APPLYING FEDERATED LEARNING WITH SMART CONTRACTS IN HEALTHCARE

PROJECT SUPERVISOR

Dr. Farrukh Shahid

PROJECT CO-SUPERVISOR

Mr. Shahbaz Siddiqui

PROJECT TEAM

Sarmad Jamal	K191116
Khizer Jilani	K191057
Mansoor Butt	K191114

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering.

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

July 2023

Project Supervisor	Dr. Farrukh Shahid
Project Team	Sarmad Jamal K19-1116 Khizer Jilani K19-1057 Mansoor Butt K19-1114
Submission Date	May 2, 2023

Mr. Supervisor Name _____

Supervisor

Mr. Co-Supervisor Name _____

Co-Supervisor

Dr. Abdul Aziz _____

Head of Department

FAST SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

Acknowledgement

We would like to express our deep appreciation to our supervisor, Dr. Farrukh Shahid, whose invaluable support, constructive criticism, and unwavering encouragement were instrumental in the development and refinement of this project. His expert guidance and insights proved to be an invaluable asset, and we could not have completed this project without his guidance.

We also extend our sincere gratitude to our co Supervisor Mr. Shahbaz Siddiqui, particularly for their guidance and mentorship throughout the project. Their expertise and support helped us navigate various challenges and develop an optimized, efficient solution that met our project's objectives.

Furthermore, we would like to acknowledge the financial and emotional support of our parents. Their unwavering belief in our abilities and moral support helped us remain focused and motivated throughout the project's duration.

In conclusion, we express our heartfelt appreciation to all those who contributed to this project's success, and we hope that this work will prove to be a valuable contribution to the academic community.

Abstract

This FYP project aims to address the problem of the lack of medical data sharing in healthcare, which is a major obstacle in integrating artificial intelligence (AI) technology. The reason behind the data-sharing hurdle is that medical data is highly confidential, and hospitals are reluctant to share it due to privacy concerns. As a result, machine learning (ML) models cannot be trained, and progress in this field is stalled. To overcome this issue, this project proposes an alternative approach, which involves model sharing and federated learning. The proposed system consists of two entities, the Super user and hospitals. The Super user requests hospitals to participate in federated learning, and registered hospitals are granted access to global model files through IPFS hashes. These hashes are stored on the block chain via smart contracts. The technologies used in this project include Solidity for smart contract development, React.js for the frontend, and IPFS Piñata for hosting ML model files. This solution provides a viable and secure way to enable AI integration in healthcare while maintaining patient privacy.

Table of Contents

List of Figure

1 Introduction

Blockchain is utilized in healthcare management systems for secure maintenance of electronic health records. This research proposes integrating blockchain and Federated Deep Learning to develop a personalized recommendation system. To protect personal information, distributed storage is suggested. The study aims to make treatment recommendations by comparing patient records with historical data, addressing limitations of existing work. Incorporating federated learning can potentially enhance treatment recommendations.

1.1 Project Scope

Our final year project focuses on the medical sector, where patient data is recorded and used to train a machine learning model. We employ the concept of federated learning, where multiple organizations collaborate to solve a machine learning problem without sharing or transferring data. A central server maintains a global shared model, distributed to each institution. Individual models are trained using localized patient data, and feedback from each center is aggregated by the server to update the global model based on predefined criteria. This iterative process continues until the global model is trained.

1.2 Not in Scope

We will be only targeting one/two disease for the recommendation and it will be running on local environment not globally deployed. It will be on private block chain and only people who are connected will have the feasibility to connect to it

1.3 Problem Statement

The problem statement addresses the challenges of centralized storage for training deep learning models in healthcare, including privacy, ownership, and regulatory concerns. Federated learning (FL) is proposed as a solution, allowing decentralized training without transmitting medical data. Medical institutions train their models locally and periodically share model parameters with a central aggregator server. The server coordinates and aggregates the local models to create a global model, which is then distributed back to the nodes. FL ensures data confidentiality by transmitting only model weights and parameters, offering improved security and enabling collaboration among institutions for healthcare applications such as predicting autism spectrum disorder, mortality, and ICU stay-time.

1.4 Project Objectives

The project aims to develop a secure system for medical data sharing in healthcare using federated learning and model sharing. Objectives include addressing privacy concerns, integrating AI technology, and enabling collaborative ML model training and sharing. Technologies such as Solidity, React.js, IPFS, and blockchain will be utilized to create a robust and scalable solution while ensuring patient data confidentiality.

2 Requirement Analysis

2.1 Functional Requirements

2.1.1 Functional Hierarchy

The super Admin and providers have a specific identity number that identifies their account. The account is accessed via a special account address of web3 provider such as meta mask and for another way of authentication we will use firebase to authenticate the data

As data will be generated from multiple organizations, data must be accessible and consistent at all times. The integrity of the data must not be compromised.

Features

We have Implemented a user-friendly application user interface using React Js technology. It provides an interactive and responsive UI with adequate performance. The navigation from one page to another is explicitly shown.

The project included the following features with respect to super admin and providers (Healthcare institutions).

Login: Login page for super Admin and provider

Signup: Signup page for provider

functionalities related to provider

View Form: Doctor can view patient available respective data

Fill form: Receptionist will submit the form of the patient

Request a particular doctor: Receptionist can request a particular doctor

Take part in training: provider will have a functionality to train its model

functionalities related to super Admin

Retrieve model: super Admin can retrieve hospital model and make it interact with the global model

Aggregation of model: super admin will aggregate the model based on the weights of the different models

2.2 Use Cases

2.2.1 Form Submission

[Use Case Diagram]

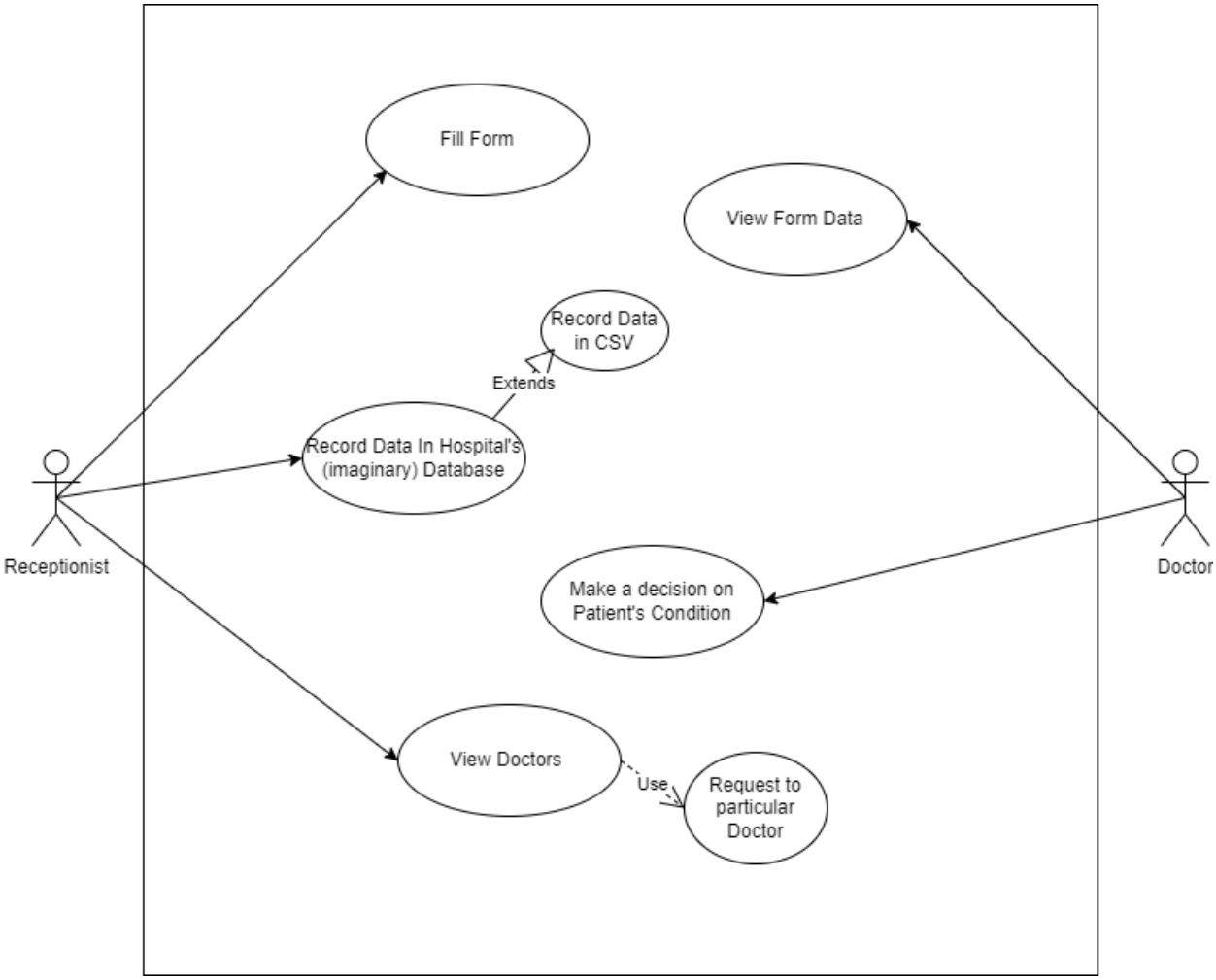


Figure 1

Form Submission		
Use case Id:	1	
Actors:	Receptionist, Doctor	
Feature:	Submission of form	
Pre-condition:	Receptionist must be logged in and have a list of doctors Doctor should be authenticated and have a verified account given by the provider	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Fill the form of the patient	Fill input fields provided on the user interface

		and submit it
2.	Doctor will make a decision on the given form data and patient condition	A form view option will be visible to the doctor and he can view the patient data on the form and will make a decision accordingly for the treatment
Alternate Scenarios:		
None		
Post Conditions		
Step#	**Description**	
1	A final decision for the patient treatment will be taken and if the patient agrees to take part in training aggregate weights will be updated on the global server	
Use Case Cross referenced	None	

Table 1

2.2.2 Hospital/Super user Use Case

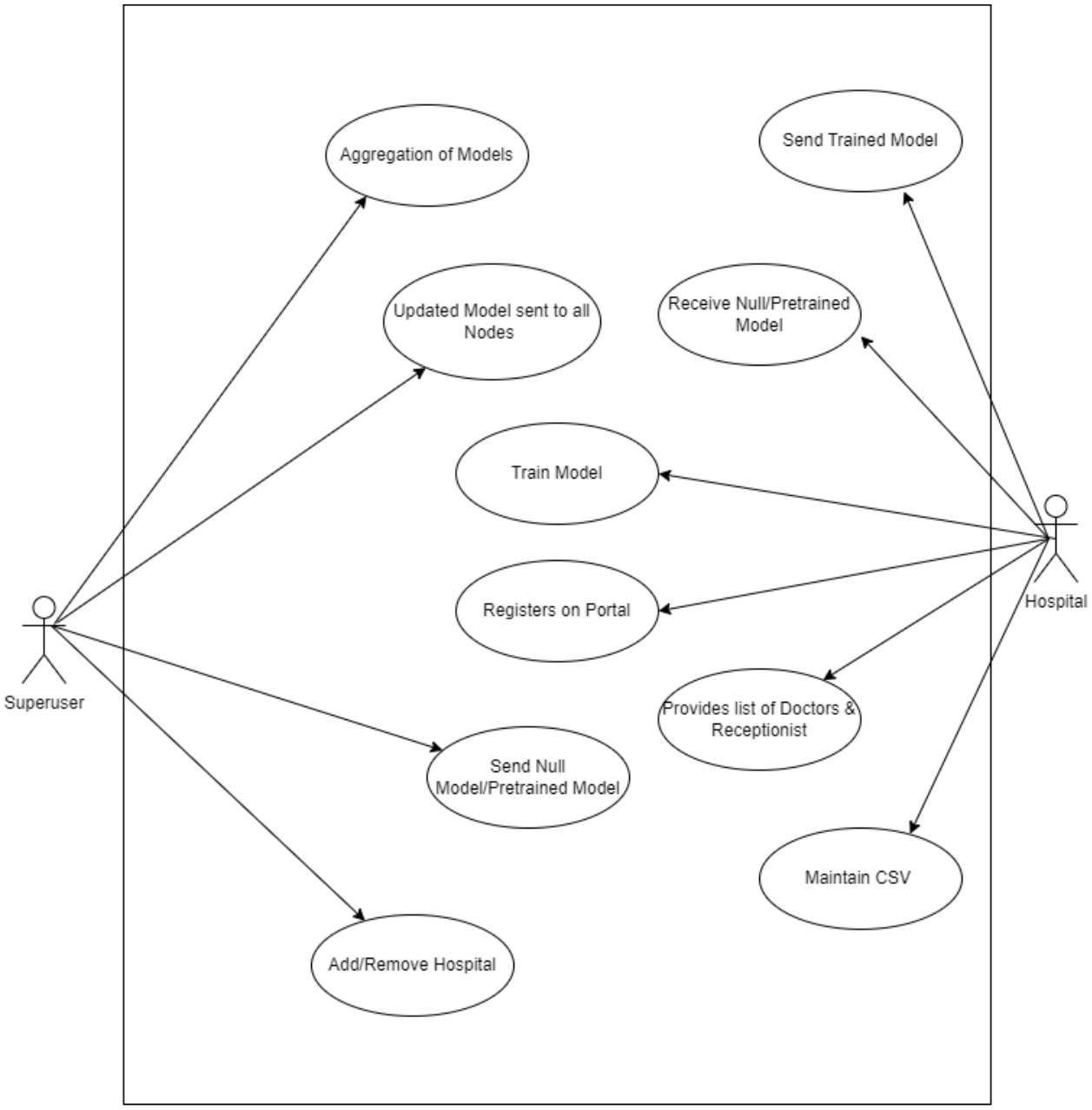


Figure 2

[Use Case Description]

Model training	
Use case Id:	2
Actors:	Super User,Hospital
Feature:	Training of Model
Pre-condition:	SuperUser must be logged in and have access to global server model and have weights passed to it by the hospital

Scenarios		
Step#	**Action**	**Software Reaction**
1.	Hospital will register itself on the portal	Hospital will get access to take part in training of model and interacting with our system
2.	Superuser can add or remove provider from the system	Hospitals with the penalty charges or the hospital which are sucessfully registered will act accordingly
Alternate Scenarios:		
None		
Post Conditions		
Step#	**Description**	
1	Trained model on the recent data will be passed to the hospital and all the nodes will be updated	
Use Case Cross referenced	1	

Table 2

2.3 Non-Functional Requirements

2.3.1 Performance Requirement

The System shall be based on web and has to be run from a web server. The system shall take initial load time depending on internet connection strength which also depends on the media from which the product is run. The performance shall depend upon hardware components of the client/customer

2.3.2 Safety Requirement

The software is completely environmentally friendly and does not cause any safety violations. The web page will have a flexible font that can be zoomed so as to not over constrain the eyes.

2.3.3 Security Requirement

Data Transfer

- The system shall use secure sockets in all transactions that include any confidential customer information.
- The system shall automatically log out all customers after a period of inactivity.
- The system shall confirm all transactions with the customer's web browser.
- The system shall not leave any cookies on the customer's computer containing the user's password.
- The system shall not leave any cookies on the customer's computer containing any of the user's confidential information.

Data Storage

- The customer's web browser shall never display a customer's password. It shall always be echoed with special characters representing typed characters.
- The customer's web browser shall never display a customer's credit card number after retrieving from the database. It shall always be shown with just the last 4 digits of the credit card number.
- The system's back-end servers shall never display a customer's password. The customer's password may be reset but never shown.
- The system's back-end servers shall only be accessible to authenticated administrators.
- The system's back-end databases shall be encrypted.

2.3.4 User Documentation

The software is accompanied by the following materials for further help:

- Software Requirement Document
- Software Design Specification Document

3 Design Details

3.1 Design Extension, Interface & Data Management

3.1.1 Future System Extension or Enhancement

The system will be built using the latest edition of framework and incase a new update occurs in technology we must be ready. Stay updated by tracking orders with customized alerts and resolve issues proactively. Track real-time, Optimize routes and schedules and resource allocation in one centralized view.

3.1.2 User Interface Paradigm

The user interface is very reliable and understandable to the user, the interface designed in such a way that mostly any expertise of people can easily learn and understand that what happens if I click this option, interface is very effective and efficient to increase the usability of users.

3.1.3 Data Management

Main Data which are models will be stored on the IPFS and the relevant hashes of the model will be stored on smart contracts. Hospital will upload the model on web3 interface then store the model on the IPFS which will be use to apply federated leaning through aggregating of all these models.
Form typically collects the following data:
Patient Name
Fever
Blood Pressure
Heart Rate
Platelets
Blood Group
Weight
Age
ETC

4 Implementation Details

4.1 System Architecture

Hospital can request the model from the Global server then download the model. After downloading the model, they will be training it on their own data. After the training is completed, the model will be sent back to the global server this process will be repeated by all the hospitals on the network who are taking part in this training after all the models are received by the global server it will then aggregate all these models through ensemble learning and a new model with combined intelligence of all the aggregated model will be formed This new model will be delegated back to all the hospital.

4.2 System Level Architecture

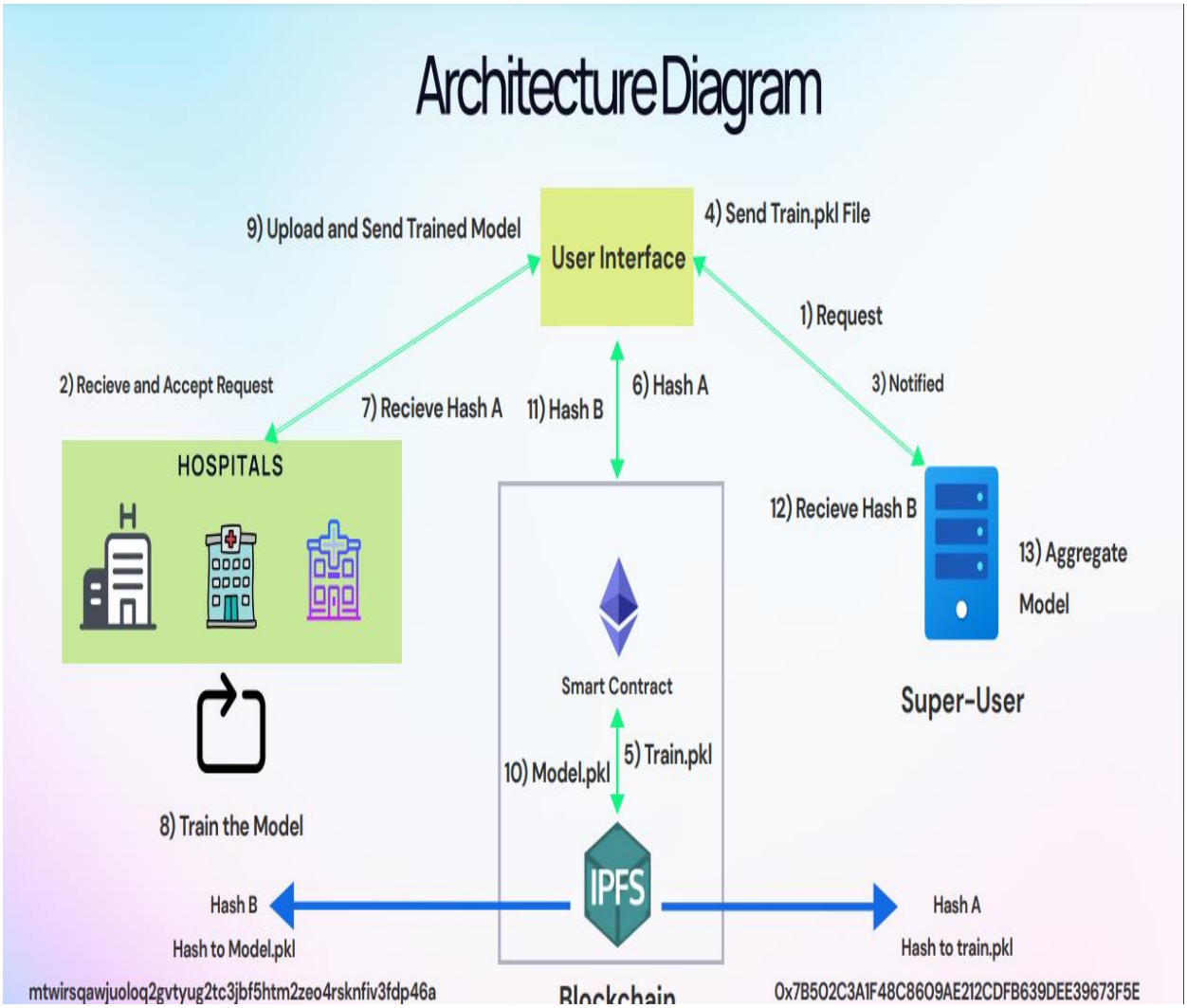
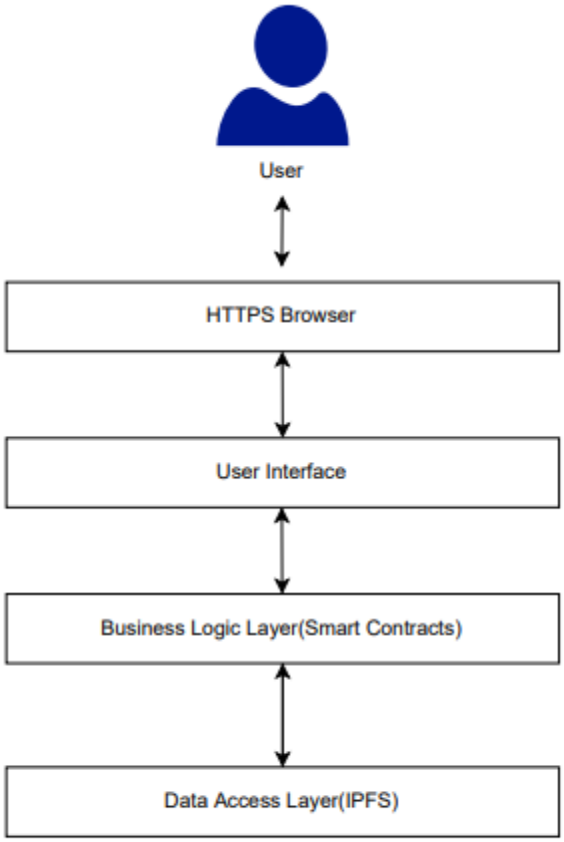


Figure 12

High Level System Architecture

4.3 Software Architecture



Layered Software Architecture
Figure 13

4.4 Development Tools Used

The development tools and software technologies used are as follows:

- React JS
- Node JS
- Express JS
- Git version controlling
- PyCharm
- Visual Studio Code
- Postman
- Thunder Client
- Solidity
- Remix
- EtherJS
- Web3
- Hardhat
- MetaMask
- Ganache

5 Testing and Evaluation

The purpose of this document is to provide a comprehensive outline of the validation and testing procedures necessary for ensuring the effectiveness and proper functionality of our project. The primary objective of this endeavor is to ascertain whether the project satisfies its requirements. In order to achieve this goal, three distinct types of testing willbe conducted: unit testing, performance testing, and system testing. Through these rigorous testing procedures, the entire system will be thoroughly scrutinized to guaranteeits efficient operation and robust performance. It is hoped that the results of these tests will provide valuable insights that will enable us to improve and optimize the project, thus enhancing its overall efficacy and value.

5.1 Environmental Needs

Following are the hardware assumptions for using the system

- **I3** core processor or above
- **4 GB** RAM or above
- **256 GB** SSD space
- Keyboard and Mouse
- Internet Browser (any)
- English language supported only

5.2 Validation Testing (Testing Approach)

Unit Testing: The main objective is to test individual components of a system aretested in isolation to ensure that they are working properly. This helps to identifyand fix any defects or bugs in the system's components before they are integrated.

Integration Testing: Testing all integrated modules to verify that the combined functionality after integration is in a workable state. The objective of this testingis to ensure that all the integrated modules are working properly and are producing the desired output.

System Testing: The entire system is tested as per the requirements. Overallrequirement specifications and all the combined parts of a system are tested.

Usability Testing: The objective of this testing is to ensure that the system is easyto use, user-friendly, and meets the user's expectations. It involves testing the system's navigation, interface, functionality, and user experience to identify any usability issues and to provide recommendations for improvement.

Performance Testing: The objective of this testing is to ensure the system worksunder heavy traffic that includes large size video length processing.