

SMART TENDER CONTRACT MANAGEMENT SYSTEM USING BLOCK CHAIN

Submitted in partial fulfillment of the requirements for
the award of Bachelor of Engineering degree in
Computer Science and Engineering

by

KUNTAMUKALA SASIDHAR (40110654)
ANEGONDI ASHOK RAJU (40110090)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Category – 1 University by UGC

**Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICET
JEPPIAAR NAGAR, RAJIV GANDHISALAI, CHENNAI – 600119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **KUNTAMUKALA SASIDHAR (40110654)** and **ANEGONDI ASHOK RAJU (40110090)** who carried out the project entitled "**Smart Tender Contract Management System Using Blockchain**" under my supervision from November 2023 to April 2024.

Internal guide

Dr. E. MURALI, M.TECH., Ph.D.

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA Ph.D.
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

Submitted for Viva voce Examination held on

13/4/2024

Internal Examiner

External Examiner

DECLARATION

I **ANEGONDI ASHOK RAJU** here by declare that the Project Report entitled "**SMART TENDER CONTRACT MANAGEMENT SYSTEM USING BLOCKCHAIN**" is done by me under the guidance of **Dr. E. MURALI, MTECH., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 13-04-2024

PLACE: CHENNAI

A. Ashok

SIGNATURE OF THE CANDIDATE

DECLARATION

I **KUNTAMUKALA SASIDHAR** hereby declare that the Project Report entitled “**SMART TENDER CONTRACT MANAGEMENT SYSTEM USING BLOCKCHAIN**” is done by me under the guidance of **Dr. E. MURALI, MTECH., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. L. Lakshmanan, M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. E. Murali, MTECH., Ph.D., Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Governments and public sector entities around the world are actively exploring new ways to keep up with technological advancements to achieve smart governance, work efficiency, and cost optimization. Blockchain technology is an example of such technology that has been attracting the attention of Governments across the globe in recent years. Enhanced security, improved traceability and lowest cost infrastructure empower the blockchain to penetrate various domains. Generally, governments release tenders to some third-party organizations for different projects. During this process, different competitor tries to eavesdrop the tender values of others to win the tender. The corrupt government officials also charge high bribe to pass the in favour of some particular third party. In this paper, we presented a secure and transparent framework for government tenders using blockchain. Blockchain is used as a secure and immutable data structure to store the government records that are highly susceptible to tampering. This work aims to create a transparent and secure edge computing infrastructure for the work-flow in government tenders to implement government schemes and policies by limiting human supervision to the minimal.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
1.	INTRODUCTION	1
1.1	AIM	1
1.2	PURPOSE OF PROJECT	1
1.3	PROJECT SCOPE	1
2.	LITERATURE SURVEY	3
3.	MATERIALS METHOD AND ALGORITHM	6
3.1	EXISTING SYSTEM	6
3.2	PROPOSED SYSTEM	6
3.3	REQUIRMENT SPECIFICATION	7
3.3.1	INTRODUCTION	7
3.3.2	HARDWARE AND SOFTWARE SPECIFICATION	8
3.1.1.1	INTRODUCTION TO JAVA	9
3.1.1.1.2	WORKING OF JAVA	10
3.1.1.1.4	THE JAVA PLATFORM	12
3.1.1.2	INTRODUCTION TO IOT	15
4.	SYSTEM DESIGN & IMPLEMENTATION CONSTRAINTS	22
4.1	CONSTRAINTS IN ANALYSIS	22
4.2	CONSTRAINTS IN DESIGN	22
4.3	CONSTRAINTS IN IMPLEMENTATION	22
4.4	OTHER NON-FUNCTIONAL REQUIRMENTS	23
4.5	SYSTEM DESIGN	23
4.6	MODULES	24

5.	SCHEMATIC REPRESENTAION OF UML DIAGRAMS	27
5.1	ARCHITECTURE DIAGRAM	27
5.2	SEQUENCE DIAGRAM	28
5.3	USE CASE DIAGRAM	29
5.4	ACTIVITY DIAGRAM	30
5.5	COLLABORATION DIAGRAM	30
5.6	DATA FLOW DIAGRAM	31
6.	RESULTS AND DISCUSSIONS	34
7.	CONCLUSION AND REFERENCES	35
A	CONCLUSION	35
B	REFERENCES	36
C	APPENDIX	38
D	RESEARCH PAPER	48

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
3.1	JAVA VIRTUAL MACHINE	11
3.2	JAVA COMPILER	12
3.3	JAVA HARDWARE PLATFORM	13
3.4	APPLICATIONS OF IOT	16
5.1	ARCHITECHTURE DIAGRAM	27
5.2	SEQUENCE DIAGRAM	28
5.3	USE CASE DIAGRAM	29
5.4	ACTIVITY DIAGRAM	30
5.5	COLLABORATION DIAGRAM	30
5.6	DATA FLOW DIAGRAM	31
6.1	HOME PAGE	34
6.2	ADMIN DASHBOARD	34

CHAPTER 1

INTRODUCTION

1.INTRODUCTION TO THE PROJECT:

Governments and public sector entities around the world are actively exploring new ways to keep up with technological advancements to achieve smart governance, work efficiency, and cost optimization. Block chain technology is an example of such technology that has been attracting the attention of Governments across the globe in recent years. Enhanced security, improved traceability and lowest cost infrastructure governments release tenders to some third-party organizations for different projects. The corrupt government officials also charge high bribe to pass the tender in favor of some particular third party. In this paper, we presented a secure and transparent framework for government tenders using block chain. Block chain is used as a secure and immutable data structure to store the government records that are highly susceptible to tampering. This work aims to create a transparent and secure edge computing infrastructure for the work-flow in government tenders to implement government schemes and policies by limiting human supervision to the minimal.

1.1 AIM:

The main aim of this project is used to transparent and secure edge computing infrastructure for the work-flow in government tenders to implement government schemes and policies by limiting human supervision to the minimal.

1.2 PURPOSE:

The main aim of this project is used to transparent and secure edge computing infrastructure for the work-flow in government tenders to implement government schemes and policies by limiting human supervision to the minimal.

1.3 PROJECT SCOPE:

Government announces new projects, invites vendors to give quotations/tenders and approves the best candidate. The vendor is given the project and a Government officer is appointed to keep track of the status of project.

A timeline is defined for project and officer to keep a check on progress and to get it complete before deadline. Government officer often visit the site, to take an overview of the progress and report it to the officer. We are developing a very comprehensive system for this business process which will manage each and every aspect starting from vendor registration and tender submission and up to the completion of the project.

1.4 PRODUCT PERSPECTIVE:

A block chain is a public ledger or distributed database of records or transactions that are shared between multiple parties within the network. Keeping this in mind a block chain can be divided into two categories Permission- less (Public) and Permission (Private) based on whether participants are allowed to or not, join the network freely.

The smart contract is coded in the solidity language. The results of the smart contract are observed after deploying the smart contract in the Rink by test network of the ethereum block chain platform.

1.5 SYSTEM FEATURES:

The smart contract is coded in the solidity language. The results of the smart contract are observed after deploying the smart contract in the Rinke by test network of the Ethereum blockchain platform.

CHAPTER 2

LITERATURE REVIEW

Wazie M. Abdulkawi 1, *, Khaled Issa Year of Publish (2020):

There is a growing interest in chipless radio-frequency identification (RFID) technology for a number of Internet of things (IoT) applications. This is due to its advantages of being of low-cost, low-power, and fully printable. In addition, it enjoys ease of implementation. In this paper, we present a novel, compact, chipless radio-frequency identification (RFID) tag that can be read with either vertical or horizontal polarization within its frequency bandwidth. This increases the sturdiness and detection ability of the RFID system. In addition, the difference between the vertical and horizontal responses can be used for tag identification. The proposed tag uses strip length variations to double the coding capacity and thereby reduce the overall size by almost 50%. It has a coding capacity of 20 bits in the operating bandwidth 3 GHz–7.5 GHz, and its spatial density is approximately 11 bits/cm². The proposed tag has a 4.44 bits/GHz spectral capacity, 2.44 bits/cm²/GHz encoding capacity, a spatial density at the center frequency of 358.33 bits/_2, and an encoding capacity at the center frequency of 79.63 bits/_2/GHz. A prototype is fabricated and experimentally tested at a distance of 10 cm from the RFID reader system. Then, we compare the measured results with the simulations. The simulated results are in reasonable agreement with the simulated ones.

Vikas hassija 1, Vinay chamola2, Vikas Saxena (2019):

Internet of things (IoT) is the next era of communication. Using IoT, physical objects can be empowered to create, receive and exchange data in a seamless manner. Various IoT applications focus automating different tasks and are trying to empower the inanimate physical objects to act without any human intervention. The existing and upcoming IoT applications are highly promising to increase the level comfort, efficiency, and automation for the users. To be able to implement such a world in an ever-growing fashion requires high security, privacy,

authentication, and recovery from attacks. In this regard, it is imperative to make the required changes in the architecture of IoT applications for achieving end-to-end secure IoT environments. In this paper, a detailed review of the security-related challenges and sources of threat in IoT applications is presented. After discussing the security issues, various emerging and existing technologies focused on achieving a high degree of trust in IoT applications are discussed. Four different technologies: Blockchain, fog computing, edge computing, and machine learning to increase the level of security in IoT are discussed.

Neşe Ganiç Sağlam^{1*}, A. Zerrin Yılmaz², Stefano P. Corgnati³(2019):

Cost-optimal and nearly-zero energy building (NZEB) levels are two interrelated concepts identified for upgrading energy performance of buildings in Europe. In parallel, many research activities on retrofitting existing buildings in Turkey follow the methodology framework introduced by the European Commission. However, in Turkey, there is a process called “urban transformation” due to the earthquake risk, but the practice is based on new construction after demolishment of existing buildings. Especially in Istanbul, this process has been conducted rapidly, This specific aspect requires assessment of cost-optimal retrofit analyses considering the remaining lifespan of analysed buildings. This study presents a cost- optimality assessment method for retrofitting towards NZEB in Turkey. The method integrates payback period and investment cost assessment to the methodology in order to obtain whole picture for retrofit alternatives. In the paper, suggested method is applied to a reference residential building in Istanbul. After the initial cost-optimal analyses, payback periods and initial investment costs for selected retrofit packages are assessed considering the future lifespan of the building. Together with these, possible subsidy opportunities are also investigated. Results show that, if the expected future lifespan is higher than 10 years, retrofit actions achieving 56.2kWh/m²y primary energy consumption level are considerable. Subsidies are beneficial to obtain reasonable initial investment costs.

Lei Cui ¹, Zonghua Zhang ¹, Nan Gao ¹, Zhaozong Meng (2019):

Radio Frequency Identification (RFID) sensors, integrating the features of Wireless Information and Power Transfer (WIPT), object identification and

client sensing capabilities, have been considered a new paradigm of sensing and communication for the futuristic information systems. RFID sensor tags featuring contactless sensing, wireless information transfer, wireless powered, light weight, non-line-of-sight transmission, flexible and paste able are a critical technology for future Internet-of-Things (IoT) applications, such as manufacturing, logistics, healthcare, agriculture and food. They have attracted numerous research resorts due to their innovative potential in the various application fields. However, there has been a gap between the in-lab investigations and the practical IoT application scenarios, which has motivated this survey of this research to identify the promising enabling techniques and the underlying challenges. This study aims to provide an exhaustive review on the state-of-art RFID sensor technologies from the system implementation perspective by focusing on the fundamental RF energy harvesting theories, the recent technical progresses and commercial solutions, innovative applications and some RFID sensor based IoT solutions, identify the underlying technological challenges at the time being, and give the future research trends and promising application fields in the rich sensing applications of the forthcoming IoT era.

CHAPTER 3

MATERIALS METHOD ALGORITHM

3.1 EXISTING SYSTEM

The allocation of tenders from government lenders to constructors. Initially, government lenders and constructors enter into the block chain network to construct a secure edge computing infrastructure. The government lender shares the tender details with all the relevant constructors. Further, double auctioning takes place between the constructors and government lenders. The target of the model is to efficiently handle the government tender process securely. The system mainly consists of three types of entities: government officials, external parties like construction companies or individual constructors, and banks.

Problem Definition:

The problem for dynamic pricing which is also aimed at enhancing the overall profit for the constructors and the government lenders. The government lenders get the best resources at the lowest prices, and the constructors get the best tenders matching to the resources they have.

3.2 PROPOSED SYSTEM

Government announces new projects, invites vendors to give quotations/tenders and approves the best candidate. The vendor is given the project and a Government officer is appointed to keep track of the status of project. A timeline is defined for project and officer to keep a check on progress and to get it complete before deadline. Government officer often visit the site, to take an overview of the progress and report it to the officer. We are developing a very comprehensive system for this business process which will manage each and every aspect starting from vendor registration and tender submission and up to the completion of the project.

Advantage:

A block chain is a public ledger or distributed database of records or transactions that are shared between multiple parties within the network. Keeping this in mind a block chain can be divided into two categories Permission -less (Public) and Permission (Private) based on whether participants are allowed to or not, join the network freely.

The smart contract is coded in the solidity language. The results of the smart contract are observed after deploying the smart contract in the Rink by test network of the ethereum block chain platform.

3.3 REQUIREMENT SPECIFICATIONS**3.3.1 INTRODUCTION:**

There has been a rapid growth in the use of blockchain technology in almost all domains and all parts of the world. Blockchain technology is a highly promising solution that can be employed in the government tender process to enhance the level of security, privacy, transparency, and speed of work. Blockchain can allow all the parties involved in a particular tender to be a part of the same network and to monitor the work-flow step by step [1]. Governments like Georgia, UK, UAE, Australia, China, Japan, and Russia are currently progressing at a rapid pace in adapting blockchain in their day to day functioning. The government of Dubai has an ambitious plan of becoming completely paperless through the widespread implementation of blockchain [2]. Governments of several developing countries like India have also been promoting various projects and policies for adaptation of blockchain technology in recent years [3]. There have been various attempts to implement technology to make government processes paperless and instantaneous, such as online ticketing systems, online issuing of tenders, filing tax returns, etc. Although most of these systems seem and well-implemented, all of them are based on the idea of a central server which has a single point of failure, as hackers can easily hack or disrupt its functioning by attacks, such as DOS,

Slow-loris, SYN Flooding, etc [4]. In most governments, complicated bureaucratic systems often result in highly inefficient work-flow fraught with corruption, mismanagement, and human errors. Some of the governance processes, such as government tenders include malpractices like information leaks, corruption, bribery, etc. Most of the existing electronic services and IT infrastructure have the above-mentioned limitations, however, new technologies such as blockchain have the potential to greatly ameliorate the existing problems [5], [6]. A permissioned blockchain network can provide the necessary transparency to effectively implement government policies for the benefit of the citizens of the country and fix responsibilities in case of abuse of the system.

3.3.2 HARDWARE AND SOFTWARE SPECIFICATION

3.3.2.1 HARDWARE REQUIREMENTS:

- Hard Disk : 250GB and Above
- RAM : 4GB and Above
- Processor : i3 and Above

3.3.2 SOFTWARE REQUIREMENTS:

- Windows 10
- JDK 1.8
- JDK 1.7
- My SQL 5.0
- Tomcat 9

3.1.1 TECHNOLOGIES USED:

- J2EE (JAVA, JSP, JSTL, JSON)
- JavaScript, HTML, CSS
- Spring MVC
- JDBC

3.1.1.1 JAVA

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

3.1.1.1.1 INTRODUCTION TO JAVA:

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (*7) that consisted of the Oak language, an operating system called Green OS, a user interface, and hardware. The name *7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First-Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus

proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention. Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

3.1.1.1.2 **WORKING OF JAVA:**

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main**, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the argv [] of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out**, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

Java consists of two things:

- Programming language
- Platform

3.1.1.1.3 **THE JAVA PROGRAMMING LANGUAGE:**

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

- Simple
- Object-oriented and Distributed

- Interpreted
- Robust
- Secure
- Architecture-neutral
- Portable
- High-performance
- Multithreaded
- Dynamic

The code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Java is 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 codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how it works:

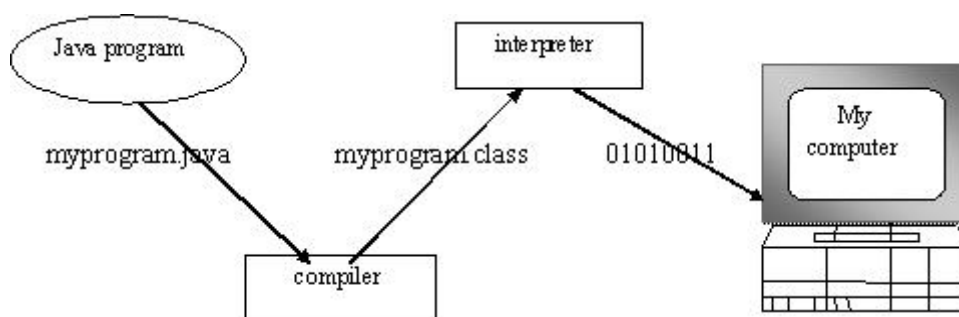


Fig.3.1.Java Virtual Machine

You can think of Java byte codes as the machine code instructions for the **Java Virtual Machine (JVM)**. Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is

an implementation of JVM. That JVM 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 any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. For example, that same Java program can be run on Windows NT, Solaris and Macintosh

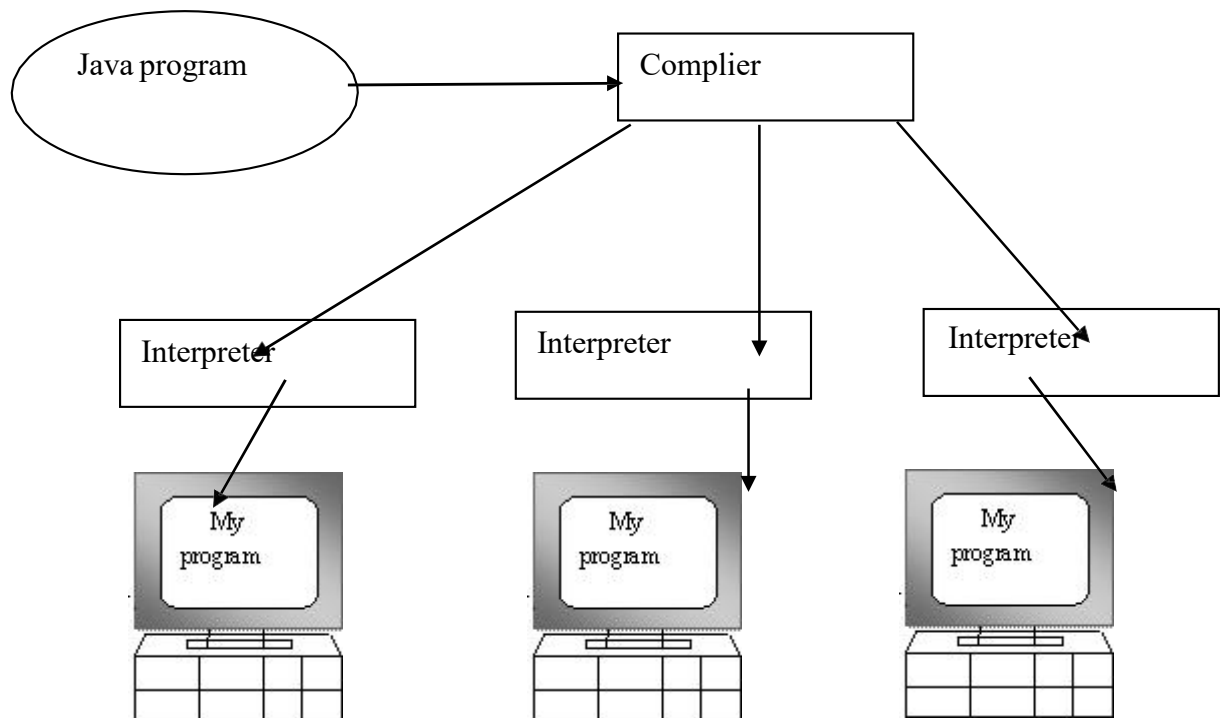


Fig.3.2 JAVA COMPILER

3.1.1.14 THE JAVA PLATFORM:

A platform is the hardware or software environment in which a program runs. 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. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components:

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

You've already been introduced to the JVM. 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 (**packages**) of related components. The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.

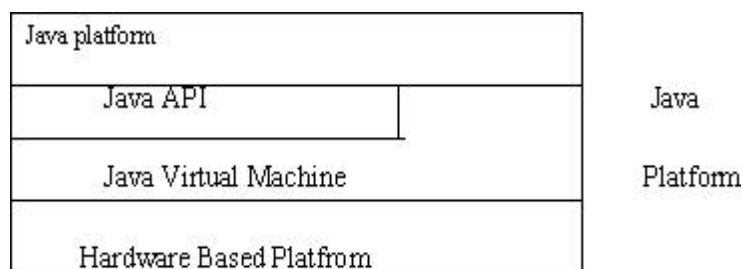


Fig.3.3. Java Hardware Platform

As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, weel-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening portability.

ENVIRONMENT:

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

PRODUCT FEATURES:

Tomcat 3.x (initial release)

- implements the Servlet 2.2 and JSP 1.1 specifications
- servlet reloading
- basic HTTP functionality Tomcat 4.x
- implements the Servlet 2.3 and JSP 1.2 specifications
- servlet container redesigned as Catalina
- JSP engine redesigned as Jasper
- Coyote connector
- Tomcat 5.x
- implements the Servlet 2.4 and JSP 2.0 specifications
- reduced garbage collection, improved performance and scalability
- native Windows and Unix wrappers for platform integration
- faster JSP parsing

HISTORY:

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project

open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

3.1.1.2 INTRODUCTION TO IOT:

IoT refers to an Internet of Things (IoT). Connecting any device (including everything from cell phones, vehicles, home appliances and other wearable embedded with sensors and actuators) with Internet so that these objects can exchange data with each other on a network. It is interesting to note that there is a difference between IoT and the Internet; it is the absence of Human role. The IoT devices can create information about individual's behaviors, analyze it, and take action (IoT is smarter than Internet: D)

A question would arise in your mind that why we are concerned about IoT? Here is the answer that why you should be concerned about IoT. Say for example you are on your way to a meeting; your car could have access to your calendar and already know the best route to take. If the traffic is heavy your car might send a text to the other party notifying them that you will be late. What if your alarm clock wakes up you at 6 a.m. and then notifies your coffee maker to start making coffee for you? Being able to turn the lights on in your house or heating before coming home using your smartphone? Yes, all these things are possible because of IoT.

Smart System and the Internet of the Things are driven by a combination for:

1. Sensors & Actuators
2. Connectivity
3. People & Process

INTEROPERABILITY IN IOT:

- The Internet of Things (IoT) is an incredibly diverse space, encompassing a large variety of hardware form factors and software ecosystems unlike anything we have seen in technology. Smart watches, connected cameras, drones, thermostats, voice-enabled speakers, smart appliances and more they all live together within the IoT.
- The diversity and innovation that excites many IoT fans is a big challenge not just for manufacturers and developers, but also (and most importantly) consumers. Which technology options should be used when designing or deploying IoT devices? How do they keep up with updated or new operating systems? What about new software and connectivity technologies coming up? Those are just some of today's challenges.
- Having a single, unified communication and software framework for the IoT seems like an ideal solution, but the diverse and fast-paced nature of the IoT makes this utopia a big challenge. Diversity in the IoT is not something to be solved, but an aspect that must be embraced and managed.

Application areas for the Internet of Things:

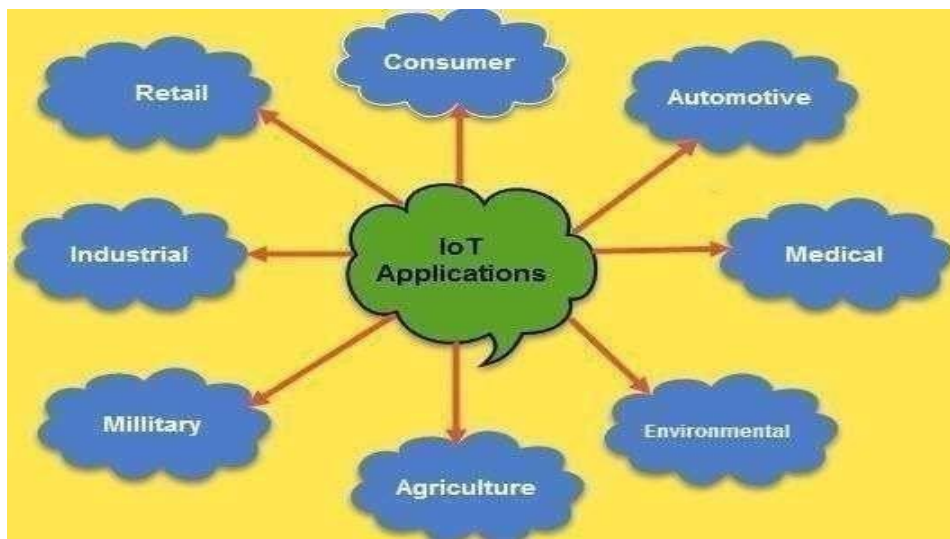


Fig.3.4.APPLICATIONS OF IOT

Smart Home

The concept of Smart Home is brought up to save time, energy and money. With the introduction of Smart Homes, we would be able to switch on air conditioning before reaching home or switch off lights even after leaving home or unlock the doors to friends for temporary access even when you are not at home.

Smart cities

Smart surveillance, automated transportation, smarter energy management systems, water distribution, urban security and environmental monitoring all are examples of internet of things applications for smart cities. IoT will solve major problems faced by the people living in cities like pollution, traffic congestion and shortage of energy supplies etc. By installing sensors and using web applications, citizens can find free available parking slots across the city. Also, the sensors can detect meter tampering issues, general malfunctions and any installation issues in the electricity system.

Wearable's

Wearable devices are installed with sensors and software's which collect data and information about the users. This data is later pre-processed to extract essential insights about user. These devices broadly cover fitness, health and entertainment requirements. The pre-requisite from internet of things technology for wearable applications is to be highly energy efficient or ultra-low power and small sized.

Healthcare

IoT in healthcare is aimed at empowering people to live healthier life and regular checkup by wearing connected devices. The collected data will help in personalized analysis of an individual's health and provide tailor made strategies to combat illness.

Challenges to IoT

In the era of IoT, everything is connected, linked up much more than we see in and around us. IoT is certainly opening door to a lot of opportunities but also to many challenges.

Security changes

Security is a big issue with IoT devices. With billions of devices being connected together over Internet, how can people be sure that their information is secure? These security issues can be of the following kinds

Data Encryption

IoT applications collect tons of data. Data retrieval and processing is integral part of the whole IoT environment. Most of this data is personal and needs to be protected through encryption.

Encryption is widely used on the internet to protect user information being sent between a browser and a server, including passwords, payment information and other personal information that should be considered private. Organizations and individuals use encryption to protect sensitive data stored on computers, servers and mobile devices like phones or tablets.

Data Authentication

After successful encryption of data chances of device itself being hacked still exist. If there is no way to establish the authenticity of the data being communicated to and from an IoT device, security is compromised.

For instance, say you built a temperature sensor for smart homes. Even though you encrypt the data it transfers is there is no way to authenticate the source of data then anyone can make up fake data and send it to your sensor instructing it to cool the room even when its freezing or vice versa.

Side-channel Attacks

Encryption and authentication both in place still leave scope for side channel attacks. Such attacks focus less on the information and more on how that

information is being presented. For instance, if someone can access data like timing information, power consumption or electromagnetic leak, all of this information can be used for side channel attacks.

Privacy Challenges

Then we have the issue of privacy and data sharing. That is because these devices not only collect personal information like users' names and telephone numbers, but can also monitor user activities (e.g., when users are in their houses and what they had for lunch).

Connectivity Challenges—Billions of devices on a centralized server

One of the biggest challenges for IoT in the future is to connect large number of devices and massive amounts of data that all of these devices are going to produce. There will be need to find out a way to store, track, analyze and make sense of the vast amounts of data that will be generated.

Presently, we rely upon centralized, server/client model to authorize, authenticate, and connect several nodes present on the network. This model is sufficient for the number of IoT devices that are currently a part of the ecosystem. However, in the future, when hundreds of billions of devices will join the network, it will be difficult to manage all the data. Moreover, the capability of current cloud servers is so less that it can breakdown if it has to handle large amounts of information.

Compatibility and Longevity Challenges-Extra hardware and software

Different technologies like ZigBee, Z-Wave, Wi-Fi, Bluetooth and, Bluetooth Low Energy (BTLE) are all battling to become the dominant transport mechanism between devices and hubs. This becomes a major source of problems when a lot of devices have to be connected; such dense connectivity requires the deployment of extra hardware and software.

Conversations about the IoT are taking place all over the world as we are trying to understand how this will impact our lives. We are also trying to understand what the many opportunities and challenges are going to be as more

and more devices start to join the IoT. So, all that we can do is educate ourselves about what the IoT is and how it will be after some years.

Introduction Base 64:

Base64 is an encoding algorithm that allows you to transform any characters into an alphabet which consists of Latin letters, digits, plus, and slash. Thanks to it, you can convert Chinese characters, emoji, and even images into a “readable” string, which can be saved or transferred anywhere.

To figuratively understand why Base64 was invented, imagine that during a phone call Alice wants to send an image to Bob. The first problem is that she cannot simply describe how the image looks, because Bob needs an exact copy. In this case, Alice may convert the image into the binary system and dictate to Bob the binary digits (bits), after that he will be able to convert them back to the original image. The second problem is that the tariffs for phone calls are too expensive and dictate each byte as 8 binary digits will last too long. To reduce costs, Alice and Bob agree to use a more efficient data transfer method by using a special alphabet, which replaces every “six digits” with one “letter”.

Advanced Encryption Standard

The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six times faster than triple DES.

A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.

The features of AES are as follows - Symmetric key symmetric block cipher 128-bit data, 128/192/256-bit keys Stronger and faster than Triple-DES

Provide full specification and design details.

Software implementable in C and Java Operation of AES

AES is an iterative rather than Feistel cipher. It is based on 'substitution– permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix -

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

CHAPTER 4

DESIGN AND IMPLEMENTATION CONSTRAINTS

4.1 CONSTRAINTS IN ANALYSIS:

- ◆ Constraints as Informal Text
- ◆ Constraints as Operational Restrictions
- ◆ Constraints Integrated in Existing Model Concepts
- ◆ Constraints as a Separate Concept
- ◆ Constraints Implied by the Model Structure

4.2 CONSTRAINTS IN DESIGN:

- ◆ Determination of the Involved Classes
- ◆ Determination of the Involved Objects
- ◆ Determination of the Involved Actions
- ◆ Determination of the Require Clauses
- ◆ Global actions and Constraint Realization

4.3 CONSTRAINTS IN IMPLEMENTATION:

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore, it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity- relationship modeling and many object-oriented methods. Therefore, it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. A flat relation corresponds with the relation concept of entity- relationship.

4.4 OTHER NON-FUNCTIONAL REQUIREMENTS:

4.4.1 Performance Requirements:

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service;
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

4.4.2 Safety Requirements:

1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
6. Systems with different requirements for safety levels must be separated.
7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

4.5 SYSTEM DESIGN

A blockchain is a public ledger or distributed database of records or transactions that are shared between multiple parties within the network. Keeping this in mind a blockchain can be divided into two categories Permission-less (Public) and Permissioned (Private) based on whether participants are allowed to or not, join

the network freely. A Permissionless blockchain network allows anyone to participate in the transaction validating process by providing them with a unique address through which they can interact with the network like Bitcoin. A Permissioned Blockchain is a closed ecosystem, where users are not allowed to join the network, view the recorded ledger, or issue transactions of their own freely. One of the major drawbacks of Permissionless Blockchain which holds most of the enterprises from employing it is that it requires massive computing power to achieve consensus and the transaction history is visible to everyone.

4.6 MODULES

- Vendor Login and Register
- Government verification of vendors.
- Government Tender Allocation in Vendor.

Tender Details stored in block chain using Smart Contracts

4.6.1 MODULE EXPLANATION:

4.6.1.1 Vendor Login and Register:

Initially Vendor need to create an account in the Government tender Process

Government verification of vendors:

Government announces new projects, invites vendors to give quotations/tenders and approves the best candidate. The vendor is given the project and a Government officer is appointed to keep track of the status of project. A timeline is defined for project and officer to keep a check on progress and to get it complete before deadline. Government officer often visit the site, to take an overview of the progress and report it to the officer.

Government Tender Allocation in Vendor:

We are developing a very comprehensive system for this business process which will manage each and every aspect starting from vendor registration and tender submission and up to the completion of the project. Our system will automate this process and make it really smooth and will also provide a lot of intelligent features which are not possible in manual system. The system will provide an interface to the Government Administrator to define new projects and allocate to officers. The vendors will fill in tenders from the system itself and the tender approval process will also be facilitated by our system.

MODULES

- ☐ Admin Module
- ☐ Company Module

MODULES DESCRIPTION:

Admin:

- Login
- Manage Tender
- List of all the tender
- View / update / delete / Add
- Add Tender
- Update Tender
 - Only a few fields which are not very important and likely to have changed can be updated.
- Ongoing Tender
 - list of all the tenders which are open for bids/proposals
 - list of all the applications received for the tender
 - Admin can update the status of the Application (Accept/ Reject/ Shortlist)
- Past Tenders
 - List of all the tenders which are closed for bids and like ongoing can view applications & update the status

- View Transactions
 - List of tenders & their applications with Manipulation status
 - It will indicate if there were any manipulations done without the consent or knowledge of the Admin.
 - We will explain the manipulation part at the end of the video.

Company:

- Register
- Login

- Company Profile
- Tenders
 - List of all the ongoing tenders which are open for bids
 - Company can upload their proposal and participate.
- Change Password
- Let's go to the admin panel and view the application and change the status
- Let's go back to the Companies panel
- My Application
- List of my application & Status

Tender Details stored in block chain using Smart Contracts:

Most importantly, we will also have an android based tracking module which will be used by officer to keep a live track on the progress by updating. This will assure the management that the photo presented is that of the actual site and not a dummy photo. The overall aim of system is to increase human efficiency and keep the vendors and officers on their toes which should help in on-time completion of Government Projects which is very rare in today's scenarios.

CHAPTER 5

SCHEMATIC REPRESENTAION OF UML DIAGRAMS

ARCHITECTURE DIAGRAM

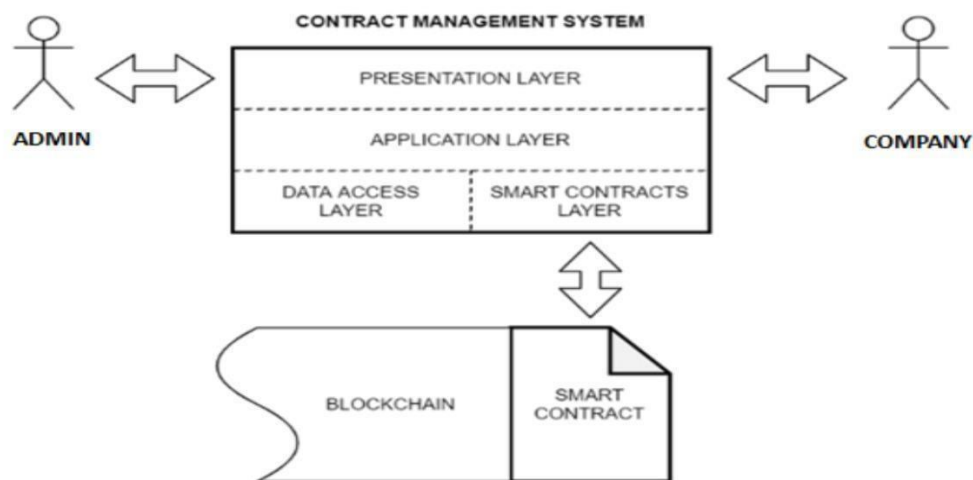


Fig.5.1.Architecture Diagram

5.2 SEQUENCE DIAGRAM:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

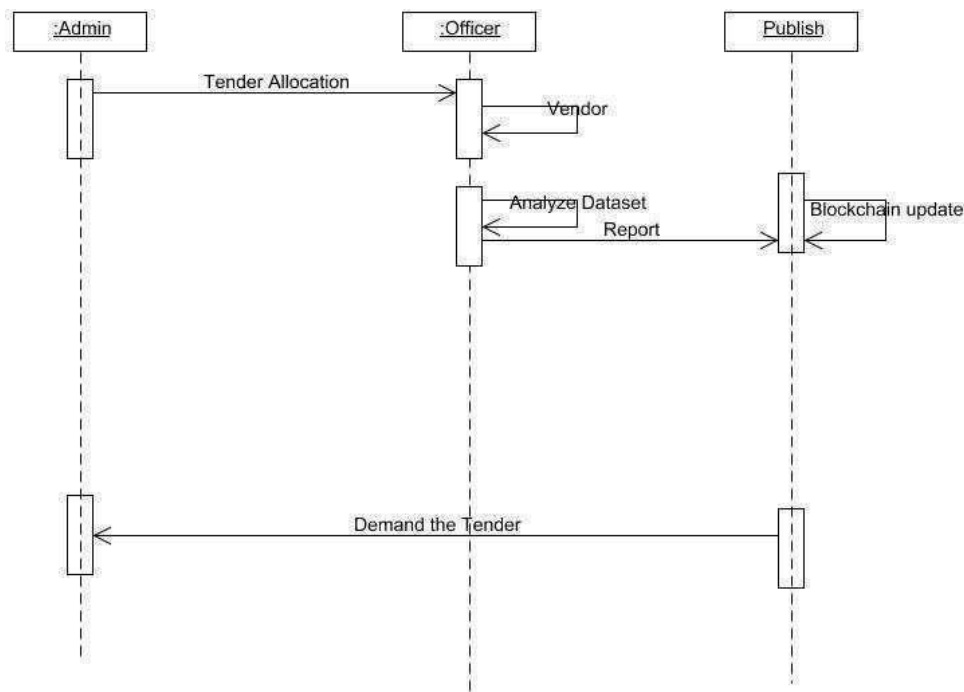


Fig.5.1.Sequence Diagram

5.3 USE CASE DIAGRAM:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

5.3.1 USECASE DIAGRAM:

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

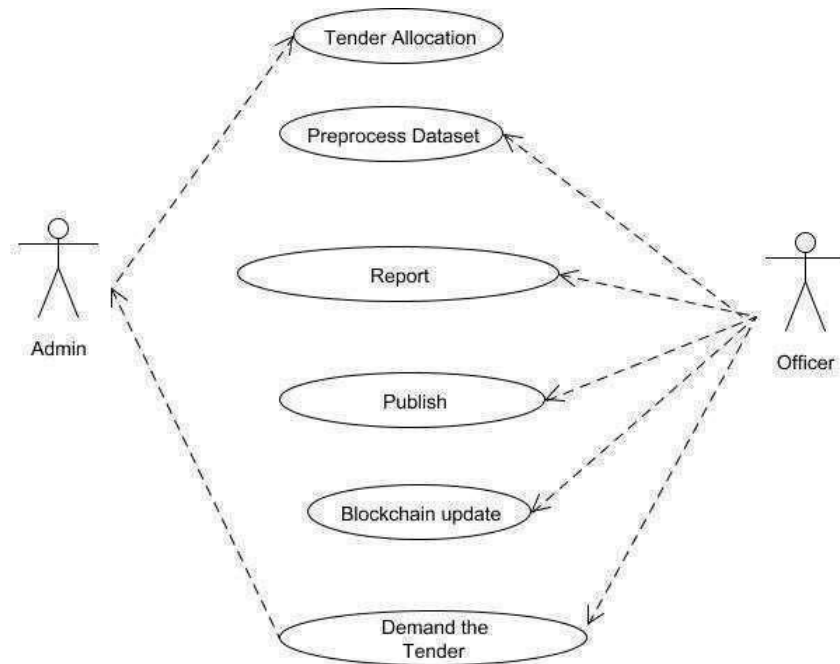


Fig.5.1.Use Case Diagram

5.4 ACTIVITY DIAGRAM:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

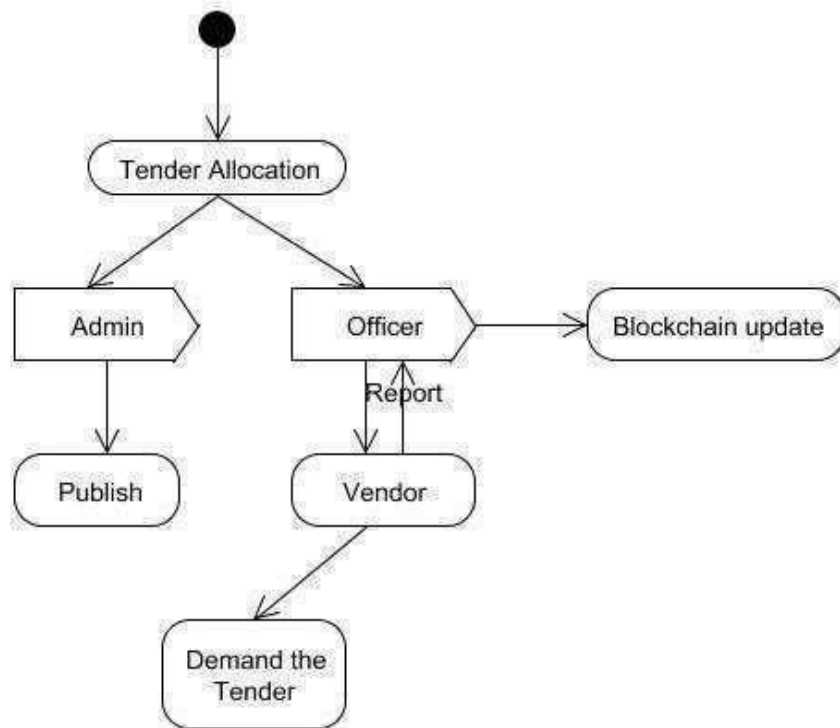


Fig.5.1.Activity Diagram

5.5 COLLABORATION DIAGRAM:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

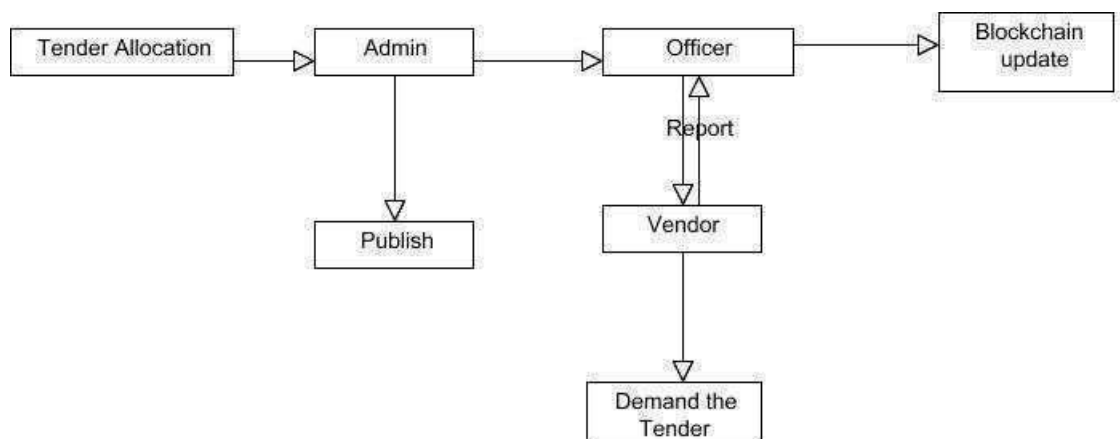


Fig.5.1.Collaboration Diagram

5.6 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

Level 0:

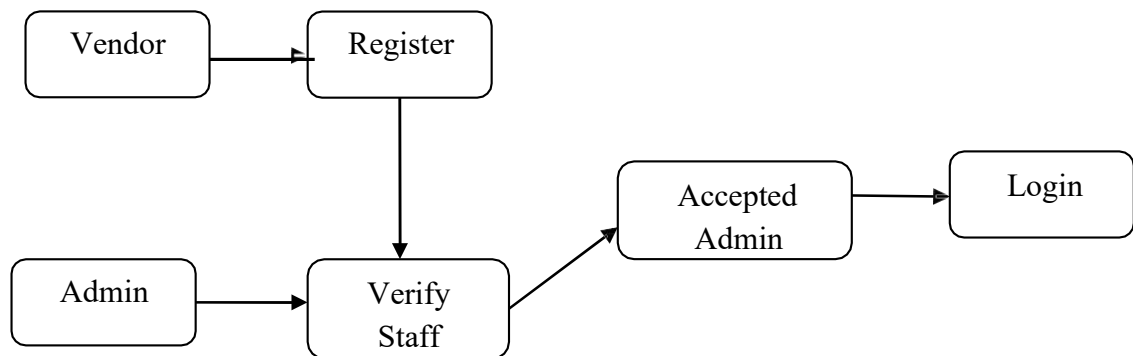


Fig.5.6.1.DATA FLOW DIAGRAM LEVEL 0

Level 1:

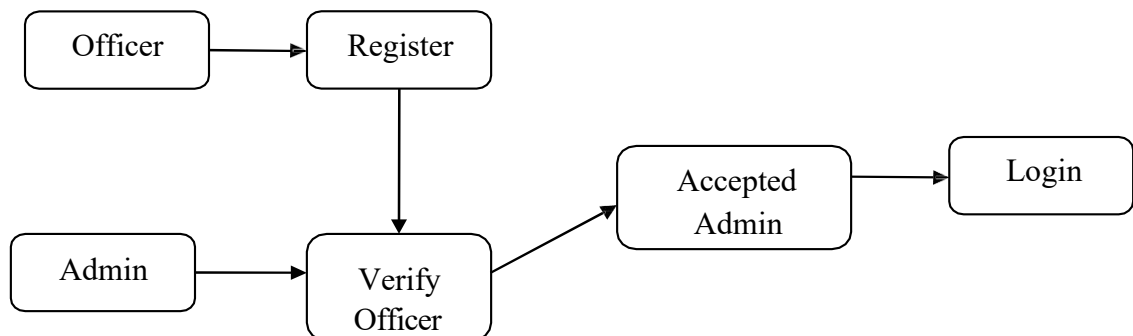


Fig.5.6.1.DATA FLOW DIAGRAM LEVEL 1

Level 2:

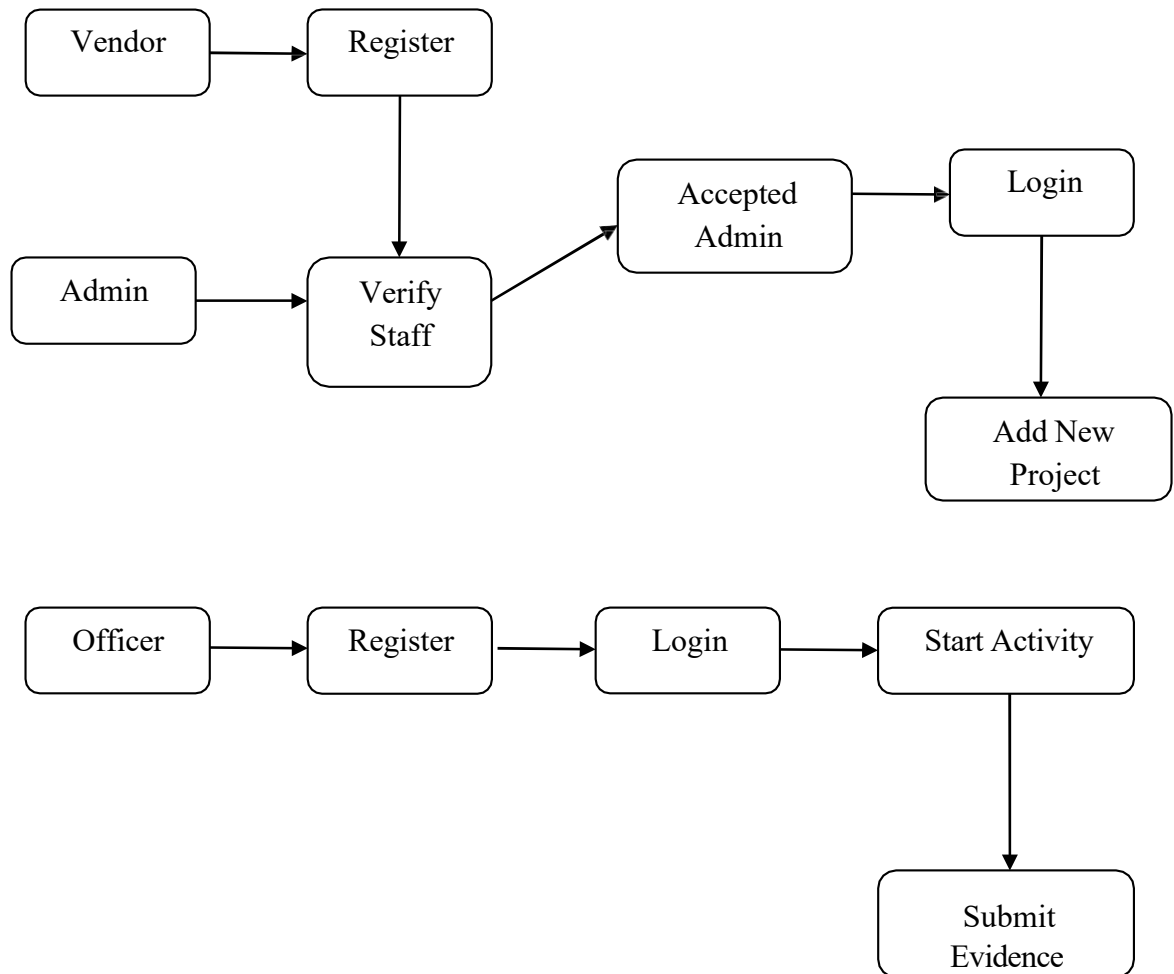


Fig.5.6.1.DATA FLOW DIAGRAM LEVEL 2

Level 3:

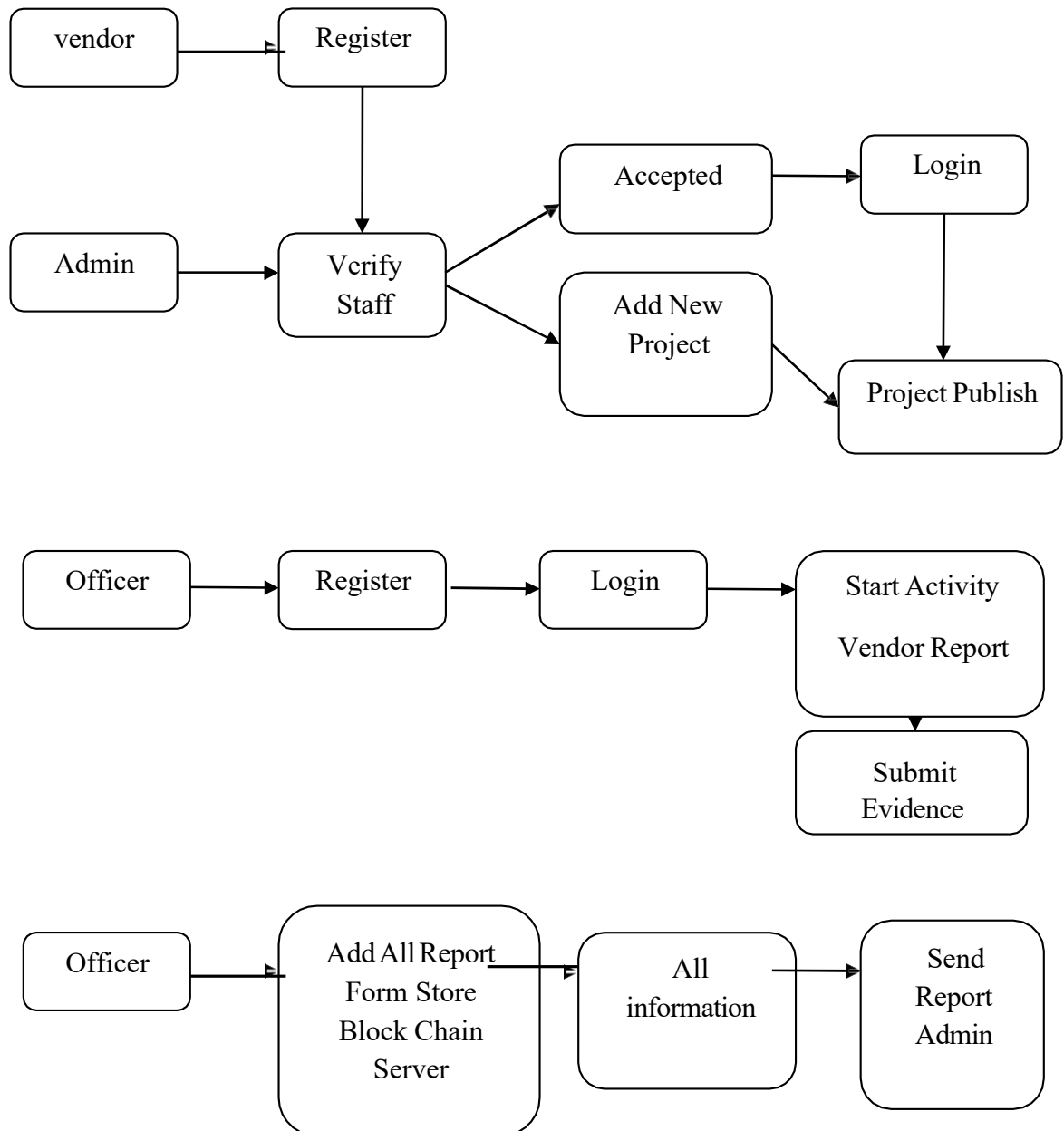


Fig.5.6.1.Data Flow Diagram

CHAPTER 6

RESULT AND DISCUSSIONS

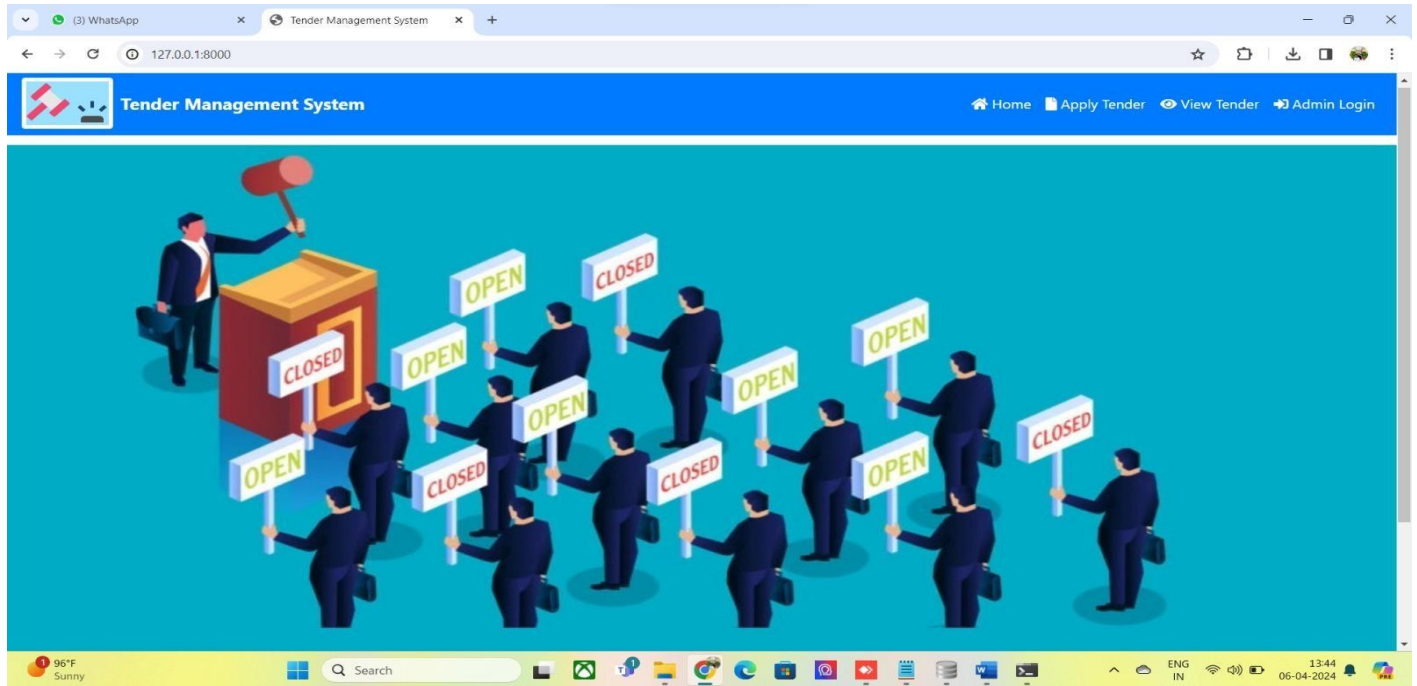


Fig.6.1.HOME PAGE

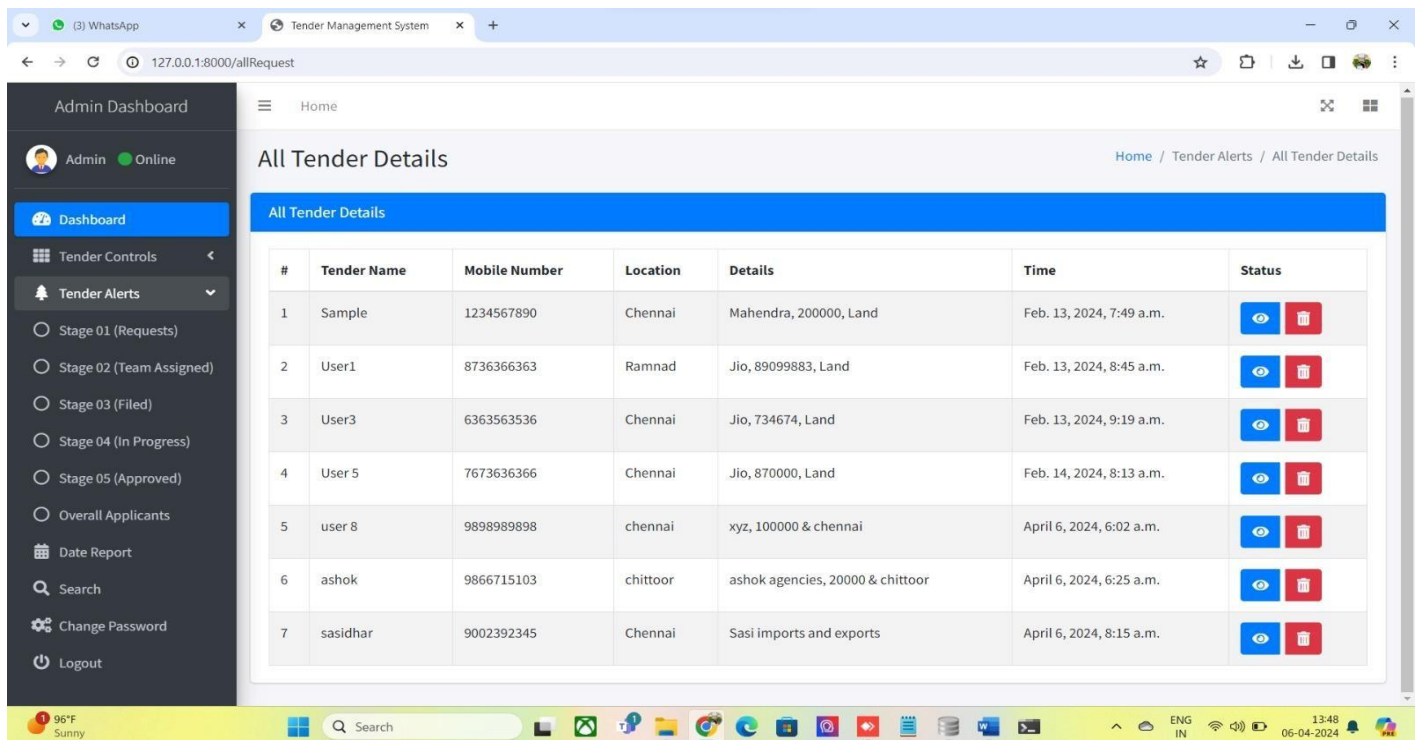


Fig.6.1.ADMIN DASHBOARD

CHAPTER 7

CONCLUSION AND REFERENCES

CONCLUSION:

In this paper, we have elaborated on the need and benefits of using blockchain technology in the government tender assignment process. We have used Ethereum to implement the end-to-end edge computing framework for government tender work-flow. Iterative auction algorithm is proposed to associate the best- suited constructors to the tender projects, thereby enhancing the profit of both the government lenders and the construction companies. We have also studied the performance evaluation of the proposed model.

The proposed model proves to give better results in terms of different tender parameters as compared to its counterparts. Implementing blockchain technology in smart tender contract management holds significant promise for enhancing transparency, security, and efficiency throughout the entire process. The decentralized and immutable nature of blockchain ensures that all parties involved have access to a tamper-proof and real-time record of contract-related information. This not only mitigates the risk of fraud but also streamlines the auditing process, reducing the potential for errors and disputes.

The use of smart contracts within the blockchain framework automates and enforces the execution of contract terms, minimizing the need for intermediaries and reducing administrative costs. This automation can lead to faster and more accurate processing of tender contracts, ultimately benefiting both the smart organization and its stakeholders.

Moreover, the transparency provided by blockchain technology fosters trust among the parties involved in the tender process. Stakeholders can access a shared, decentralized ledger that provides a comprehensive view of the contract's lifecycle, from creation to execution. This transparency helps build confidence in the integrity of the tender process and can contribute to stronger relationships with suppliers, contractors.

REFERENCES

- [1] C. Cachin, “Architecture of the hyper ledger blockchain fabric,” in Workshop on distributed cryptocurrencies and consensus ledgers, vol. 310, 2016.
- [2] C. D. Clack, V. A. Bakshi, and L. Braine, “Smart contract templates: essential requirements and design options,” arXiv preprint arXiv:1612.04496, 2016.
- [3] H. Hou, “The application of blockchain technology in e-government in china,” in 2017 26th International Conference on Computer Communication and Networks (ICCCN). IEEE, 2017, pp. 1–4.
- [4] S. Ølnes and A. Jansen, “Blockchain technology as s support infrastructure in e-government,” in International Conference on Electronic Government. Springer, 2017, pp.215–227.
- [5] A. Alketbi, Q. Nasir, and M. A. Talib, “Blockchain for government services—use cases, security benefits and challenges,” in 2018 15th Learning and Technology Conference (L&T). IEEE, 2018, pp. 112–119.
- [6] H. Cho, “Asic-resistance of multi-hash proof-of-work mechanisms for blockchain consensus protocols,” IEEE Access, vol. 6, pp. 66 210– 66 222, 2018.
- [7] J. D. Groot, “The History of Data Breaches.” <https://digitalguardian.com/blog/history-data-breaches>, online; Oct. 24, 2019.
- [8] coindesk, “The Indian government is preparing a national framework to support the wider deployment of blockchain use cases.” <https://www.coindesk.com/india-plans-to-issue-a-national-blockchain-framework>, online; Nov. 27, 2019.

- [9] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, July 2019.
- [10] J. A. Jaoude and R. G. Saade, "Blockchain applications—usage in different domains," *IEEE Access*, vol. 7, pp. 45 360–45 381, 2019.
- [11] H. Cho, "Correction to asic-resistance of multi-hash proof-of-work mechanisms for blockchain consensus protocols," *IEEE Access*, vol. 7, pp. 25 086–25 086, 2019.
- [12] V. Hassija, V. Chamola, S. Garg, N. G. K. Dara, G. Kaddoum, and D. N. K. Jayakody, "A blockchain-based framework for lightweight data sharing and energy trading in v2g network," *IEEE Transactions on Vehicular Technology*, 2020.
- [13] V. Hassija, V. Chamola, D. N. G. Krishna, and M. Guizani, "A distributed framework for energy trading between uav's and charging stations for critical applications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5391– 5402, 2020.
- [14] V. Hassija, V. Chamola, G. Han, J. J. Rodrigues, and M. Guizani, "Dagiov: A framework for vehicle to vehicle communication using directed acyclic graph and game theory," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4182– 4191, 2020.
- [15] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*. ACM, 2020, p. 30.

APPENDIX:

```
import datetime
```

```
from django.db.models import Q
```

```
from django.shortcuts import render, redirect
```

```
from .models import *
```

```
from datetime import date
```

```
from datetime import datetime
```

```
from django.contrib.auth.models import User
```

```
from django.contrib.auth import login, logout, authenticate
```

```
# Create your views here.
```

```
def index(request):
```

```
    return render(request, 'index.html')
```

```
def reporting(request):
```

```
    error = ""
```

```
    if request.method == "POST":
```

```
        FullName = request.POST['FullName']
```

```
        MobileNumber = request.POST['MobileNumber']
```

```
        Location = request.POST['Location']
```

```
        Message = request.POST['Message']
```

```
        try:
```

```
            Firereport.objects.create(FullName=FullName,
```

```
            MobileNumber=MobileNumber,
```

```
            Location=Location, Message=Message)
```

```
            error = "no"
```

```
        except:
```

```
            error = "yes"
```

```
    return render(request, 'reporting.html', locals())
```

```
#Admin New Add
```

```

def adminaddnewtender(request):
    error = ""
    if request.method == "POST":
        FullName = request.POST['FullName']
        MobileNumber = request.POST['MobileNumber']
        Location = request.POST['Location']
        Message = request.POST['Message']
        try:
            Firereport.objects.create(FullName=FullName, MobileNumber=MobileNumber,
Location=Location, Message=Message)
            error = "no"
        except:
            error = "yes"
    return render(request, 'reporting.html', locals())

def viewStatus(request):
    sd = None
    if request.method == 'POST':
        sd = request.POST['searchdata']
        try:
            firereport = Firereport.objects.filter(Q(FullName_icontains=sd) | Q(MobileNumber=sd) |
Q(Location_icontains=sd))
        except:
            firereport = ""
    else:
        try:
            firereport = Firereport.objects.filter()
        except:
            firereport = ""
    return render(request, 'viewStatus.html', locals())

```

```

def viewStatusDetails(request,pid):
    firereport = Firereport.objects.get(id=pid)
    report1 = Firetequesthistory.objects.filter(firereport=firereport)
    reportcount = Firetequesthistory.objects.filter(firereport=firereport).count()
    return render(request, 'viewStatusDetails.html', locals())

def admin_login(request):
    error = ""
    if request.method == 'POST':
        u = request.POST['uname']
        p = request.POST['password']
        user = authenticate(username=u, password=p)
        try:
            if user.is_staff:
                login(request, user)
                error = "no"
            else:
                error = "yes"
        except:
            error = "yes"
    return render(request, 'admin_login.html', locals())

def dashboard(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    totalnewequest = Firereport.objects.filter(Status_isnull=True).count()
    totalAssign = Firereport.objects.filter(Status='Assigned').count()
    totalontheway = Firereport.objects.filter(Status='Team On the Way').count()
    totalworkprocess = Firereport.objects.filter(Status='Fire Relief Work in Progress').count()
    totalreqcomplete = Firereport.objects.filter(Status='Request Completed').count()
    totalfire = Firereport.objects.all().count()

```

```

return render(request, 'admin/dashboard.html', locals())

def addTeam(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    error = ""
    if request.method == "POST":
        teamName = request.POST['teamName']
        teamLeaderName = request.POST['teamLeaderName']
        teamLeadMobno = request.POST['teamLeadMobno']
        teamMembers = request.POST['teamMembers']
        try:
            Teams.objects.create(teamName=teamName, teamLeaderName=teamLeaderName,
teamLeadMobno=teamLeadMobno, teamMembers=teamMembers)
            error = "no"
        except:
            error = "yes"
    return render(request, 'admin/addTeam.html', locals())

#Add Tender by Admin
# def adminaddnewtender(request):
#     if not request.user.is_authenticated:
#         return redirect('admin_login')
#     error = ""
#     if request.method == "POST":
#         FullName = request.POST['FullName']
#         MobileNumber = request.POST['MobileNumber']
#         Location = request.POST['Location']
#         Message = request.POST['Message']

#     try:
#         Teams.objects.create(FullName=FullName, MobileNumber=MobileNumber,

```

```

Location=Location, Message=Message)
#         error = "no"
#     except:
#         error = "yes"
#     return render(request, 'reporting.html', locals())


def manageTeam(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    teams = Teams.objects.all()
    return render(request, 'admin/manageTeam.html', locals())


def editTeam(request,pid):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    teams = Teams.objects.get(id=pid)
    error = ""
    if request.method == "POST":
        teamName = request.POST['teamName']
        teamLeaderName = request.POST['teamLeaderName']
        teamLeadMobno = request.POST['teamLeadMobno']
        teamMembers = request.POST['teamMembers']

        teams.teamName = teamName
        teams.teamLeaderName = teamLeaderName
        teams.teamLeadMobno = teamLeadMobno
        teams.teamMembers = teamMembers

    try:
        teams.save()

```

```

        error = "no"
    except:
        error = "yes"
    return render(request, 'admin/editTeam.html', locals())

def deleteTeam(request,pid):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    teams = Teams.objects.get(id=pid)
    teams.delete()
    return redirect('manageTeam')

def newRequest(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.filter(Status_isnull=True)
    return render(request, 'admin/newRequest.html', locals())

def assignRequest(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.filter(Status='Assigned')
    return render(request, 'admin/assignRequest.html', locals())

def teamontheway(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.filter(Status='Tender Assigned')
    return render(request, 'admin/teamontheway.html', locals())

def workinprogress(request):

```

```

if not request.user.is_authenticated:
    return redirect('admin_login')
firereport = Firereport.objects.filter(Status='In Progress')
return render(request, 'admin/workinprogress.html', locals())

def completeRequest(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.filter(Status='Approved')
    return render(request, 'admin/completeRequest.html', locals())

def allRequest(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.all()
    return render(request, 'admin/allRequest.html', locals())

def new_Request(request):
    firereports = Firereport.objects.all()
    return render(request, 'newRequest.html', {'firereports': firereports})

def deleteRequest(request,pid):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    firereport = Firereport.objects.get(id=pid)
    firereport.delete()
    return redirect('allRequest')

def viewRequestDetails(request,pid):
    if not request.user.is_authenticated:

```

```

    return redirect('admin_login')
firereport = Firereport.objects.get(id=pid)
report1 = Firetequesthistory.objects.filter(firereport=firereport)
firereportid = firereport.id
team = Teams.objects.all()
reportcount = Firetequesthistory.objects.filter(firereport=firereport).count()
try:
    if request.method == "POST":
        teamid = request.POST['AssignTo']
        Status="Assigned"
        team1 = Teams.objects.get(id=teamid)
        try:
            firereport.AssignTo = team1
            firereport.Status = Status
            now = datetime.now()
            firereport.AssignedTime = now.strftime("%m/%d/%Y %H:%M:%S")
            firereport.save()
            error = "no"
        except:
            error = "yes"
except:
    if request.method == "POST":
        status = request.POST['status']
        remark = request.POST['remark']

        try:
            requesttracking = Firetequesthistory.objects.create(firereport=firereport,status=status,
remark=remark)
            firereport.Status = status
            firereport.save()
            firereport.UpdationDate = date.today()

```

```

        error1 = "no"
    except:
        error1 = "yes"
    return render(request, 'admin/viewRequestDetails.html', locals())

def dateReport(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    error = ""
    if request.method == 'POST':
        fd = request.POST['fromDate']
        td = request.POST['toDate']
        firereport = Firereport.objects.filter(Q(Postingdate_gte=fd) & Q(Postingdate_lte=td))
        return render(request, 'admin/betweendateReportDtls.html', locals())
    return render(request, 'admin/dateReport.html', locals())

def search(request):
    if not request.user.is_authenticated:
        return redirect('admin_login')
    sd = None
    if request.method == 'POST':
        sd = request.POST['searchdata']
        try:
            firereport = Firereport.objects.filter(Q(FullName_icontains=sd) | Q(MobileNumber=sd) |
            Q(Location_icontains=sd))
        except:
            firereport = ""
    return render(request, 'admin/search.html', locals())

def changePassword(request):
    if not request.user.is_authenticated:

```

```
    return redirect('admin_login')
error = ""
user = request.user
if request.method == "POST":
    o = request.POST['oldpassword']
    n = request.POST['newpassword']
    try:
        u = User.objects.get(id=request.user.id)
        if user.check_password(o):
            u.set_password(n)
            u.save()
            error = "no"
        else:
            error = 'not'
    except:
        error = "yes"
return render(request, 'admin/changePassword.html', locals())
```

RESEARCH PAPER:



Smart Tender Contract Management System Using Block-Chain

Sasidhar Kuntamukala

Computer Science and Engineering
Sathyabama Institute of Science and Technology Chennai, India
sasidhark6626@gmail.com

Anegondi Ashok Raju
Computer Science and Engineering
Sathyabama Institute of Science and Technology Chennai, India
ashokraju1425@gmail.com

Dr. E. MURALI, MTECH., Ph.D.
Computer Science and Engineering Sathyabama Institute of Science and Technology
Chennai, India
murali.cse@sathyabama.ac.in

ABSTRACT- Supply chain communication is a key hobby that facilitates corporations find the right human beings to hire. In structure and development corporations, customers lease individuals who calculate and provide estimates of the expenses related to constructing of entirety. Thus, the purchaser could make his choice primarily based on the end result of tenders to pick out the most suitable challenge contractor. Donors need to acquire the e- book manager, compare it and produce it to this office before being notified. In this paper, we gift the layout and implementation of an electronic procurement system the use of internet-primarily based control as a mechanism for such furnish activities. In this paper, we've got used the innovation

of a complicated block chain structure to overcome security demanding situations and meet infrastructure necessities. A comfy structure of privilege in any union or U.S.A. Permits new freedoms to be expressed. This notion is higher than other schemes, due to the fact it is a decentralized structure that operates a spot with exclusive currencies at stake. We additionally need to interrupt down the security of our proposed grant shape, which indicates that our meeting is secure inside the event of conflicts with numerous agreements.

INTRODUCTION

Existing c-trade structures are not "truthful and open", that means that statistics isn't shared with all people. Stakeholders (right to Information). Information

For example, it's far published "at its discretion" - whilst a firm is selected because the winner of the agreement, other companies participating within the gentle are not informed why their bids have been rejected.?, a success corporation can request this records, but obtaining this information is a complex technique. Although it's far viable to hear those files, it is still time to assess them. Without simplicity, safety is a enormous difficulty of these portals, which results in the distortion and control of the records accrued within the information. If a hacker gains manipulate of this centralized database, it can be leaked to competitors, ensuing in huge financial and strategic commercial enterprise losses. The innovation barrier to those security features may be met because its miles based on decentralization and statistics encryption with an impregnable quadratic boom of alternate directors. Thus, the block chain can be used as a transparent, decentralized and comfortable gadget of tenders, which allows to control the charges of its gates and display the operations of all of the sports finished via the gentle gates.

OBJECTIVES

We use modern-day generation to resolve soft strength troubles and make sure safety and stability.

LITERATURE REVIEW

Literature review is an important step in the software program improvement method. Before the tool is evolved, it's miles important to determine the time factors, budget and strengths of the agency. With all these conditions met, the following step is to determine which running system and language can be used to increase the tool. When programmers begin developing a tool, they may want extra help. This assist can be obtained from senior packages, books or websites. Before designing the machine, the above factors are saved in mind while designing the machine. The maximum important part of the paintings of the venture improvement department is the specified evaluation of all the necessities which can be important to broaden the mission. Literature assessment is the maximum essential area of application improvement for any venture. Before

developing the equipment and the related plan, it's far important to discover and analyze the factors of time, aid requirements, manpower, monetary and organizational strengths. After a lot of these questions were replied and the studies carried out, the next step is to determine the specification of the software in the pc mission, what form of operating system will be required for the reason, and what software might be required for the challenge. Work Making tracks like equipment and associated sports.

1. *Intelligent software tools the use of Ethereal.*

By growing a software for the Ethereal block chain that can be used to control software program. Smart agreement facilitates in constructing the software as per the guidelines and conditions of the corporation. In addition, this transaction, applied at the ethereal block chain, makes the utility reliable, getting rid of the involvement of 0.33 parties. The application can be accessed and downloaded by using exclusive users, and the data isn't always saved in one vicinity. This decentralization is finished via disbursed belongings at the ethereal block chain. Casper on ethereal has verified powerful for verifying all varieties of transactions inside the software. In ethereal, the various types of orders created for a particular reason of the software are effectively represented and managed. When comparing Hyper ledger and ethereal, we used ethereal as it is basically a consumer-facet application. Different development groups can use it to create programs and share rewards through collaboration.

2. *Decentralized Market Application on Block chain* Modern, centralized on-line markets like eBay provide consumers with an opportunity choice to without problems sell and buy merchandise. However, the risks of these markets consist of the capacity of the platform to block dealers at will, the charges that must be paid to the platform when listing and promoting the product, and the lack of person privacy statistics. In this article, we advocate an software that addresses all 3 of those shortcomings the use of the ethereal block chain platform.

The application become advanced the use of a TUBE development surroundings. The feature of the application is contained in an ethereal smart settlement, that's then broadcast on the ethereal network. User enter is read via the net interface and dispatched to the ethereal community via the Web3.Js API. Statistics had been accumulated using the Rinke by take a look at network. The common application transaction time changed into proven to be

3. Eight seconds, and the common fuel intake changed into 4.6 VE. The contract creation time for the application is proven in less than a 2nd. After this the cost evaluation turned into carried out. Gas intake for corporations to shop for and promote gadgets is transformed to US greenbacks, and the gas price for the app is compared to the cost of using an public sale website online along with eBay and in-character auctions. Like Sotheby's homes Results show that selling via an app is inexpensive than existing on-line options and present in-individual alternatives. These tests display that our software efficiently addresses the failure of our public sale market.

3. *Smart Contracts: Methods for Ecosystem Security and Sustainability.*

Smart agreement technology built at the block chain are attracting quite a few attention in new enterprise programs and the medical network, because they allow untrusted events to explicit the terms of the settlement in software program code, eliminating the want for a depended on third party. The method of writing properly-functioning and comfy contracts on ethereal, modern-day most famous smart contract platform, is a daunting mission. Research in this topic has lately started in enterprise and academia. Based on the analysis of the accumulated information the use of idea-based totally methods, we've developed numerous fashionable security standards primarily based on solidity, the principle ethereal programming language. Templates provide answers to explain common safety issues and can be utilized by developers to mitigate not unusual attack eventualities.

4. *Design Patterns for Smart Contracts*

The concept of digitally simplifying agreement law and enterprise practices the usage of computer software program has given upward push to the idea of clever contracts. Today, the most distinguished patron agreement surroundings is ethereal, a block chain-primarily based, distributed computing platform. Writing legitimate and at ease smart contracts for ethereal is tough, due to the inherent nature of block chain-based contract execution, low programming abstraction, and the regular evolution of systems and security issues. Based on interdisciplinary literature research and facts evaluation accumulated through qualitative research techniques, we recognized numerous layout patterns that provide design recommendations. We describe those patterns in detail and provide pattern code to higher illustrate them. Our studies suggests that the models are extensively used for software necessities and common problems. We wish to generalize some or all the smart settlement models to other ecosystems, however that is outdoor the scope of this look at, which simply examines clever settlement methods in ethereal.

EXISTING SYSTEM

Lockdown has come to be an inevitable destiny in corporate finance, especially in commercial enterprise operations and logistics. Introducing a similar clever settlement concept, the Trust Sub-Ledger Operation (TSLO), this paper proposes a holistic block chain- primarily based framework to offer traceability and trustworthiness of documents throughout a pool of property. TSLO is a very flexible and adaptable approach for coping with things in accounting and making plans useful resource systems.

This method is primarily based at the Decentralized Micro service Tree (DMST) and an extensible shape of digital auction TEA (Triple Accounting). Instead of the usage of a multi-ledger structure, the skeleton of Hyper ledger Fabric, most effective for collaborating channels within a corporation or enterprise, our method makes use of decentralized sub-ledgers with processing bushes (TMST) for asset-based totally transactions. In addition, the government's method for auditing and taxing financial businesses will offer extra authority and protect more events via a aggregate of proof of authority and proof of possession.

Dis-Advantages

- Smart financial procedures are almost not possible to exchange, and solving each malicious program in the application is time-ingesting and high priced.
- Although there are continually twists and turns in financial affairs, not usually smart creditors can address the twists and turns.

PROPOSED SYSTEM

- Currently, corporations consisting of agriculture, healthcare, economic offerings and the general public sector are beginning to make the leap. Although this gentle element remains hindered with the aid of this new era. If soft control is going wrong, abuses will cause massive losses. These include favoritism closer to contractors, terrible report retaining, loss of transparency, hacking, conversion of information and other comparable issues.
- We use technology to clear up the problems of gentle control and to ensure that it enables to maintain protection and performance. Thanks to encryption

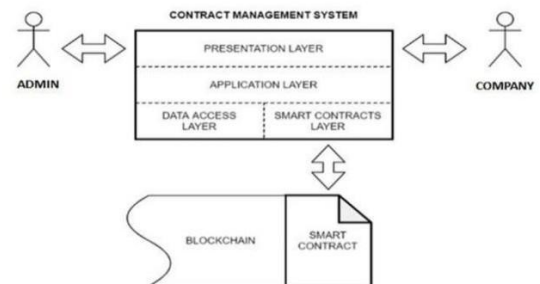
mixed with an impenetrable block architecture, block chain transaction management is absolutely comfy.

- Here we use block chain generation to attain project primarily based documents like tender documents, bids, bids, team details, beyond facts, authority approvals info and rejection information.

Advantages

- An smart gentle management machine by way of 0.33 parties reduces the possibility of manipulation. Pain tenders although the price is no intermediary.
- Encryption and cryptography protects all documents from unauthorized access.
- Automate laptop systems the usage of protocols, thereby saving time in performing many business strategies.

SYSTEM ARCHITECTURE



SYSTEM REQUIREMENTS

Hardware Requirements System: Pentium i3 processor. Hard Drive: 500 GB. Monitor: 15" LED

Input tool: keyboard, mouse RAM: 2GB

Software Requirements Operating device: Windows 10 Coding Language: Java

MODULES

- ❖ Admin Module
- ❖ Company Module

Admin:

- log in
- Manage tenders
- List of all tenders
- View / Update / Delete / Add
- add tender
- Update tenders
- Only certain essential and miscellaneous fields can be up to date.
- Tender present day
- Open a list of all tenders for packages/proposals.
- List of all programs obtained for tenders.
- Administrator replace by way of application fame (Approved / Rejected / Shortlisted)
- Previous tenders
- View packages as a list of all tenders for order status and current and closed renewal.
- View transactions
- List of tenders and programs for them with processing status
- It will display if a manipulation changed into performed without the consent or knowledge of the administrator.
- At the quit we are able to see how we are able to broaden the treatment.

Company;

- Registration
- log in
- Company is registered
- Tenders

- List of all tenders open to the government
- The enterprise can participate in its proposed undertaking.
- alternate the password
- Go to the admin panel, view the application and alternate the reput.
- Let's visit the "Companies" group.
- My utility
- List of packages and standing

