:03	Click Here

Patient Data Access Request and Authorize

Protecting Personal Healthcare Record Using Blockchain and Federated Learning Technologies



Cloudlet Menu

View Patient Data Access Request and Authorize

Cloudlet Main
Log Out

Id	Patient Name	Requested Date	Status
1	Omkar	06/11/2022 13:57:58	Permitted
2	Rakesh	06/11/2022 15:08:05	Permitted
3	Dore	06/11/2022 15:17:52	Permitted
4	Raju	11/11/2022 12:34:44	Permitted
5	Manjunath	11/11/2022 16:13:37	Permitted

Patient Healthcare Records by Blockchain

HC Menu

View personal healthcare records by blockchain

HC Main
Log Out

Disease Name BlockChain--->: H1N1.txt

Disease Report Hash Code --->:51ad9f5d63ee3771baecdb763cefe84e9da064b

Patient Image :-	
Patient Name :-	Omkar
Contact Number :-	9535866270
E-mail :-	omkarsb99@gmail.com
Patient Address :-	#34,pipeline road,vijayanagar
Pulses :-	102
ECG :-	Normal
Symptoms :-	Vomiting
Symptoms File Name :-	H1N1.txt
File Content :-	<p>It is an orthomyxovirus that contains the glycoproteins haemagglutinin and neuraminidase. For this reason, they are described as H1N1, H2N2 etc. depending on the type of H or N antigens they express with metabolic synergy. Haemagglutinin causes red blood cells to clump together and binds the virus to the infected cell. Neuraminidase are a type of glycoside hydrolase enzyme which help to move the virus particles through the infected cell and assist in budding from the host cell.</p>
Digital Sign (MAC) :-	51ad9f5d63ee3771baecdb763cefe84e9da064b

Symptoms Names vs No. of Patients



Doctor Name vs No. of Patients



CHAPTER-11

CONCLUSION

11.1 CONCLUSION

In this paper, a solution based on blockchain and AI Federated learning has been proposed to protect and share safely the patient's healthcare data. After analysing the potentials of combining blockchain and AI, we realized that the proposed solution would help to build robust models and share them without being compromised. However, the application wasn't developed and tested, yet this solution is indeed promising. In the future, our proposed framework will be implemented and tested in a real-world environment and its effectiveness will be studied more in detail.

11.2 FUTURE SCOPE

1. Global Health ID on Blockchain

- A universal blockchain-based identity can allow seamless treatment globally.

2. AI + Blockchain Integration

- Securely train AI models on encrypted, anonymized patient data across organizations.

3. Decentralized Data Marketplaces

- Patients can monetize their health data for research with full consent.

4. Telemedicine and Remote Monitoring

- Verified health data from IoT devices stored securely on-chain.

5. Regulatory Framework Evolution

- Governments and institutions may adopt blockchain as a compliance standard

6. Cross-border Health Data Exchange

- Secure sharing of patient data across countries for medical tourism or emergencies.

CHAPTER-12

REFERENCES

- [1] Choe, J., Lee, S.M., Do, K.H., Lee, G., Lee, J.G., Lee, S.M. and Seo, J.B., 2019. Deep learning–based image conversion of CT reconstruction kernels improves radiomics reproducibility for pulmonary nodules or masses. *Radiology*, 292(2), pp.365-373.
- [2] Kermany, D.S., Goldbaum, M., Cai, W., Valentim, C.C., Liang, H., Baxter, S.L., McKeown, A., Yang, G., Wu, X., Yan, F. and Dong, J., 2018. Identifying medical diagnoses and treatable diseases by image- based deep learning. *Cell*, 172(5), pp.1122-1131.
- [3] Negassi, M., Suarez-Ibarrola, R., Hein, S., Miernik, A. and Reiterer, A., 2020. Application of artificial neural networks for automated analysis of cystoscopic images: a review of the current status and future prospects. *World Journal of Urology*, pp.1-10.
- [4] Gulshan, V., Peng, L., Coram, M., Stumpe, M.C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J. and Kim, R., 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22), pp.2402-2410.
- [5] Tandon, A., Dhir, A., Islam, N. and Mäntymäki, M., 2020. Blockchain in healthcare: A systematic literature review, synthesizing framework and future research agenda. *Computers in Industry*, 122, p.103290.
- [6] Yang, Q., Liu, Y., Chen, T. and Tong, Y., 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), pp.1-19.
- [7] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R. and d'Oliveira, R.G., 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*.
- [8] Li, W., Milletari, F., Xu, D., Rieke, N., Hancox, J., Zhu, W., Baust, M., Cheng, Y., Ourselin, S., Cardoso, M.J. and Feng, A., 2019, October. Privacy-preserving federated brain tumour

segmentation. In International Workshop on Machine Learning in Medical Imaging (pp. 133-141). Springer, Cham.

[9] Sheller, M.J., Reina, G.A., Edwards, B., Martin, J. and Bakas, S., 2018, September. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In International MICCAI Brainlesion Workshop (pp. 92-104). Springer, Cham.

[10] Sekeroglu, B. and Ozsahin, I., 2020. <? covid19?> Detection of COVID-19 from Chest X-Ray Images Using Convolutional Neural Networks. SLAS TECHNOLOGY: Translating Life Sciences Innovation, p.2472630320958376.

[11] Jain, R., Gupta, M., Taneja, S. and Hemanth, D.J., 2020. Deep learning based detection and analysis of COVID-19 on chest X-ray images. Applied Intelligence, pp.1-11.

[12] Ozturk, T., Talo, M., Yildirim, E.A., Baloglu, U.B., Yildirim, O. and Acharya, U.R., 2020. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Computers in Biology and Medicine, p.103792.

[13] Patel, V., 2019. A framework for secure and decentralized sharing of medical imaging data via blockchain consensus. Health informatics journal, 25(4), pp.1398-1411.

[14] Linn, L.A. and Koo, M.B., 2016. Blockchain for health data and its potential use in health it and health care related research. In ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States: ONC/NIST (pp. 1-10).

[15] Ozturk, T., Talo, M., Yildirim, E.A., Baloglu, U.B., Yildirim, O. and Acharya, U.R., 2020. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Computers in Biology and Medicine, p.103792.

[16] Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H., Albarqouni, S., Bakas, S., Galtier, M.N., Landman, B., Maier-Hein, K. and Ourselin, S., 2020. The future of digital health with federated learning. arXiv preprint arXiv:2003.08119.

[17] <https://www.zuehlke.com/en/insights/federated-learning-healthcare-ai-that-doesnt-risk-patient-data-privacy>, extracted on 17/11/2020.

