# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ACRONYMS/ABBREVIATIONS

| ACRONYM | Definition of Acronym |
|---|---|
| API | Application Programming Interface |
| CLI | Command-Line Interface |
| dApps | Decentralized Applications |
| DBMS | Database Management Systems |
| DOM | Document Object Model |
| EOAs | Externally Owned Accounts |
| ETH | Ether |
| EVM | Ethereum Virtual Machine |
| HTTP | Hypertext Transfer Protocol |
| IDE | Integrated Development Environment |
| IPC | Inter-Process Communication |
| IPFS | InterPlanetary File System |
| JSON | JavaScript Object Notation |
| JSON-RPC | Remote Procedure Call protocol encoded in JSON |
| JSX | JavaScript Syntax Extension |
| NPM | Node Package Manager |
| P2P | Peer to Peer |
| PLA | Palestinian Land Authority |

| | |
|---|---|
| PoS | Proof of Stack |
| PoW | Proof of Work |
| SOA | Service-Oriented Architecture |
| SQL | Structured Query Language |
| TOR | The Onion Routing |
| XML | Extensible Mark-up Language |

*Chapter One*

# 1 INTRODUCTION

With the rapid technological advancement and recent development, all organizations need to transform their systems to digitization and digital transformation. As there are many institutions that may be exposed to fraud in one of the most important services they provide, which leads to an imbalance in the security and reliability of the institution.

Among the institutions that this report will be devoted to mentioning is the Palestinian Land Authority, which is considered one of the most important institutions that preserve the rights of citizens and is frequented by citizens because of the daily services it provides. One of the most important services it provides is the direct buying and selling process.

It is one of the important operations in the institution, as noticed during our continuous interviews with the employees of the Palestinian Land Authority the possibility of fraud in one of the contract parts from internal or external parties, which affects the ownership in the long term, and this is what prompted us to find a proposed solution based on Blockchain technology to save contracts and maintain them in the long run .

"A Blockchain is a decentralized, distributed and public digital ledger that is used to record transactions across many computers so that any involved record cannot be altered re proactively, without the alteration of all subsequent blocks. This allows the participants to verify and audit transactions independently and relatively inexpensively "[1]

Therefore, this report will provide a clear explanation of the Blockchain technology and will focus on how to use it in the proposed solution and clarify the practical steps to implement this solution. As mentioned, previous research and studies followed the same technique in the solution, with differences in the mechanism of the solution and the nature of its implementation. It was talked about the practical results resulting from the practical steps followed and finally the future plan for developing the proposed system

## 1.1 PROBLEM STATEMENT

In the process of sales contracts in general, such as sale and purchase contracts, fraud is carried out either by one of the internal parties (employees) in the Land Authority or external parties (lawyer, citizen) this leads to long-term ownership problems.

## 1.2 OBJECTIVE OF THE PROJECT

### 1.2.1 Main Objective

Develop a model based on Blockchain technology to verify the validity of contracts (the study case is direct buying and selling of land).

### 1.2.2 Other Objective

- Examining the Land Authority system in depth and studying it well to achieve the greatest benefit from the project.
- Deep understanding of previous studies and works in order to master the merits and try to implement them as much as possible to improve performance and avoid the negatives found in previous work to produce satisfactory results.
- Model development according to the following steps:
    - Designing the system.
    - Implementation of the final system currently, implementation is limited to the Land Authority, but in the future, implementation will be on any contracts.
    - Testing the system after completing it well.
- Evaluate the Model after the design and testing phase.

## 1.3 SCOPE OF PROJECT

- The project targets the Palestinian Territory Authority in the Gaza Strip.
- Blockchain technology is applied to the direct buying and selling process that takes place between the two parties within the Palestinian Land Authority.

## 1.4 LIMITATION

- Lack of knowledge of Palestinian land employees Blockchain.
- Opposing the use of new technology and the method of decision-making.
- Cost of Blockchain need Money.
- When adding the attachments ( Legacy inventory , tax free certificate , ID photo

  Record extractor and agencies) related to the contract, we assume that they are correct So that we will take a hash of the attachments after scanning them.
- The proposed Solution hasn’t been deployed on a real Blockchain environment, because it needs a specific budget for write transactions

## 1.5 OVERVIEW OF THE DOCUMENT

The rest of the document is organized as follows:

• **Chapter 2 Theoretical Background:** Includes more detail about the Blockchain, types, platforms, block structure , Then we explained the nature of the work of the Palestinian Land Authority and the procedures followed in establishing a direct sale and purchase contract. Then we dedicated our work on the Ethereum platform and mentioned some of the concepts related to it and the nature of the tools needed to deal with it, then we mentioned the concepts related to the web that were used in the proposed solution.

• **Chapter 3 Literature Survey:** A survey of about previous studies and works that used Blockchain technology in their work, where three main sections were discussed, namely: Studies talking about smart contract, Studies talking about Blockchain land registration, Studies talking about Implementation Blockchain and we explained the differences between the previous works and the proposed solution

• **Chapter 4 Methodology:** Is where we describe the proposed solution was clarified in terms of design and implementation, where we explained the structure of the proposed system and the mechanism used in the project, where we used the Incremental prototype method, and then we talked about the operational steps through the presentation of the Usecase & UML and the testing processes that included the test API & smart contract

• **Chapter 5 Experiment and Results:** Testing the code ensuring it meets the

Required functionality and also measuring the performance of the system.

**Chapter 7 Conclusion and Future Work:** Conclusion for the report and the

System and asset its importance to the organization. It will also include Future

Work that can be done to enhance the system

*Chapter Two*

# 2 THEORETICAL BACKGROUND

In this chapter is explained in detail about the Blockchain, types, platforms, block structure, then we explained the nature of the work of the Palestinian Land Authority and the procedures followed in establishing a direct sale and purchase contract. Then we dedicated our work on the Ethereum platform and mentioned some of the concepts related to it and the nature of the tools needed to deal with it, then we mentioned the concepts related to the web that were used in the proposed solution.

This chapter divide this chapter in four important parts:

- Underline of Organization
- About Decentralized App
- About Web Application Technologies
- Summary

## 2.1 ABOUT BLOCKCHAIN

In this section we will explain the more detail about Blockchain. Include:

- Blockchain Definition
- Structure of Blockchain
- Reasons of use Blockchain
- Consensus Mechanism
- Types of Blockchain
- Blockchain Platforms
- Blockchain Vs Database

### 2.1.1 Blockchain

Blockchain technology which is "in the simplest of terms, a time-stamped series of immutable records of data that is managed by a cluster of computers not owned by any single entity. Each of these blocks of data (i.e. block) is secured and bound to each other using cryptographic principles (i.e. chain)." [2] Blockchain is a distributed database that is widely used for recording distinct transactions. Once a consensus is reached among

different nodes, the transaction is added to a block that already holds records of several transactions. Each block contains the hash value of its last counterpart for connection. All the blocks are connected and together they form a Blockchain. [3]

### 2.1.2 Structure of Blockchain

First, Blockchain involves three basic concepts: block, chain, transaction. The “block” refers to distributed data, the “chain” means the chronological string of blocks arranged by transaction ledger and “transaction” the read/write operations on the block that will store/ retrieve the data. Figure 2-1 shows that the blocks are linked into a chain in chronological order, so that each block retaining the hash value of the previous block, to give finally the criteria of de-trusted, decentralized, distributed data storage structure. Since this technology uses cryptographic hashes, it ensures the data of any transaction can’t be forged or tampered beside the ability of verification against integrity and security. While the distributed nature is served by the distributed data storage across network and peer to peer (P2P) communication [4].

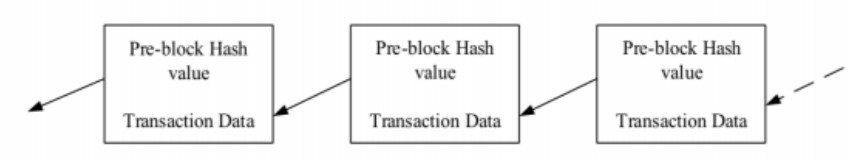



**Figure 2-1: Blockchain Technology Mechanism [5]**

### 2.1.3 Structure of Block

A block is consists of a header and a ledger of transactions and events. The structure of a block in a Blockchain is depicted in Figure 2-1 the following is an explanation:

#### a. Block Header

Three different types of info consists of the block header.

- **Version and Previous Block Hash**

The version (4-byte) field is used to keep track of software and protocol upgrades. The prior block hash field (32 bytes) refers to the preceding block's hash. By applying

SHA256 twice, the cryptographic hash algorithm is created. Every block in a Blockchain is linked to the previous block. The hash of the previous block is utilized to generate the hash of the current block. The genesis block is the first block on the Blockchain.

- **The Mining Competition Scheme**

The timestamp (4 bytes), nonce (4 bytes), and difficulty objective are all included in this collection of information (4 bytes). The timestamp indicates when the block was created. The nonce (one-time use number) is a number appended to a hashed block that fulfills the difficulty target level limitations when rehashed. Before attempting to solve for a block in the Blockchain, a Blockchain miner must first discover the nonce.

- **The Merkle Tree Root**

This is the hash of the Merkle tree's root, which holds the block's data structure. The Merkle tree root is formed by hashing pairs of transaction nodes repeatedly until only one hash remains. The procedure is carried out from the bottom up, starting with individual transaction hashes.

### b. Block Data

A Merkle tree is a digital fingerprint of a block's whole collection of transactions that allows a user to verify whether or not a transaction is part of the block. It aids in ensuring that subsequent versions of the event log contain all prior versions and that all data is captured and presented in chronological sequence. The block data includes a ledger of events, a transaction list, and any additional information for all legitimate and validated transactions that have been broadcast on the Blockchain network.

### c. Chaining of Blocks

The miners are in charge of creating new blocks. Any miner with the right to add a new block to the chain can add a block to the chain. The miner takes the hash of the previous block in the chain, adds its own set of messages, and generates a new hash for the freshly produced block. This freshly produced block now serves as the chain's new end. Transaction integrity and non-repudiation are ensured as a result of this, as any changed

blocks can reject and detect them. The generic chaining procedure in Blockchain is depicted in Figure 2-2

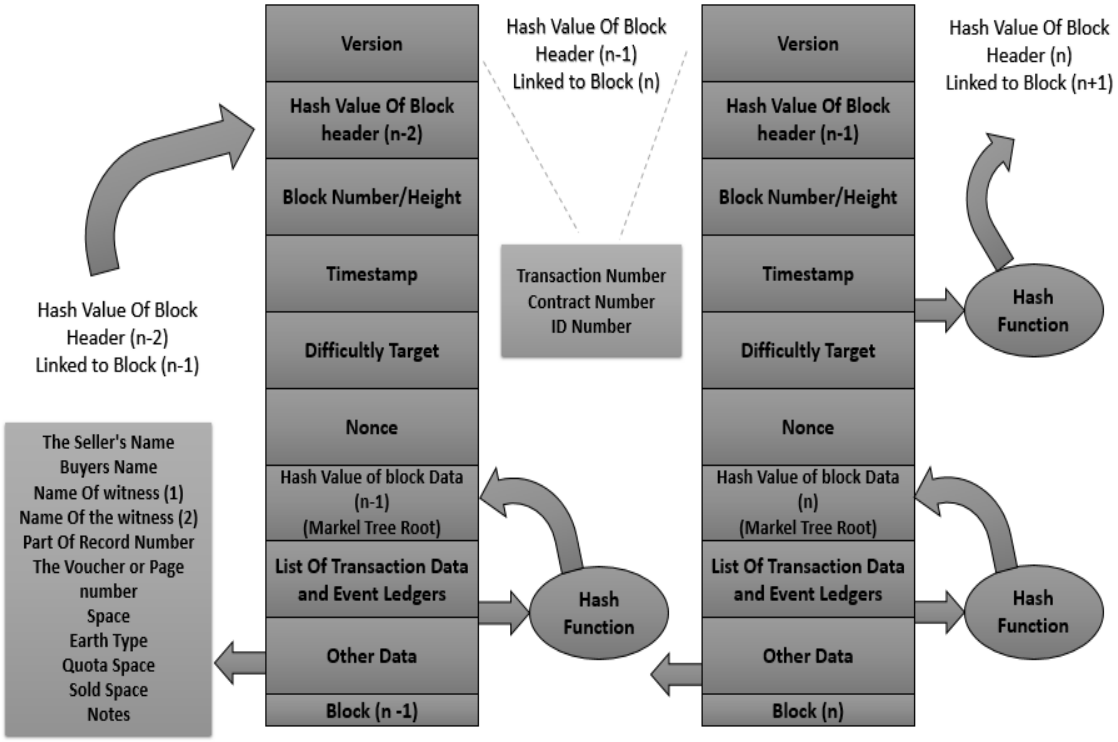



**Figure 2-2: Structure of Block**

## 2.1.4 Reasons of use Blockchain

Most of the appeal towards Blockchain technology revolves around five themes associated with its key features [6]

### a. Decentralization

Any node in the network owns the information and can access and copy the files stored in Blockchain so that, the network nodes can directly exchange data based on trusted system, which improve the efficiency of data exchange. Moreover, the damage of a single node will not affect the data of the whole network.

### b. Immutability

Means that once the data has been entered into the Blockchain, it cannot be tampered with .The reason why the Blockchain gets this property is that of the cryptographic hash function.

### c. Security

In Blockchain, data is verified and secured using cryptography. This will restrict all unauthorized changes and hacks in the system. It removes the middlemen from the system so no one can make any unauthorized changes

### d. Trust

Blockchain network makes the trust decentralized too “Unlike the centralized trust I take for granted, such as central governments issuing currencies and commercial banks, Blockchain network acts as new trust bearers with decentralized ledgers. These ledgers are shared among a network of tamper-proofed nodes.

### e. Privacy safe

The user is completely invisible in the transmission process, because the data are transmitted using public-private keys due to digital signature algorithm [7]

## 2.1.5 Consensus mechanism

A consensus method is a fault-tolerant technique used in computer and Blockchain systems to obtain the necessary agreement among distributed processes or multi-agent systems, such as cryptocurrencies, on a single data value or a single network state. It's handy for keeping track of items, among other things. [8]

### a. Proof of Work

The proof-of-work (PoW) algorithm specifies that a certain amount of work is completed for each block. Before it is proposed to the entire network, it must have a certain number of transactions. [8] If a node in a Blockchain network wants to publish a block of transactions, it must first prove that it is not likely to attack the network. In PoW, each

node calculates the hash value of the block header. A nonce is included in the block header, and miners change it frequently to obtain different hash values. [9]

The estimated value must be equal to or less than a specific value, according to the consensus. When a node reaches the desired value, it broadcasts the block to other nodes to verify that the hash value is correct. Other miners will attach this new block to their own Blockchains if the block is validated. Miners are nodes that calculate hash values, and mining refers to the PoW method. Bitcoin and other cryptocurrencies make use of it. The primary issue with this algorithm is that it consumes a lot of energy. [10]

### b. Proof of Stake

When nodes or users have enough stakes (economic stake) in the system, the proof-of-stake (PoS) method works. It's not about mining, but rather about validating transaction blocks. In PoS systems, validators must demonstrate ownership of a certain quantity of currency in order to participate in transaction validation, as it is assumed that persons with more currencies are less inclined to attack the network. A validator's possibility of producing a block is related to its stakes; the higher the stakes, the better the likelihood of validating a new block of transactions. [11]

PoS systems require less energy and work considerably faster than PoW systems since the originator of a block in a PoS system is deterministic based on the economic stake. It does, however, suffer from the problem of the rich getting wealthy. [12]

### c. Proof of Elapsed Time

The Proof of Elapsed Time method selects the producer of a new block at random and fairly depending on the amount of time they have waited. The method does this by assigning each user a random wait time, with the user whose wait time expires first producing a new block. This consensus process only works provided the system can verify that no single user may start several nodes at the same time and that the wait time is genuinely random. [13]

### d. Proof of Capacity

Solutions to complicated mathematical riddles are stored in digital storages such as hard drives using the Proof of Capacity approach. Plotting is the term for the entire procedure. Users can use a storage device to make blocks once it has been filled with solutions to mathematical puzzles. Users that find the solutions the fastest are given the opportunity to design a new block. As a result, those with the most storage space will have a better chance of creating a new block. [14]

### e. Proof of Identity

A user's private key is compared to an authorized identity in Proof of Identity. Proof of Identity is a cryptographic piece of proof for a user's private key that is cryptographically associated with a specific transaction. A Blockchain network allows any recognized user to generate a data block that may be displayed to anybody else on the network. Proof of Identity assures the data's integrity and validity. Smart cities can also leverage Blockchain consensus techniques such as Proof of Identity to validate its people identities. [15]

## 2.1.6 Types of Blockchain

Blockchains are classified in terms of permission into two major classes, namely public and private Blockchains.

### a. Public Blockchain

The public Blockchain, also known as permissionless Blockchain, is a decentralized and open system. Transactions can be read, sent, or received by any peer that joins the network. In addition, the peer can take part in the consensus process. There is no central authority in charge of membership management. A public Blockchain, such as Bitcoin, is an example. [16]

### b. Private Blockchain

Joining the network, reading or writing transactions, or participating in block creation are all restricted in a private Blockchain or permissioned Blockchain (mining). A central authority administers the membership and assigns rights to peers in this system. The

credibility of the participant peers is ensured by this management. The permissioned Blockchain Hyperledger is the most well-known. [17]

## Comparing the different Types of Blockchains

The main difference between a public and private Blockchain is the level of access granted to participants. Public Blockchains are completely open and allow anyone to participate by verifying or adding data to the Blockchain (a process called 'mining') such as Bitcoin and Ethereum. In contrary, the private Blockchain only allows certain authorized entities to participate in a closed network. Also referred to as 'permissioned Blockchain' [18]

**Table 2-1 *:* Comparing the different Types of Blockchains [19]**

| | Public (eg. Bitcoin) | Private | Consortium/ Permissioned (eg. EHRs) |
|---|---|---|---|
| **Network type** | Decentralized | Partially decentralized | Partially decentralized - hybrid between public and private Blockchains |
| **What is it?** | Anyone anywhere in the world can read and write on the network. Data is validated by every participant ("node") in the network, thus making it very secure. | Permissions to read and write data onto the Blockchain are controlled by a single "highly trusted" organization – the owner of the Blockchain. | Permissions to verify, read and write on the Blockchain controlled by a few predetermined nodes. The choice of predetermined nodes can be different for every entity on the Blockchain. |
| **Benefits** | -Secure as the entire network verifies transactions<br>-Transparent as all transactions are made public with individual anonymity | -Efficient as verification is done by just owner of the Blockchain.<br>-Private as the owner can control who has access to read or write on the Blockchain. | -Efficient as relatively lesser nodes verify transaction<br>-Private as read and write access can be controlled by the predetermined nodes<br>-No consolidation of controlling power. |
| **Challenges** | Inefficient as all nodes need to verify the transaction | -Controlling power is consolidated to a single organization.<br>- Difficult to align many organization to use the same Blockchain. | |

### 2.1.7 Blockchain platforms

The development of Blockchain-based apps is enabled by Blockchain platforms. They might be either permissioned or unpermitted. Ethereum, Hyperledger, R3, Ripple, and EOS are just a handful of the Blockchain frameworks that individuals can use to create and host applications on the Blockchain. In Table 2-1 show the Comparison of popular Blockchain platform [20]

**Table 2-2: Comparison of popular Blockchain platform [21]**

| | **Hyperledger Fabric** | **Ethereum** | **Ripple** |
|---|---|---|---|
| **Description of Platform** | General Purpose Blockchain | General Purpose Blockchain | Payments Blockchain |
| **Governance** | Linux Foundation | Ethereum Developers | Ripple Labs |
| **Currency** | None | Ether | XRP |
| **Mining Reward** | N/A | Yes | No |
| **State** | Key-value database | Account Data | None |
| **Consensus Network** | Pluggable: PBFT | Mining: Proof of Work | Ripple Protocol |
| **Network** | Private or Permissioned | Public or Private | Public |
| **Privacy** | Open to Private | Open | Open |

### 2.1.1 Database Vs Blockchain

#### Database

A database is a data structure made up of tables and schemas that stores user and system data. It has SQL for creating, reading, deleting, and updating records. A database is managed by a database management system, or DBMS. The sensitive data is usually modified by database administrators. The architecture of a database is based on the client-server concept [22] .

We can see the difference between Blockchain and database in the below Table 2-3

**Table 2-3: Comparison between Database and Blockchain [23]**

| | Databases | Blockchain |
|---|---|---|
| **Definition** | Databases are centralized ledger which stores data in a structured way and is managed by an administrator. | Blockchain is a peer-to-peer decentralized distributed ledger technology. It was first introduced in 2009. |
| **Authority** | Databases are controlled by the administrator and are centralized in nature. | Is decentralized and has no centralized approach. However, there are private Blockchains that may utilize some form of centralization. |
| **Architecture** | Database utilizes a client-server architecture. | Blockchain uses a distributed ledger network architecture |
| **Data Handling** | The database supports CRUD (Create, Read, Update and Delete). | Blockchain utilizes Read and Write operations. |

| | | |
|---|---|---|
| **Integrity** | Malicious actors can alter database data | Blockchain data supports integrity. |
| **Transparency** | Databases are not transparent. Only the administrator decides which the public can access data | Public Blockchain offers transparency. |
| **Cost** | The database being an old technology is easy to implement and maintain. | Blockchains are comparatively harder to implement and maintain. |

## 2.2 UNDERLINE OF ORGANIZATION

The PLA is a government authority with a ministerial rank, established in 2002 during the era of the Palestinian National Authority to manage the land sector. The PLA was established according to Presidential Decree No. 10 of 2002 on 5/9/2002 by President Yasser Arafat. To achieve its goals, as the PLA carries out broad tasks that fall within its general objectives, which include registering property rights, and conducting comprehensive surveys of Palestinian lands, including the process of surveying and registering property in the state of Palestine. Also, the vision of the Authority is represented by: For the Authority to seek to be the organizational and administrative reference for everything related to the Palestinian lands at the governmental, institutional and civil level, and to lay down the rules and foundations upon which it is based, and to form with the other concerned parties the basic pillar for establishing and setting up land laws and the mechanism required to implement them [24]. On the date of 11/22/2020, an engineer was interviewed in the computer department, where she explained the procedures that take place during the direct buying and selling process within the authority

Figure 2-3 illustrate their procedure which consists of the following steps:

1. The parties to the contract go to complete the registration application

2. Collecting the papers needed to open the transaction

3. The citizen submits an application to open the transaction to the Receptionist

4. The submitted papers are checked and verified Initial signature on the Registration application for all parties to the transaction

5. Transforming the transactions into a preliminary audit and they are Audited

6. The citizen turns to the receptionist and makes the payment associated with the transaction

7. The transaction is checked again for the conversion

8. Giving the transaction serial contract numbers and canceling the Properties electronically after the approval of the registration officer

9. Final approval of the transaction

10. Registration is on paper records

11. Send to the archive

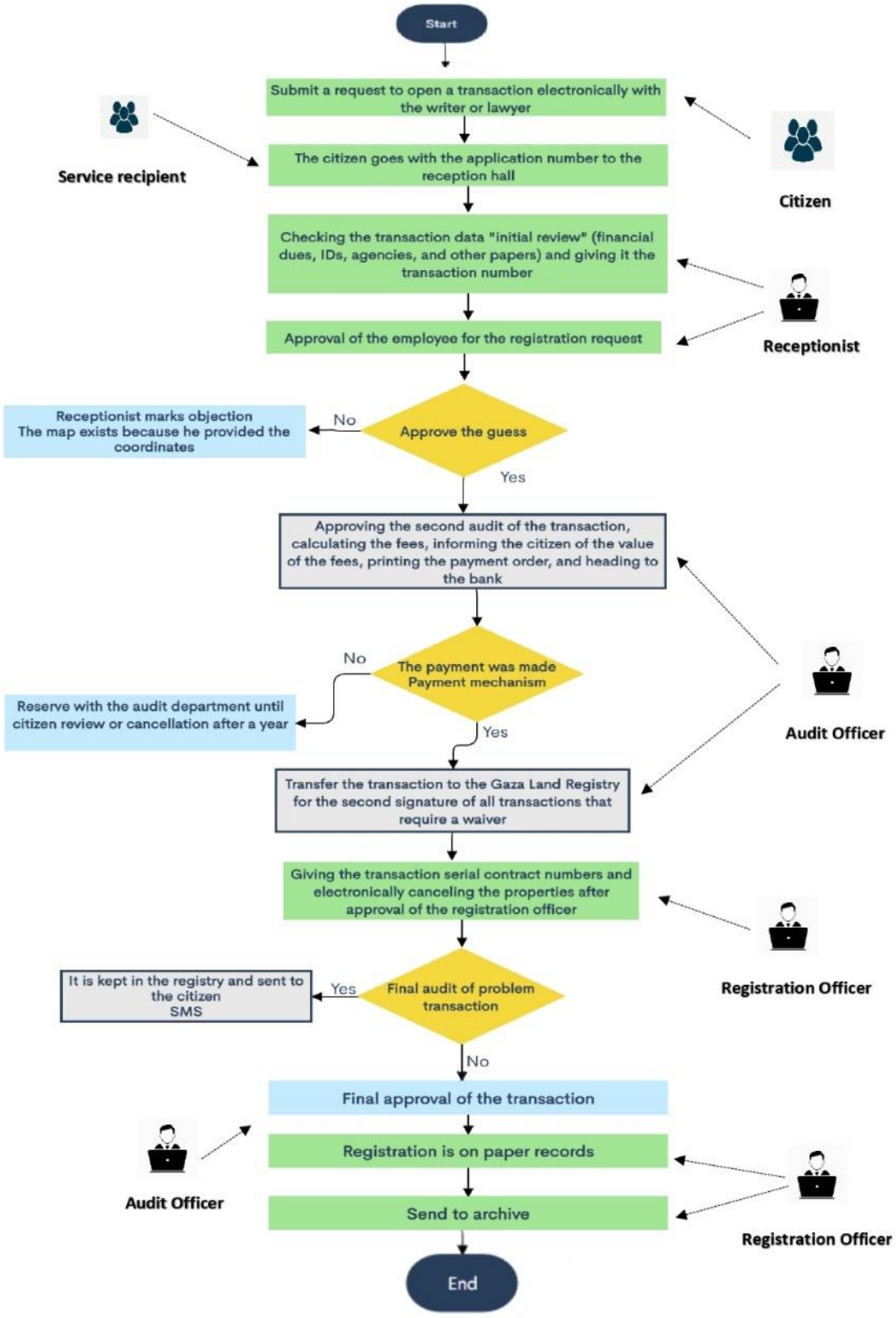



**Figure 2-3: PLA Process flow Mechanism**

## 2.3 ABOUT DECENTERALIZED APP

To build a decentralized web application, we will use Solidity to write the smart contract, Ethereum as a Blockchain network and the App will interact with Ethereum using Web3.

Above two lines lead us to little big questions:

- Decentralized applications
- Smart Contract
- Difference between dApps & smart contracts
- Ethereum Platform
- Reasons of use Ethereum
- Ether
- Gas
- Solidity
- Remix
- MetaMask
- Ganache
- Truffle suite
- IPFS

### 2.3.1 Denaturalized Applications - dApps

Decentralized apps (dApps) are digital applications or programs that reside and operate on a Blockchain or peer-to-peer (P2P) network of computers rather than a single computer, and are independent of a single authority's control. [25]

DApps can run on either a P2P or a Blockchain network. BitTorrent, Tor, and Popcorn Time, for example, are software that operate on computers that are part of a P2P network, where numerous users are consuming, feeding, or seeding material, or doing both roles at the same time. [26]

A dApps, like an app, can have frontend code and user interfaces written in any language that can communicate with its backend, the backend uses Blockchain infrastructure and returns any response to the web or mobile frontend. Frontend may also be stored on decentralized storage like IPFS. [27]

### 2.3.2 Smart Contract

Smart contracts are essentially programs that execute when certain criteria are satisfied and are maintained on a Blockchain. [28] They're often used to automate the execution of an agreement so that all parties may be confident of the conclusion right away, without the need for any intermediaries or time waste. They can also automate a workflow, starting the following step when certain circumstances are satisfied [29] Simple "if/when...then..." lines are inserted into code on a Blockchain to make smart contracts operate. When present circumstances are satisfied and validated, the activities are carried out by a network of computers. [30]

The goal of smart contracts is to provide security that is superior to traditional contract law while also lowering other contracting transaction expenses. Smart contracts have been applied in a number of cryptocurrencies. [31]

### 2.3.3 Difference between dApps & Smart Contracts

DApps are comparable to traditional web applications in appearance. The front end renders the page using the same technologies as the back end. The main distinction is that instead of an API connecting to a database, a Smart Contract connects to an Ethereum. Website with dApp support [32]:

Front End → Smart Contract → Ethereum

Decentralized apps include everything from the backend to the frontend. Only one portion of the dApp is a smart contract:

- The frontend (what you can see).

- The backend (the logic in the background).

A smart contract, on the other hand, is made up entirely of the backend and is usually only a minor component of the entire dApp.

### 2.3.4 Ethereum Platform 

Ethereum is a Blockchain platform that has its own cryptocurrency, Ether (ETH), as well as its own programming language, Solidity. [33]

Ethereum is a decentralized public ledger for validating and recording transactions as a Blockchain network. Users of the network may build, publish, monetize, and utilize apps on the platform, and they may pay using Ether, the network's cryptocurrency. Insiders refer to the network's dApps. As of May 2021, Ethereum is the second most valuable cryptocurrency behind Bitcoin. [34]

Developers can use Ethereum to raise funds for their own projects. They can create a contract and solicit pledges from the general public. It is referred to as ‘gas' in the case of Ethereum. Transaction costs are determined by bandwidth utilization, storage needs, and transaction complexity. Bitcoin transactions compete on an equal footing and are constrained by block size. [35]

### 2.3.5 Reasons of use Ethereum

- Decentralized Ethereum One of the most important features of Ethereum is that it is an open-source decentralized platform in addition to its high transparency. Things like this add value to developers who use the platform. [36]

- Ethereum is not just a cryptocurrency, it supports smart platforms, applications on the Blockchain, dApps [37]

- Multiple sources of learning and the presence of courses to explain the language of Solidity

### 2.3.6 Ether

Ether (ETH) is a cryptocurrency. It is the currency of Ethereum apps. It is a rare digital currency that can be used on the internet - comparable to Bitcoin. [38]

All of the Ethereum network's programs and services necessitate computational power (and that computing power is not free). Ether is a sort of payment used by network participants to carry out their specified network operations. [39] To understand the principle of Eth currency, it will be illustrated with an example:

If user ‘X’ wanted from user ‘Y’ to give him 100$, he would give him the amount in his hand. But in digital currencies, if the user ‘X’ asks user ‘Y’ to give him 100$, then user ‘Y’ will not send it as a monetary value, but an encrypted value called 64-bit hash will be

sent and coded at SHA 256 so that it is very difficult to guess. In the Blockchain comes a group of individuals for example who among them financial transactions recorded on the network and have an electronic ledger that records transactions of any kind and any amendment between the two on the network is the amendment to the ledgers for everyone and with the consent of all. Therefore, if user ‘X’ and user ‘Y’ are on the Blockchain and each of them has their own block, each block will have a ledger, and records contain what does the person takes from the other and what he gave. After the hash has been calculated, the process of transferring 100$ can be calculated, so that, there is 100 Eth, and the value of the digital currency going to user ‘X’. The topic needs very high encryption and security algorithms, which means they need massive computer resources. In fact, in a process called gold mining, workers took part in digging the ground and searching for gold. The same thing in digital currencies. Workers are digital computers, with participation in confirming and securing transactions. [40]

### 2.3.7 GAS

Gas is a unit of account within the EVM that is used to calculate a transaction fee, which is the amount of ETH that the sender of the transaction must pay to the miner who puts the transaction on the Blockchain. [41] The gas, which is priced in small fractions of the cryptocurrency ETH, is used to allocate resources of the EVM so that decentralized applications such as smart contracts can self-execute in a safe but decentralized form. The actual price of the gas is set by supply and demand among network miners, who can refuse to execute a transaction if the gas price does not match their threshold, and network users seeking processing capacity. [42]

### 2.3.8 Wallet

"Software applications for desktop, mobile or web-based systems (i.e., MetaMask for web, Jaxx for mobile and desktop)." [43]

Wallets maintain Externally Owned Accounts (EOAs) by storing private keys, accessing account balances, and creating, signing, and broadcasting transactions. contract call transactions to be transferred to other accounts value transactions contract creation transactions accept value exchanges from other EOAs [43]


**Figure 2-4: Metamask Wallet [94]**

### 2.3.9 Solidity

Solidity is high-level programming language, statically typed curly braces programming language designed for creating Ethereum-based smart contracts. [44] IT is highly influenced by C++, Python, and JavaScript and was created with the EVM. [45]

### 2.3.10 Remix

Remix IDE is a strong open source tool for writing Solidity contracts directly from the browser. It is written in JavaScript and can be used in the browser, in the browser but executed locally, or on the desktop. It includes modules for smart contract testing, debugging, and deployment, among other things. [46]

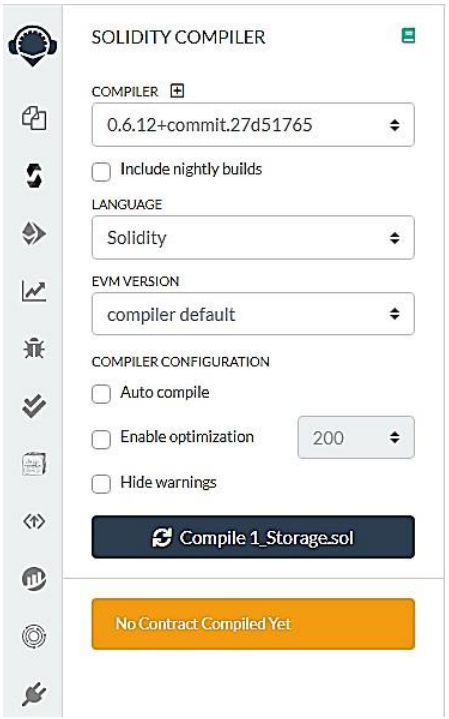



**Figure 2-5: Remix IDE Compile [90]**

### 2.3.11 MetaMask

A cryptocurrency wallet that also serves as a portal to Blockchain applications. Metamask, which is available as a browser extension and a mobile app, provides you with a key vault, secure login, token wallet, and token exchange, as well as everything you need to manage your digital assets. [47] The wallet works with the most popular browsers, including Chrome, Firefox, Brave, and Microsoft Edge. [48]

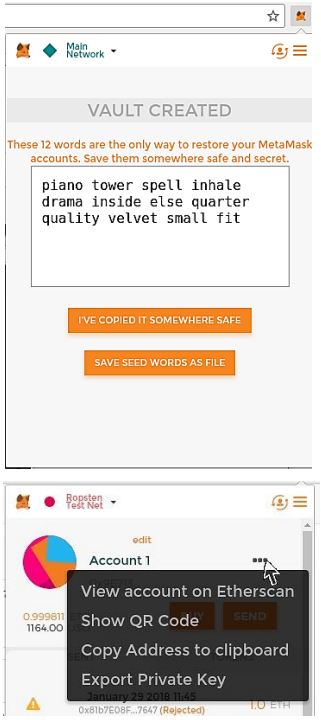



**Figure 2-6: Metamask Extension [91]**

### 2.3.12 Ganache

Is a personal Blockchain, which is a local development Blockchain that can be used to mimic the behaviour of a public Blockchain It enables the developer to deploy their smart contracts for testing purposes. It is available on Windows, Mac, and Linux as a desktop application and as a command-line tool called ganache-CLI or TestRPC previously. Using Ganache, Smart Contract can be deployed, Applications can be developed, tasks can be performed and tests without ant costs

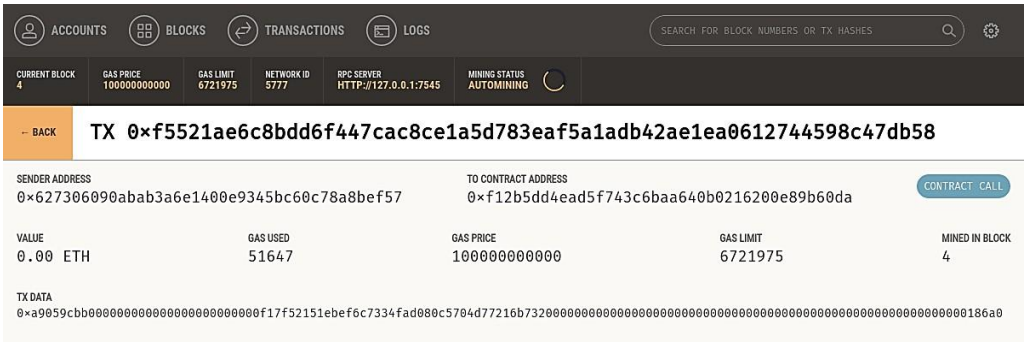



**Figure 2-7: Ganache Environment [92]**

### 2.3.13 Truffle suite 

A framework provides a suite of tools for compilation, linking, migrations and deploying an Ethereum smart contracts to any Blockchain networks [49]

It is an environment as platform for decentralized applications built on Blockchain and Ethereum technology. Truffle is a complete environment with all of the requirements to construct these applications, allowing you to collect contracts, publish them, and start them in the program, as well as work on interfaces and thorough testing. [50]

### 2.3.14 IPFS 


The InterPlanetary File System (IPFS) is a protocol and peer-to-peer network that allows data to be stored and shared in a distributed file system. A fundamental protocol for developing dApps. [51]

## 2.4 ABOUT WEB APPLICATION TECHNOLOGIES

To build a decentralized web application, we will use Solidity to write the smart contract, Ethereum as a Blockchain network and the App will interact with Ethereum using Web3.

To help you understand the coming practical work, we are going to give some technical background the following terms we will be covered in this section:

- Web Services
- API
- Web Services Vs API
- Web3
- Json
- Node JS
- Reasons of Node JS
- React
- Reasons of React
- Express
- NPM
- Postman

### 2.4.1 Web Services

A web service is a standardized method for communicating between client and server applications. A web service is a software module that is intended to carry out a certain set of duties. [52]

Web services, in other terms, are XML-based data exchange systems that leverage the internet for A2A (application-to-application) communication and interfacing. Programs, messages, documents, and/or objects are all involved in these processes. [53]

A significant aspect of web services is that applications developed in different languages may nevertheless connect with one another by transferring data over a web service between clients and servers. A client invokes a web service by submitting an XML

request, and the service responds with an XML response. SOA is frequently related with web services [53]

### 2.4.2 API

API stands for Application Programming Interface. APIs are a collection of methods and processes that enable the development of applications that access data and features of other programs, services, or operating systems. [54]

APIs provide a variety of functions. Some software products, for example, utilize APIs to communicate data to other software products, where the data is processed and/or shown to the user. In other circumstances, APIs are used by software applications to input data into databases. These are just a few instances of how an API may be put to use in practice. [55]

A developer can use API to transmit or receive data by making a specified "call" or "request." This communication is carried out with the help of a programming language known as JSON It can also be used to perform a predefined action, such as data update or deletion. [56]

There are four fundamental request methods available through API:

- GET: Gathers information
- PUT: Updates pieces of data
- POST: Creates
- DELETE

### 2.4.3 Web Services Vs API

Show the difference between API and Web Services [57]

**Table 2-4: Comparison between WEB SERVICE and API [58]**

| WEB SERVICE | API |
|---|---|
| All web services are APIs. | All APIs are not web services. |
| It can only be hosted on IIS. | It can be hosted within an application or IIS. |
| It is not open source but can be used by any client that understands XML. | It is open source and it can be used by any client that understands JSON or XML. |
| It requires a SOAP protocol to receive and send data over the network, so it is not a light-weight architecture. | It is light-weighted architecture and good for devices which have limited bandwidth, like mobile devices. |
| A Web service uses only three styles of use: SOAP, REST and XML-RPC for communication. | API may use any style of communication. |
| It only supports the HTTP protocol. | It supports the HTTP protocol: URL, Request/Response Headers, caching, versioning, content formats. |

### 2.4.4 Web3

Web3 is a convenience JavaScript library that allows you to communicate with a local or distant Ethereum node via HTTP, IPC, or WebSocket. [51]

Web3.js operates by exposing RPC-enabled functions and objects, allowing the building of user interfaces that disguise the JSONRPC API encoding. [52]

Web3.js is also used by development tools like Truffle and Remix, allowing contract interaction via their internal JavaScript Console. [52]

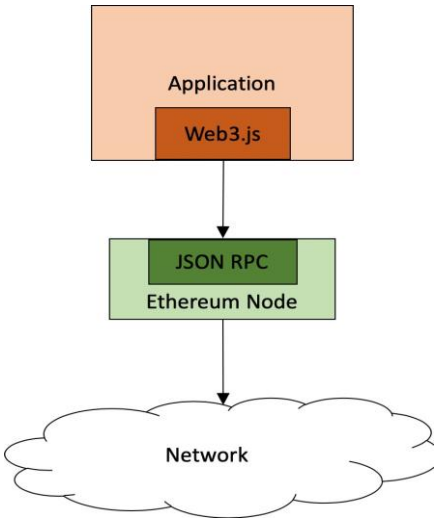



Figure 2-8: Web2 Component [93]

### 2.4.5 JSON

JSON (JavaScript Object Notation) is a simple data-transfer format. Humans find it simple to read and write. [59]

JSON is available as a string, which is important when transmitting data over a network. When you want to access the data, it must be transformed to a native JavaScript object. [60]

### 2.4.6 Node JS

Node.js is a cross-platform, open-source runtime environment for developing server-side and networking applications. Node.js apps are written in JavaScript and can run on OS X, Microsoft Windows, and Linux using the Node.js runtime. [61]

Node.js also has a large library of JavaScript modules, which greatly simplifies the creation of web applications using Node.js. [62]

**Reasons of use Node JS**

- **Easy learning curve**

  Knowing JavaScript will help a developer get started with Node.js. Knowledge of the programming language will make things much easier. [63]

- **Large community**

  As an open-source project, Node.js promotes support and contributions targeted at improving and expanding the platform's adoption. [64]

- **Scalability**

  Node.js works well with systems that use the microservices architecture or containerization to gain scalability and flexibility fast and efficiently. [65]

- **Fast**

  The advantage of Node.js is that it speeds up the execution of coding (written by developers when building the software). As a result, Node.js can accelerate any framework. [66]

### 2.4.7 React 

React is a JavaScript library for creating user interfaces that is declarative, efficient, and customizable. It allows you to build complicated user interfaces out of small, isolated sections of code known as "components." [67]

**Reasons of use React** [68]

- **Fast Learning Curve**

  React is very a simple and lightweight library that only deals with the view layer.

- **Testability**

  ReactJS apps are extremely simple to test. React views may be thought of as functions of the state, which means we can manipulate the state with the state we

supply to the ReactJS view, examine the output and triggered actions, events, functions, and so on.

- **Testability**

Simplicity React has a unique syntax known as JSX, which allows you to blend HTML and JavaScript. However, this is not mandatory, developers still can write in plain JavaScript, although JSX is far more user-friendly.

### 2.4.8 Express

Express is a lightweight and adaptable Node.js web application framework that offers a comprehensive range of functionality for web and mobile apps. [69]

**Features of Express.js** [70]

- **Faster Server-side development**

Many commonly used Node.js functionalities are provided by Express.js in the form of functions that can be used anywhere in the program. This eliminates the need to code for several hours, saving time.

- **Middleware**

  A middleware is a component of a program that has access to the database, client requests, and other middleware's.

- **Routing**

  ExpressJS includes a sophisticated routing mechanism that uses URLs to assist preserve the state of the webpage.

- **Templating**

  ExpressJS has Templating engines that enable developers to create dynamic content on web pages by creating HTML templates on the server.

- **Debugging**

  Debugging is critical to the success of web application development. ExpressJS simplifies debugging by offering a debugging technique that can pinpoint the particular area of the web application that has errors.

**Reasons of use Express.js** [71]

- Express.js supports JavaScript, which is a widely used language that is simple to learn and is extensively supported.
- As a result, if you already know JavaScript, programming with Express.js will be a breeze.
- With the help of Express.js, you can quickly create many types of web applications.
- Express.js supports simple client-side routing.
- Without Express.js, you must write your own code to create a routing component, which is a time-consuming and laborious operation.

### 2.4.9 NPM

Node Package Manager (NPM) The default package manager for JavaScript's runtime Node.js [72] . It facilitates the sharing and reusing of open-source code by allowing it to be installed as a module. Modules are JavaScript packages that may be installed on your system via NPM. NPM assists you in managing packages as dependencies in your projects. [73]

**NPM is divided into two parts:**

- Command-line interface (CLI) tool for publishing and downloading packages
- Online repository for JavaScript packages

### 2.4.10 Postman

Postman is a stand-alone software testing API (Application Programming Interface) platform for developing, testing, designing, modifying, and documenting APIs. It's a straight forward graphical user interface for sending and examining HTTP requests and answers. [74]

Almost all capability that a developer would want is incorporated in this program. This tool can perform HTTP queries such as GET, POST, PUT, and PATCH, as well as transform the API to code for languages such as JavaScript and Python. [75]

**Reasons of use Postman** [76]

- **Accessibility**

  After installing Postman on the device, users may access it from anywhere by checking in to their account.

- **Use of Collections**

  Postman users may create collections for their API calls. Every set has the ability to make numerous requests and subfolders. It will aid in the organization of the test suites.

- **Collaboration**

  To improve file sharing, you may import and export collections and environments. You may also share the collections via a direct link.

- **Creating Environments**

  The creation of several contexts leads in reduced test duplication because the same collection may be used in different settings.

- **Creation of Tests**

  Test checkpoints, such as confirming a valid HTTP response status, may be added to each Postman API call to assure test coverage.

- **Automation Testing**

  Using the Collection Runner or Newman, tests may be conducted in several repeats or iterations, saving time for repeated tests.

- **Debugging**

  The Postman console allows you to see what data was retrieved, making it easier to debug tests.

## 2.5 SUMMARY

In this chapter, we talked about the most important techniques used to build our project, through which we learned about the concept of Blockchain and its types and platforms.

Then we talked about an overview of the Palestinian Land Authority and the steps they take to create a direct purchase and sale contract

Then we moved on to talk about the Ethereum platform on which our project was built, and we talked about why we chose it

And while we were talking about it, we talked about the following concepts (Ether, Wallet, and Gas).

As we talked about the requirements needed to deal with Ethereum and what is required for it as a work environment, we talked about (Truffle suite, Ganache and Metamask)

And we talked about the language used to build smart contracts, which is “Solidity “

And we talked about the IPFS technology used to deal with images and documents of contract attachments

Then we moved on to talk about the web technologies used, where we talked about the API in addition to the languages used

React that was used to create simulated interfaces for the Palestinian Land Authority interfaces

In addition to Node, through which the API was created, and it was also talked about the postman through which the process of verifying the work of the API was carried out.

Where in the chapter is devoted to theoretical and introductory concepts for the upcoming chapters

*Chapter Three*

# 3 RELATED WORKS

At a unique moment in history where our society is undergoing a transition from an industrial economy to an economy defined by a new set of technologies, ranging from digital to ultrafine technology. Among the latest waves of digitization is Blockchain, a technology that many say promises to redefine trust, transparency and inclusion around the world.

After our extrapolation of some of the previous studies, it was found that Blockchain technology can be used in many areas, including in the sale and purchase contracts of the authority, as we noticed that there are previous indicators in our current era and this is what we have noticed in previous papers and studies, which indicates the Blockchain technology in that.

In our turn, in this report we will divide the scientific papers studies dealing with the Blockchain property into three important parts:

- The first section: Studies talking about smart contract.
- The second section: Studies talking about Blockchain land registration
- The Third section: Studies talking about Implementation Blockchain

## 3.1 FIRST SECTION: STUDIES TALKING ABOUT BLOCKCHAIN AND REGISTRATION

R.C. Suganthe aimed to develop a Blockchain Land Registry System with detailed and easy to use feature with high reliability and good interface.

It mainly focuses on covering the rules and procedures set by the Indian government in relation to land registration. Ensures enhanced security and accuracy of records.

But he did not mention the method of implementing the project and whether they used the public or the private, and when they mentioned the methods of implementing the

program in India and its benefits on the progress of the process of transferring land ownership [77].

Mohammed Shuaib presented in his paper a secure and reliable framework for a land registry system using Blockchain is proposed. The proposed framework uses the concept of a smart contract at different stages of the land registry and gives an algorithm for prior agreement. First, the traditional land registry system was codified and the issues in it reviewed. Next, he identified the potential benefits of using Blockchain technology in the land registry system and presented a framework or a number of case studies.

In our opinion, it would have been better to mention the impact of the application of algorithms on the transfer of ownership and how they helped provide security and reliability on the identification papers. [78]

Yogesh K.Dwivedic explores the use of Blockchain technology to manage land records in India.

It highlighted issues such as minimal transparency and accountability, inconsistent data sets with different government departments related to the same plot of land, delays in the existing land records management process, and how to overcome these problems using Blockchain technology. The paper described the current process of maintaining land records and land registry in the country, and discussed various challenges encountered during the implementation of Blockchain technology such as public key infrastructure, internet, privacy rules and security issues.

Finally, the paper explained the design of a system using Blockchain technology to implement the land titling system in the country, so that land titles are non-manipulable, and provide authentic and unequivocal rights over ownership. [79]

## 3.2 SECOND SECTION: STUDIES TALKING ABOUT SMART CONTRACT

Ilhaam A. Omar propose a general framework for decentralized auctions that leverages (1) Ethereum smart contracts to track and track bids, (2) decentralized storage systems for uploading documents related to bids and (3) trusted timers that act as a gateway

between smart feeds of contractual and external data. In the proposed solution, they developed detailed algorithms that define the working principles of smart contract for the auction process. They provided a detailed cost analysis of the solution to demonstrate its economic viability, providing a secure, transparent and reliable approach to online auctions. But the paper did not focus on the security aspect of using the Blockchain in the work of smart contracts. [80]

Merlinda Andoni focused on the issue of storage security, particularly the safety of customer data. Hence, this article has proposed a solution to monitor the integrity of files stored in the cloud, based on the use of smart contracts in Blockchain networks, symmetric encryption, and computational trust. The proposed solution consists of a protocol that provides confidentiality, decentralization, audit availability, and secure sharing of file integrity monitoring results, without overloading the respective services, as well as a non-abbreviated reference implementation that was used to validate the proposal. The results obtained during the validation tests showed that the solution is feasible and flawless in detecting corrupt files. These tests also confirmed that sharing integrity control results, along with applying computational confidence techniques, significantly increased the efficiency of the proposed solution. [81]

## 3.3 THE THIRD SECTION: STUDIES TALKING ABOUT IMPLEMENTATION BLOCKCHAIN.

Fahim Ullah provides a proof of concept for a use case that uses the Ethereum platform to build a Blockchain network to buy, sell, or rent property. This study concluded with several results that provided an opportunity to create new scalable decentralized solutions for the development of smart cities by enabling paperwork. There are huge opportunities in this distributed ledger technology that will bring about revolutionary change in the coming years. The concept of Blockchain can be used alongside smart contracts as a promising technology for sharing services and is a common requirement in smart cities. All Blockchain transactions are stored in a shared, decentralized database. The transaction registered in the decentralized system is not subject to change and cannot be changed. Therefore, the opportunity for fraud is negligible. [82]

## 3.4 COMPARISON BETWEEN SIMILAR PROJECTS

Table 3-1 compares our proposed solution with similar projects we mentioned above.

**Table 3-1: Comparison between Similar Projects**

| | **Merlinda Andoni** | **Ilhaam A.Omar** | **Yogesh K.Dwivedic** | **Mohammed Shuaib** | **R.C. Suganthe** | **Proposed Solution** |
|---|---|---|---|---|---|---|
| **Classification of Blockchain** | public | public | private | Private | Private | Public |
| **Scope** | storage security | providing a secure, transparent and reliable approach to online auctions | land records management | Protect archived records | Modernization of the land records management system. | Direct buying and selling |
| **Integeration with other system** | No | No | yes | No | yes | yes |
| **Platform** | - | Ethereum | - | - | - | Ethereum |

### 3.4.1 Summary

Through previous studies, we have attempted to demonstrate and illustrate the efforts of many researchers in reaching the concepts and features of Blockchain technology as it may be broadly scalable for a variety of situations. These features do not only apply to the immediate context of currency and payments, or to contracts, property and all financial market transactions, but outside them to sectors as diverse as government, health, science, literacy, publishing, economic development, art and culture, and perhaps on a larger scale to enable human progress on a larger scale, and we tried Focusing on the topic of smart contracts to fit our report on smart contracts at the Land Authority.

Blockchain technology could be fully complementary in a potential future world space that includes both centralized and decentralized models. Like any new technology, Blockchain is an idea that initially disrupted, and over time could foster the development of a larger ecosystem that incorporates both the old method and the new innovation. Some historical examples are that the advent of radio actually increased record sales, and e-readers such as the Kindle increased book sales. Now, we get news from The New York Times, blogs, Twitter, and personal drone feeds alike. We consume media from big entertainment companies and YouTube. Hence, over time, Blockchain technology could exist in a larger ecosystem with both centralized and decentralized models.

*Chapter Four*

# 4 METHODOLOGY

This chapter covers the details explanation of methodology that is being used to build this project complete and working well. This methodology is used to achieve the objective of the project.

So, this chapter divide this chapter in four important parts:

- Project Plan Management
- Project Design and System Modeling
- Summary

## 4.1 PROJECT PLAN MANAGEMENT

### 4.1.1 Methodology

We will choose the Incremental Prototype

**Reasons of use Incremental Prototype**

-Our understanding of the current system is not 100% in-depth

- Creating a Blockchain-based system that needs to be constantly modified due to the ever-evolving Blockchain technology

### 4.1.2 Software Mapping

The system consists of four independent components: Ethereum Smart Contract, API, and Web Services and React so for:

**Software requirements**

Main software applications were used to code the previous layers:

- Postman Tool to test the API

- Online editor remix for smart contract.
- Code editors (Visual Studio Code)
- Node JS
- React JS

**Ethereum dependencies:**

- NPM
- Truffle Framework
- Metamask Ethereum Wallet extension
- Ganache
- WEB3

### 4.1.3 Schedule Plan TimeLine

In this section, we explained the distribution of tasks to the team over the two semesters

And then clarified Gantt chart which reflects the timeline for the implementation of the project

**Table 4-1: Distribution Work to the team**

| Task | | Manar | Sahar | Saja | Rola |
|---|---|---|---|---|---|
| **Collect the information** | Information about Blockchain &Smart Contract | * | * | * | * |
| | Attend Online Blockchain Course (1) | * | * | * | * |
| | Attend Practical Online Blockchain Course (2) | * | * | * | * |
| | Analyzing Similar Projects (Related Works) | * | * | * | * |
| | • Learn Solidity<br>• Learn react<br>• Learn Node Js | * | * | | |
| | • Read this book (Blockchain A Practical Guide to Developing Business, Law, and Technology Solutions)<br>• Read this book (Ethereum for Web Developers)<br>• Read this book (Blockchain for Cybersecurity and Privacy) | * | | | |
| | Information about PLA System | * | * | * | * |

| | | | | | |
|---|---|---|---|---|---|
| **PLA Interviews & Analyze Current System** | Land Authority Interview(1) | * | * | * | |
| | Land Authority Interview(2) | * | | * | * |
| | Land Authority Interview(3) | * | | * | |
| | Land Authority Interview(4) | * | | | * |
| | Land Authority Interview(5) | * | * | | * |
| | Land Authority Interview(6) | * | * | | |
| | Land Authority Interview(7) | * | * | | |
| | Understanding the Current System | * | * | * | * |
| **Prepare Work Environment** | • Install Ethereum Dependencies<br>• Install React JS Rendering Library , Express and Node JS | * | * | | |
| **Analyze & Design** | Analyze the Current System | * | * | | |
| | Design figure & Flow chart | * | | * | |
| **Coding and programming** | • Write Smart Contract Code in Solidity<br>• Deploy Smart Contract into local Ganache network | * | * | | |

| | | | | | |
|---|---|---|---|---|---|
| | • API Back-end Server implementation to handle the requests<br>• Embedded Web3 JS to user interfaces for local testing | | | | |
| **Testing and evaluation** | • Test smart contract<br>• Test user interfaces<br>• Test functionality of each method<br>• Test API<br>• Test Integration | * | * | | |
| **Write Documentation** | Overview Of Blockchain & PLA | * | | | * |
| | Related Works | | | * | |
| | Ethereum and Technical Knowledge | * | | | |
| | Methodology | * | * | | |
| | Experiment and Results | * | * | | |
| | Conclusion and Future Work | | * | * | |

**Table 4-2: Gantt Chart of Distribution Work**

| Task | Sep | Oct | Nov | Dec | Jan | Feb | Mar | Apr | May | June | July |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Graduation Project Idea** | | | | | | | | | | | |
| Searching graduation project ideas | | | | | | | | | | | |
| Finding graduation project idea | | | | | | | | | | | |
| **Information Gathering** | | | | | | | | | | | |
| Information about Blockchain &Smart Contract | | | | | | | | | | | |
| Palestinian Land Authority interviews | | | | | | | | | | | |
| Understanding the Current System | | | | | | | | | | | |
| Search For Related Work | | | | | | | | | | | |
| Attend Online Blockchain Course (1) | | | | | | | | | | | |
| Attend Practical Online Blockchain Course (2) | | | | | | | | | | | |
| Learn React | | | | | | | | | | | |
| Learn Node JS | | | | | | | | | | | |
| **Prepare work Environment** | | | | | | | | | | | |
| Blockchain platform Selection & Usage dependences | | | | | | | | | | | |
| Web development Environment | | | | | | | | | | | |
| **Coding and Programming** | | | | | | | | | | | |
| Write Smart Contract Code in Solidity | | | | | | | | | | | |
| Deploy Smart Contract into local Ganache network | | | | | | | | | | | |
| API Back-end Server implementation to handle the requests | | | | | | | | | | | |
| Embedded Web3 JS to user interfaces for local testing | | | | | | | | | | | |
| **Testing** | | | | | | | | | | | |
| Test smart contract | | | | | | | | | | | |
| Test functionality of each method | | | | | | | | | | | |
| Test API | | | | | | | | | | | |
| Test user interfaces | | | | | | | | | | | |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Evaluation** | | | | | | | | | | | |
| Usability Evaluation | | | | | | | | | | | |
| Functional Evaluation | | | | | | | | | | | |
| Performance Evaluation | | | | | | | | | | | |
| **Documentation** | | | | | | | | | | | |
| Final Report | | | | | | | | | | | |

## 4.2 SYSTEM DESIGN

In this section we clarify the Proposed Solution Design.

This chapter divide this chapter in four important parts:

- Proposed Solution Overview
- System Architecture Diagram
- Proposed Solutions Diagram
- Summary

### 4.2.1 Proposed Solution Overview

Through the process of analyzing the direct buying and selling process of the PLA system. We noticed in the sequence of the process steps that fraud may occur in one of the parts of the contract. Among the solutions was the use of Blockchain technology in the proposed solution and the work of a model that includes two stages:

- Operational smart contract stage: At this stage, all parts of the contract from the beginning of the registration process to the final exit are stored in the Blockchain.

- Smart contract archiving stage: the final contract resulting from the operational smart contract process is stored in the Blockchain

### 4.2.2 System Architecture Diagram

Figure 4-1 According to the diagram below, our solution will provide a group of web services that will be integrated with the PLA system. The PLA system will send a request to the web services each time an action (add/ update/ delete/ retrieve) is performed in order to keep the data up to date and validate each action. The web service will contact the Blockchain application to either add new data to the chain or validate transactions.

Once the validity of the data checked, and the Blockchain responds either with validation status or retrieves data, the web service will format the data, and return only the needed fields. This feature will allow us to add rules and user authorization, to allow or prevent different users from performing different actions

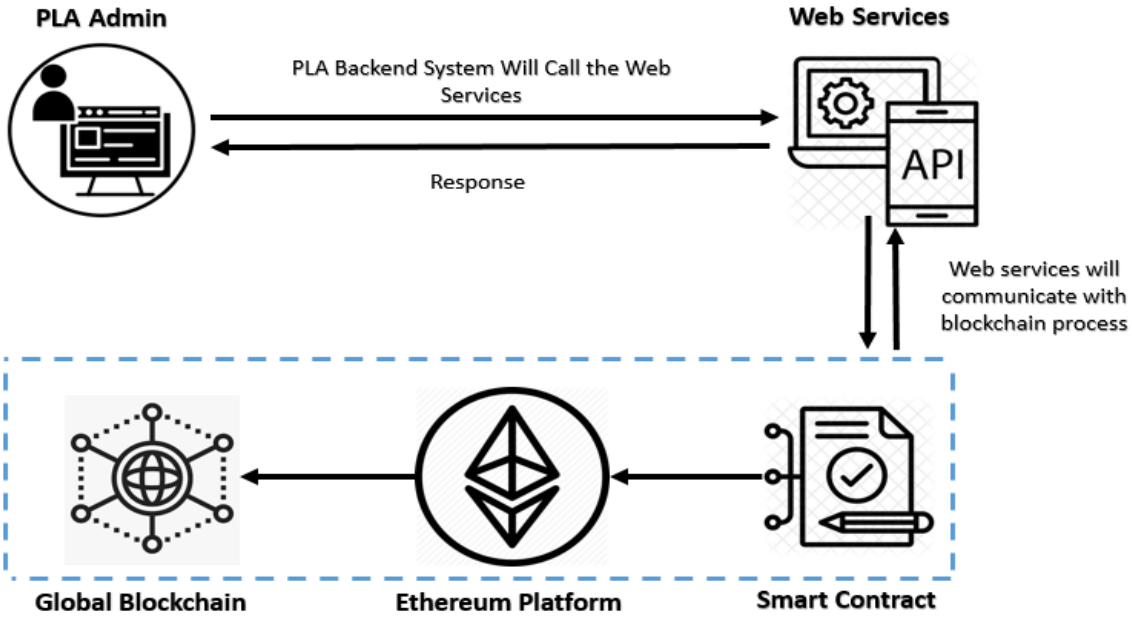



**Figure 4-1: System Architecture Diagram**

## Proposed Solution Diagram

In the Figure 4-2 this figure shows the structure of the proposed solution as previously mentioned, the solution is divided into two stages:

- Operational smart contract stage
- Smart contract archiving stage

So that in Operational Smart Contract the writer will open a transaction by creating a sale contract, he /she will enter the data for the sales contract and then save it in the Blockchain

After that, the seller and buyer data will be entered by the writer and the data will be saved in the Blockchain.

Then the contract will reach the Reception, which will put the order number and transaction number to produce the final contract. This contract will be stored in the Blockchain

Then the final contract resulting from the stage Operational Smart Contract will be taken, saved, and stored in the Blockchain

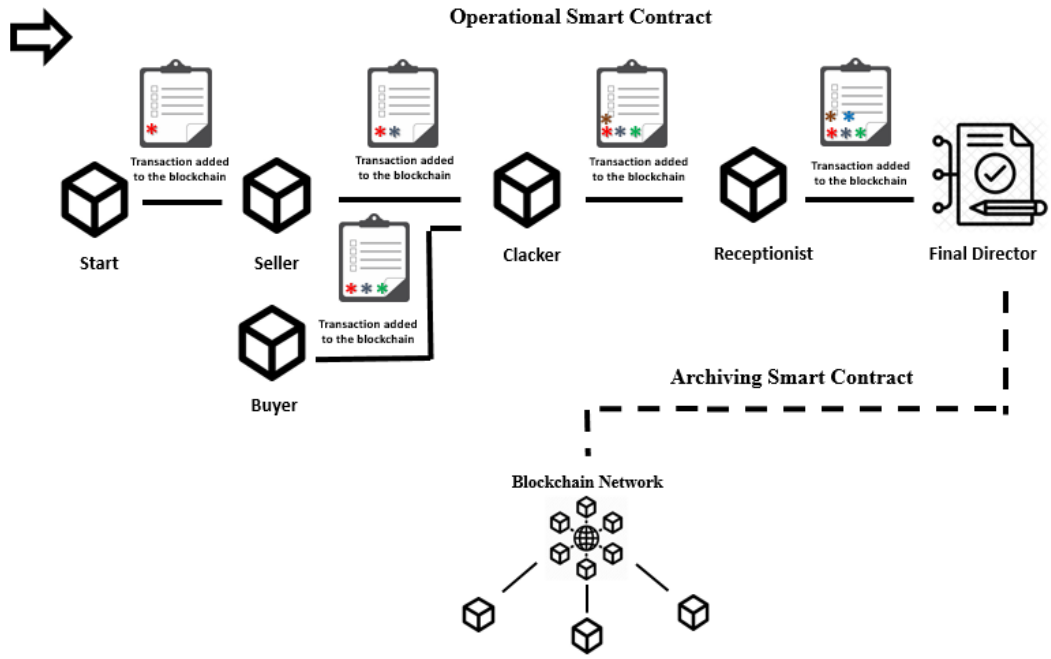



**Figure 4-2: Proposed Salutation Diagram**

## Operational Smart Contract Stage:

In this section, we have clarified the sequence of saving the buying and selling process according to the following schemes and clarifying the nature of the stored data (land sales contract, buyer, seller, content contract, attached document, final contract) where these stages will be stored in a Blockchain so called it the operational smart contract .

The beginning of the process starts with the writer, In Figure 4-3 clacker do the following functions:

Add Contract of Sell Land → Add Buyer → Add Seller → Add Content Contract → Add Attached Document

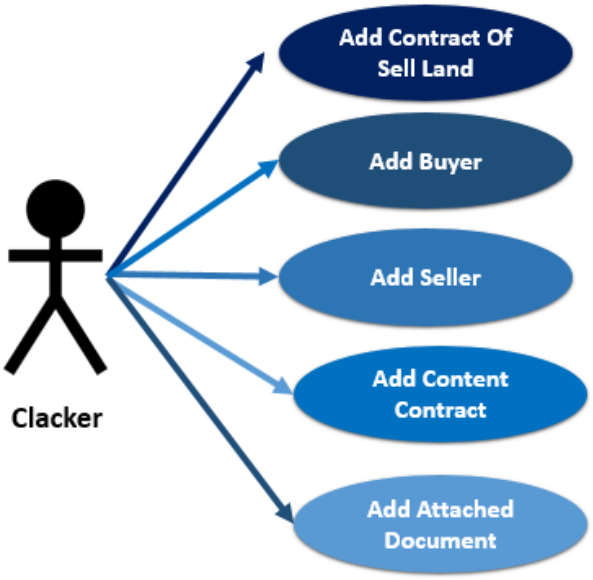



**Figure 4-3: Usecase Diagram for Clacker**

Where he/she will add the sales contract, which contains the following data (city, cost, taxCcearNumber, mobile number) So that after the addition process, it will be saved and stored in the Blockchain.

Then he will add the seller's data, which includes the following data (id Numbers Buyer, first Name Buyer, father Name Buyer, Grandfather Buyer, last Name Buyer, relation) after the addition process, the data will be saved and stored in the Blockchain.

Then he will add the seller's data, which includes the following data (id Numbers Seller, first Name Seller, father Name Seller, Grandfather Seller, last Name Seller, relation) after the addition process, the data will be saved and stored in the Blockchain.

After adding the previous data, the writer will enter data related to the contract text, which includes (Piece, coupon, area, and note) after the addition process, the data will be saved and stored in the Blockchain.

Then the writer will add the attachments for the buying and selling process. The attachments include (inventoryOfTheLagacyHash, taxFreeCertificateHash, idPhotoHash, enteryExtractHash, agenciesHash)

So that the writer will upload the custom attachments, and then a hash will be stored for these attachments where each attachment has its own hash where IPFS it was used, and it is the one that converts the attachment into a hash. This means the hashes of attachments are stored and saved in the Blockchain. When retrieving attachments, the hash of the attachments stored in the Blockchain is taken to be then retrieved through Online IPFS where the hash is placed in https://ipfs.infura-ipfs.io/ipfs/hash and then you will search for the hash that has already been stored in the Blockchain.

After the clacker's role in creating the contract ends, the role of the Reception is begin, who adds the most important parts of the contract. It is (order number, transaction number).

In Figure 4-4 Reception does the following functions:

Add Order Number → Add Transaction Number→ Return to clacker → View Data → Transaction Approve

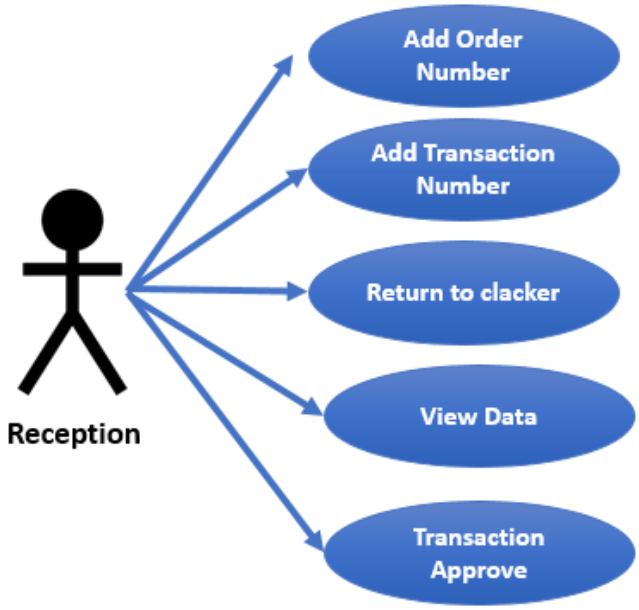



**Figure 4-4: Usecase Diagram for Reception**

Where after adding an order number, transaction number from the Reception to the contract, the data of which was taken from the writer this will be the final contract of the buying and selling process, which will be stored in the Blockchain

This diagram shows a summary of the Operational Smart Contract Stage

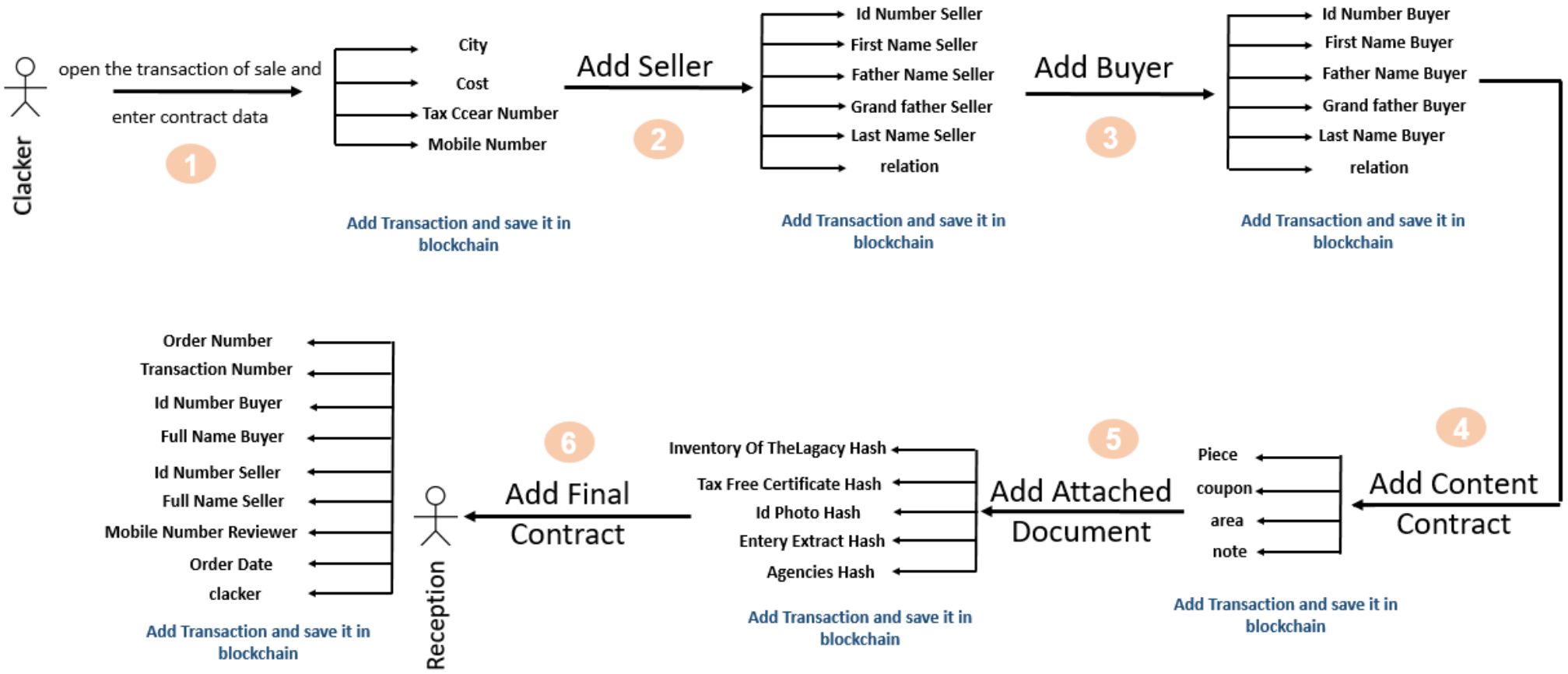



**Figure 4-5: Operational Smart Contract Summary Diagram**

### Archiving Smart Contract Stage:

At this point, the nodes are stored in their final form resulting from the previous process (the operational smart contract) which contains (orderNumber , transactionNumber , idNumberBuyer , fullNameBuyer , idNumberSeller , fullNameSeller , clerks , orderDate , moblieNumberReviwer) and then stored and published in the Blockchain .

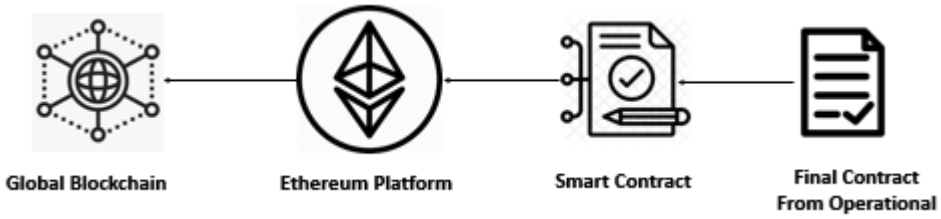



**Figure 4-6: Archiving Smart Contract Diagram**

## 4.2.3 Row Map

In the section, we have explained row mapping to implement the solution that we explained earlier so the following will be done

- Implementation: We will build the smart for the proposed solution.

  Then we will build an API Backend developed to build web services that interact with smart contract. Show the UML for API Backend endpoints.

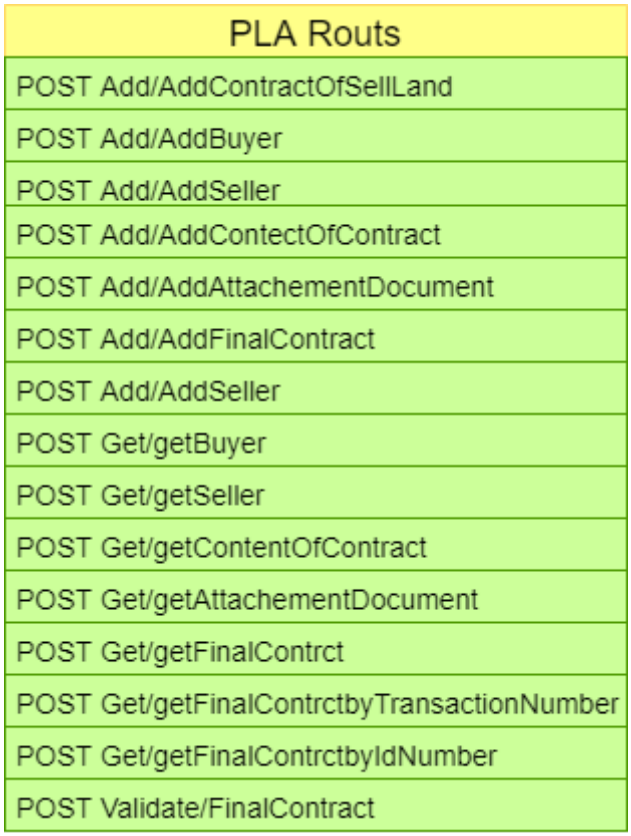



**Figure 4-7: UML for backend API endpoints**

- Testing: Through it, we will show that the proposed solution works correctly and as expected and the practical experiments we have done

- Evaluation: Through it, we explain the practical results and evaluate them through review three measures in order to evaluate the proposed system performance.

    - Usability Evaluation

    - Functional Evaluation

    - Performance Evaluation

## 4.3 SUMMARY

In this chapter, we talked about the mechanism used to implement our solution, through which we explained the time plan, and through which we explained the strategy used to implement the project.

We also explained the methodology used in implementing the project, where we used the Incremental prototype methodology, through which the existing system was dealt with.

Then we moved to talk about the proposed solution to the problem we talked about in the first quarter, as this solution is the result of a deep analysis and continuous interviews with the Palestinian Land Authority.

So that we have illustrated a diagram showing the structure of the proposed solution, as our current system is integrated with the Palestinian Land Authority, so that the interfaces related to the buying and selling process were obtained.

Accordingly, a solution has been found

Then we moved on to a detailed explanation about the solution strategy, where we have two proposals for the solution

- Operational stage: At this stage, all parts of the contract are stored from the beginning of the registration process to the establishment of the contract between the two parties.
- The Archiving stage: the final contract resulting from the operational process is stored in the Blockchain as the final contract.

So that we explained in the Operational stage the sequence of keeping sales and purchase contracts, and each stage is added and stored in the block.

Then comes the final stage in which we explained the preservation of the final contract resulting from the Operational stage in the Blockchain.

Then at the end of the chapter we talked about the implementation mechanism and explained it as an introduction to the following chapters.

*Chapter Five*

# 5 IMPLEMENTATION AND TESTING

In this chapter, we will explain the practical implementation for building and programming the system and integration it to the existing system in the PLA. For the Implementation step is divided into three parts: Build Smart Contract, Build API, and Backend Server with PLA Interfaces. Also, the testing phase which ensures that all system Modules work correctly within an expected result.

## 5.1 ETHEREUM NETWORK IMPLEMENTATION

### 5.1.1 Write Smart Contract

Before starting to write the smart contract code, the PLA was contacted and coordinated with them by holding periodic interviews to obtain accurate information to build a smart contract. In agreement with them, the interfaces for the buying and selling process were obtained, based on which the writing process took place, as we converted the functionalities in these interfaces to a smart contract code.

The smart contract file has been created in the Contracts directory. This contract is written in Solidity programming language which is similar to the other object-oriented languages. Smart Contract exposes declarations of a Struct, a State Variable, Mapping, and Functions.

The Struct with the names of Contract parts which is a custom-defined type that groups of variables that we use later in mapping and functions .In our Proposed Solution we define six of the struct and in Figure 5-8 we clarified what these are and clarified the variables containing it. The State Variable that will store an array of names of Contract parts Struct. While the functions are divided into two important parts:

- **Add Functions:** There are six add functions which used to add into Blockchain, these functions used it to create the contract of buying and selling process. By creating an object from the names of Contract parts Struct. After that, the object is pushed into array.

  In the following diagrams, we explain the mechanism of dealing with addition functions:

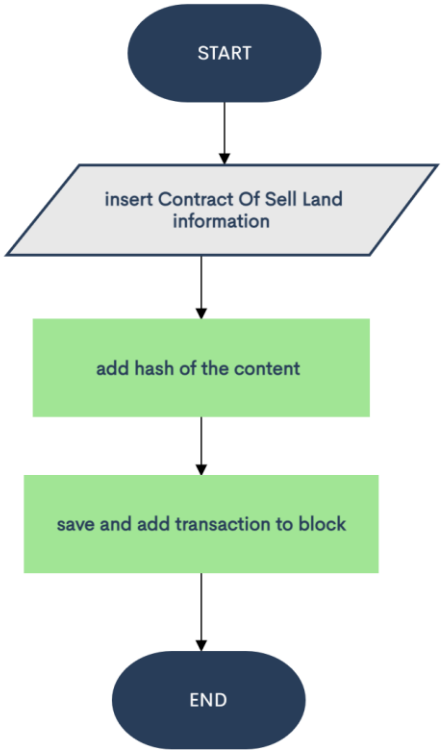



**Figure 5-1: flowchart for the process of adding a contract of selling a land**

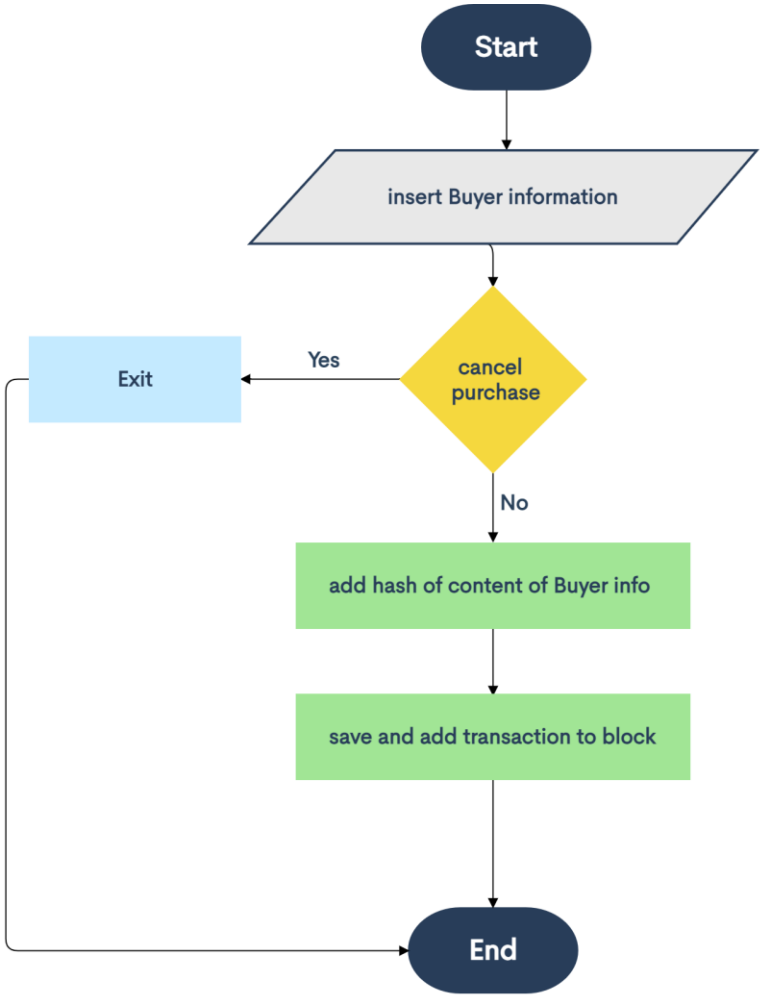



**Figure 5-2: flowchart for the process of adding a Buyer**

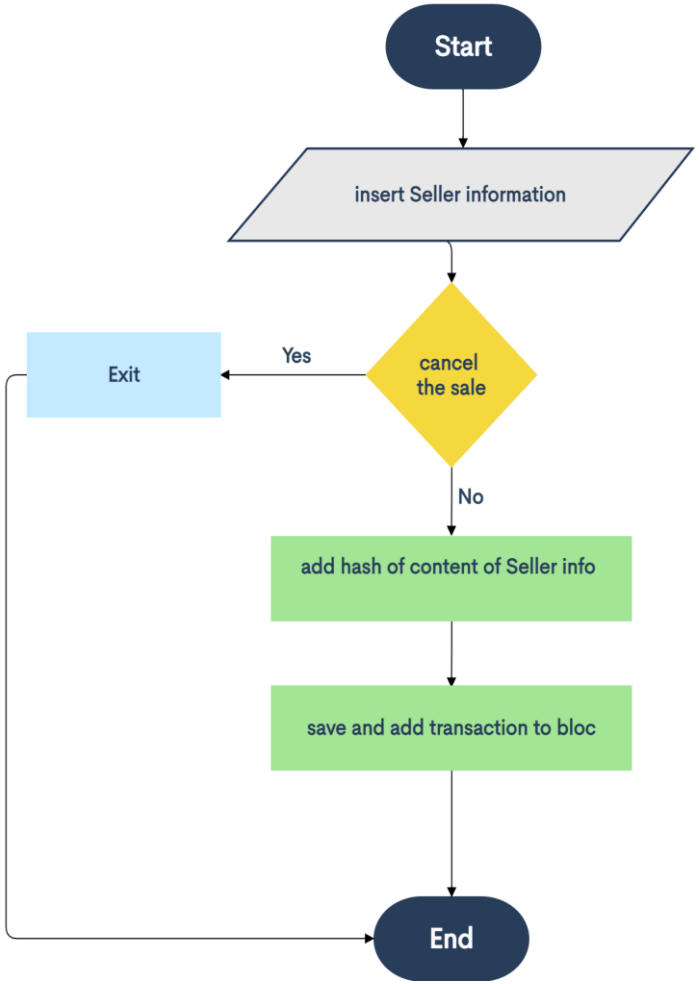



**Figure 5-3: flowchart for the process of adding a Seller**

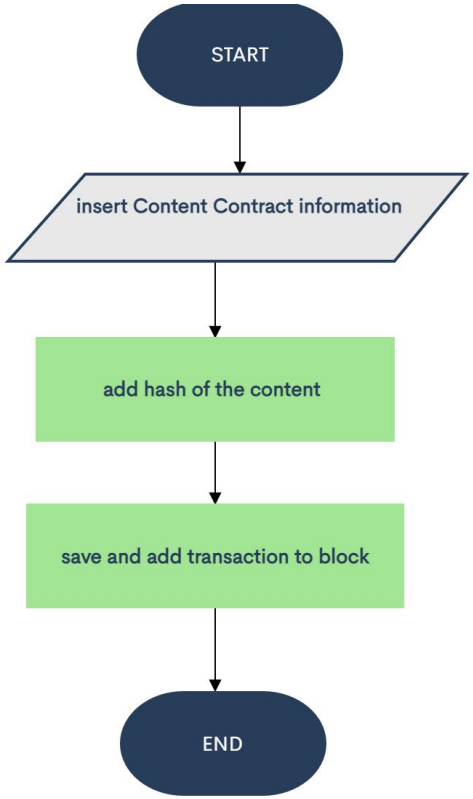



**Figure 5-4:flowchart for the process of adding a Content Contract**

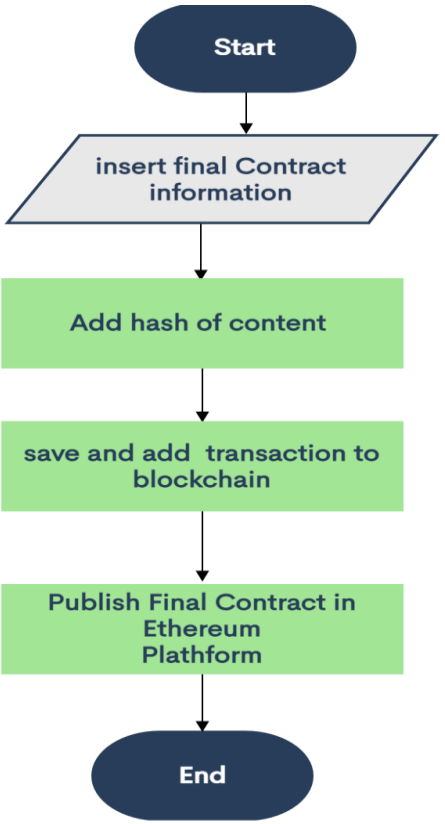



**Figure 5-5: flowchart for the process of adding a Final Contract**

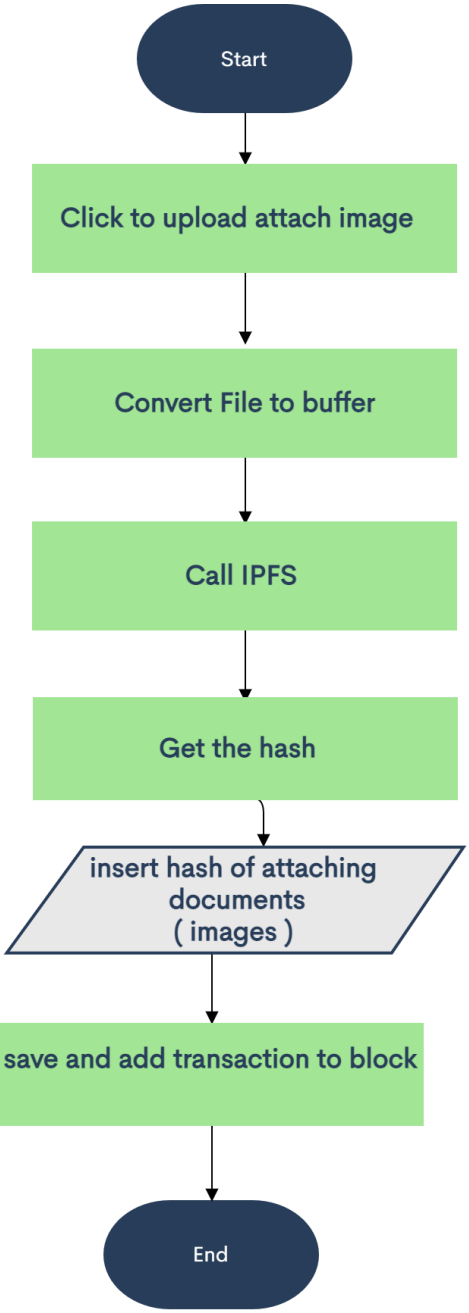



**Figure 5-6: flowchart for the process of adding an Attach Document**

- **Get Functions:** These Functions used to display the info of contract by entering id number or transaction number then it will return the contract

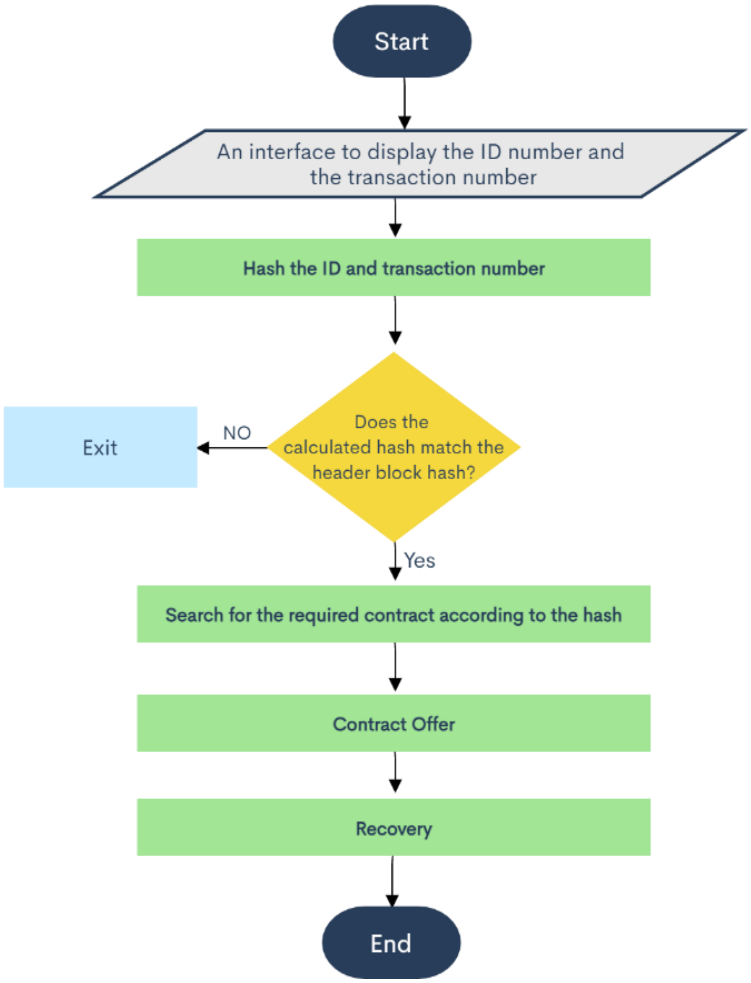



**Figure 5-7: flowchart for the process of retrieving a contract**

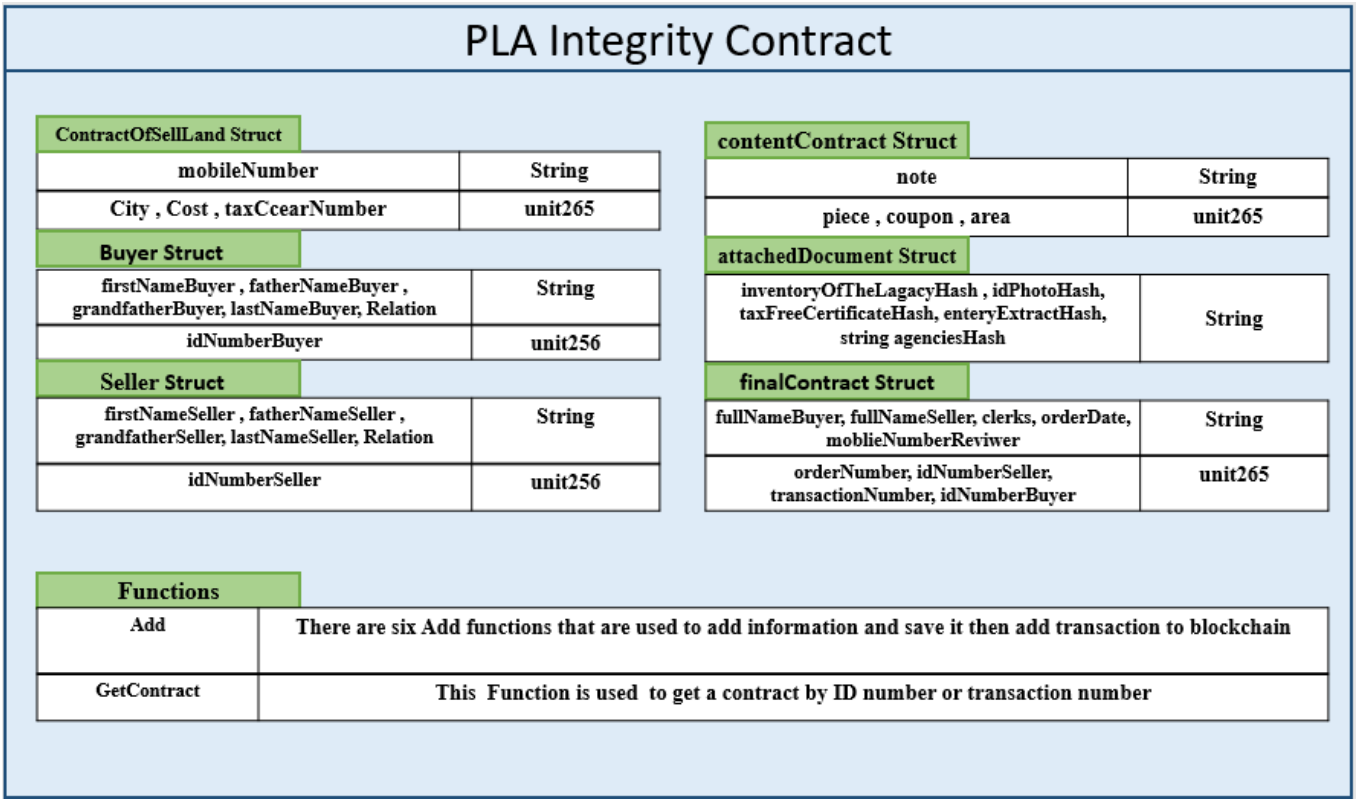



**Figure 5-8: Smart Contract Structure of Proposed Solution**

### 5.1.2 Deploy the Smart Contract

In order to deploy this contract onto a local Blockchain with Ganache, it's needed to add a ganache into the Truffle configurations.

The Truffle framework sets up three directories. [83]

- The first is called Contracts directory which contains all the contracts file.
- The second is migrations directory that used to deploy smart contract.
- The last directory is Test directory which used to test the functionality of the smart contract

The port number in the Truffle framework should be the same as the port number from the ganache. Moreover, it's needed to create a migration file which containing the path

of smart contract file. Figure 5-9 shows the Ganache port number configuration, while Figure 5-10 shows the Truffle Port number Configuration.

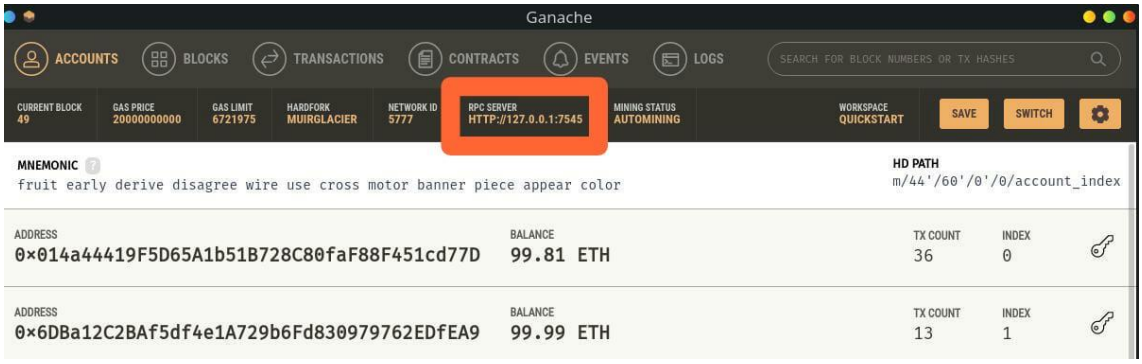



**Figure 5-9: Ganache port number configuration**

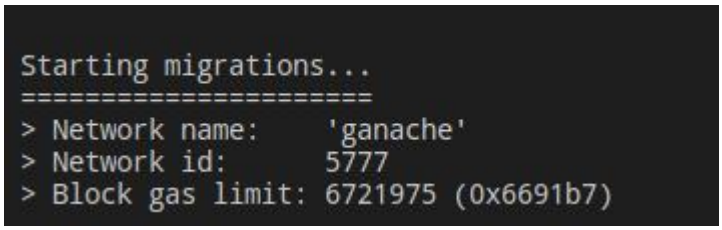



**Figure 5-10: Truffle configuration**

### 5.1.3 Ethereum Smart Contract Deployment Architecture

The architecture of smart contract includes main two parts:

- **Contract deployment:** Aims to deploy the smart contract on Ethereum Network. Deployment process begins with compile smart contract source code using Truffle framework. The result of compilation is Contract ABI which is going to be sent to Truffle migrate process that will deploy the smart contract into Ganache local network. The contract address will be the output of migration process.

- **Contract interaction:** Aims to interact with the Ethereum Smart Contract through Web3 JS by injecting this library into React UI pages and pass to it the address of deployment contact. Then any method on this contact can be called. After deploying the smart contract, it's needed to connect to the Blockchain network with personal account using an Ethereum wallet. This wallet used to interact and send transactions to this contract.

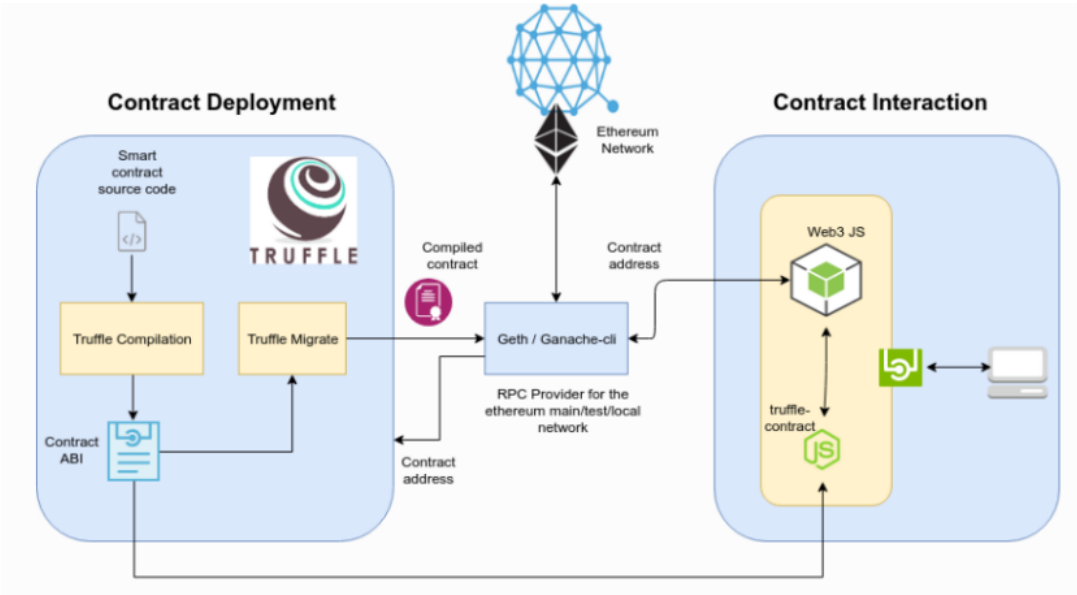



**Figure 5-11: Ethereum Smart Contract Deployment Architecture [84]**

## 5.2 CONTRACT VERIFICATION

In this process, we explained how to verify paper contracts and this is an added value to the PLA system so that in case of suspicion of any fraudulent contract, PLA Employees will fetch the data of this contract from the database, and then We will extract the final contract from the smart archiving contract so that we will retrieve it through the transaction number or identity number, and then compare between contract extract from database and contract extract from the Blockchain. Then define if the contract fraud or not.

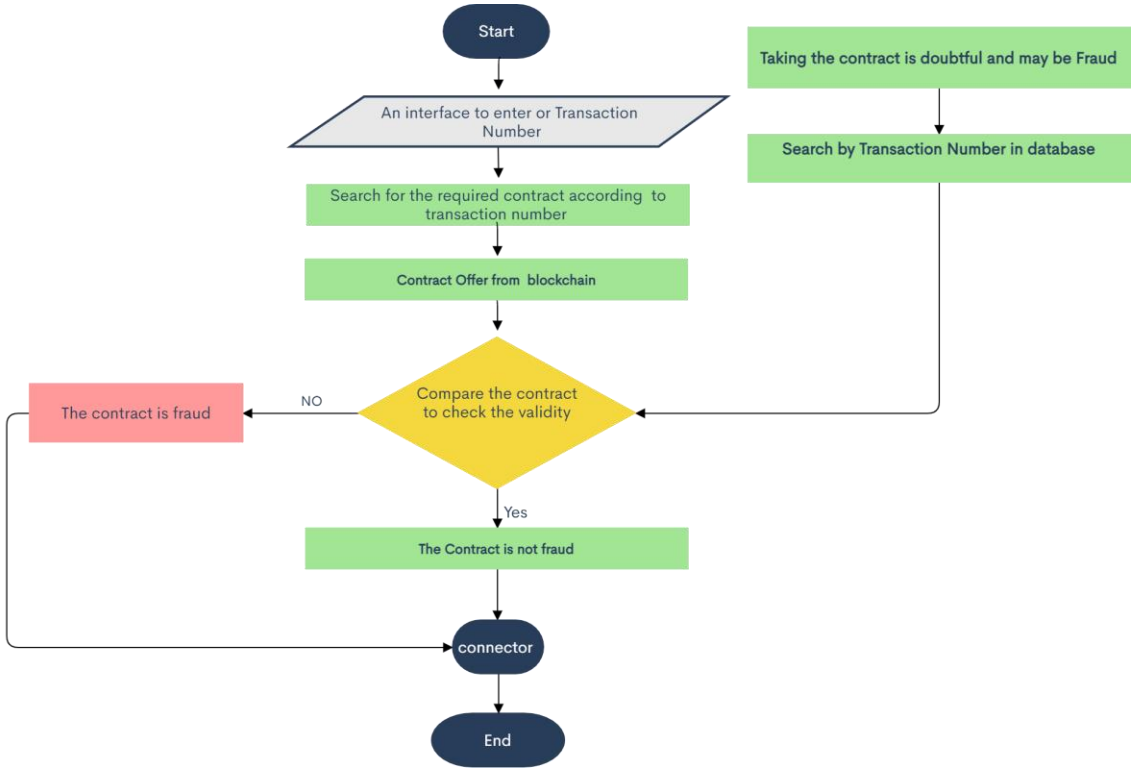



**Figure 5-12: flowchart for the Contract Verification**

## 5.3 API BACKEND IMPLEMENTATION

The API is built using node js so that data is sent as a post request from PLA Application Where the Application will send the data according to the endpoint that we provided it includes all the functions of recording the buying and selling process so that when the employee wants to add Contract Of Sell Land he/she will use" POST Add/AddCntractOfSellLand " do by placing the link in postman and pass the data, and then the smart contract receives this variable and pass them to Add Contract Of Sell Land functions so that it does the deploy process and store it in the Blockchain.

## 5.4 TESTING

This section reviews each component in the proposed system and ensures that it work correctly and as expected. In this case, testing procedures done on PC environment using test software applications will be mentioned. As well as, it explores the test results that summarize the testing phase of each functionality of the proposed system along with its state that shows how successful each operation is

We define our test in three important steps:

- Smart Contract Test
- API Backend Test
- Verification Test
- Blockchain Test

### 5.4.1 Smart Contract Testing

It is necessary before deploying the smart contract on the Blockchain network to test it and ensure it does not have any bugs that can be exploited because there is no way to rewriting the underlying code.

In order to do that, a Remix Integrated Development Environment was used to test the functionality of the smart contract. This done by opening the Remix browser and copying the smart contract to it, this will automatically compile it. If there are any warnings or errors, it will be displayed in the right panel.

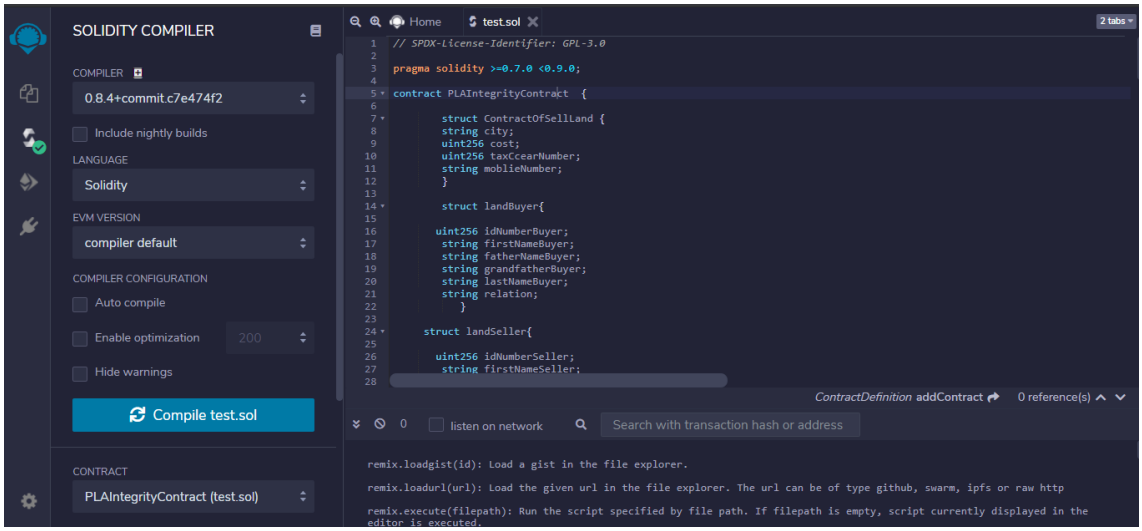



**Figure 5-13: Smart Contract Code Compile**

Then using JavaScript VM that will run the contract directly in the browser using a JavaScript implementation of the EVM and select the account will be used to deploy the contract.

After that we test all functions. They will add transactions and save to the Blockchain network.

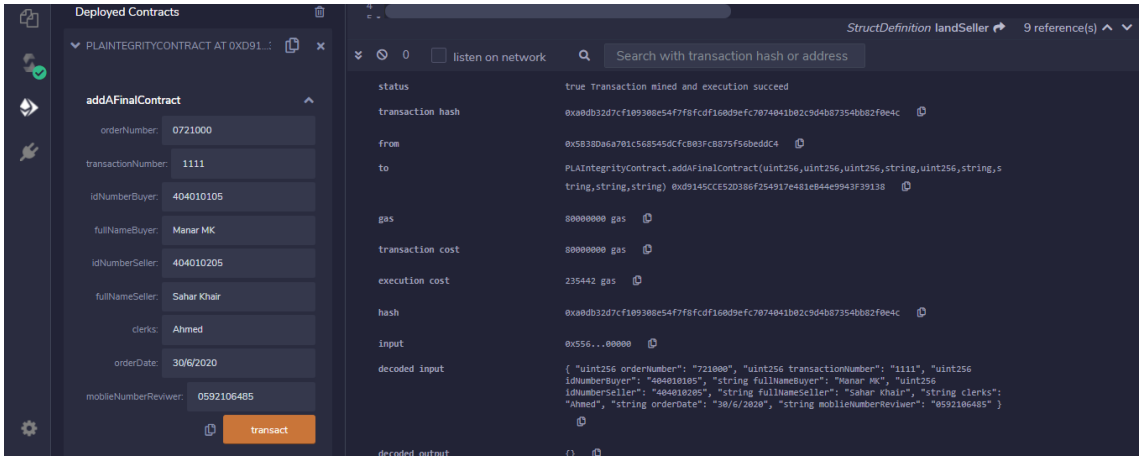



**Figure 5-14: Deploy Smart Contract on Remix**

Next, after making sure that the smart contract code does what it expects. We make similar interface as PLA interface to ensure the process do well as expected all deploying process which will deploy the smart contract on the Ganache network.

Where we used React to make these interfaces and then we used Web3 to be injecting to Smart Contract Code.

We used Ganache, Truffle, and Metamask to deal with the Ethereum network locally

We used a visual code editor. Then to run the project we opened a terminal to type the following command:

- Npm start server

Then will open the following interface as shown in Figure 5-15

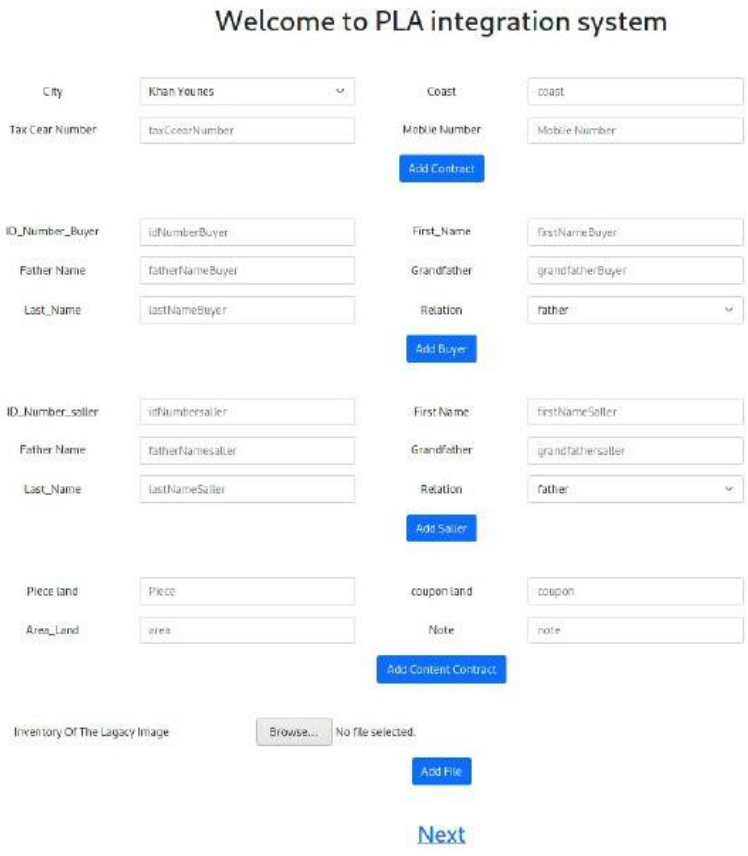



**Figure 5-15: React Local Test Interface**

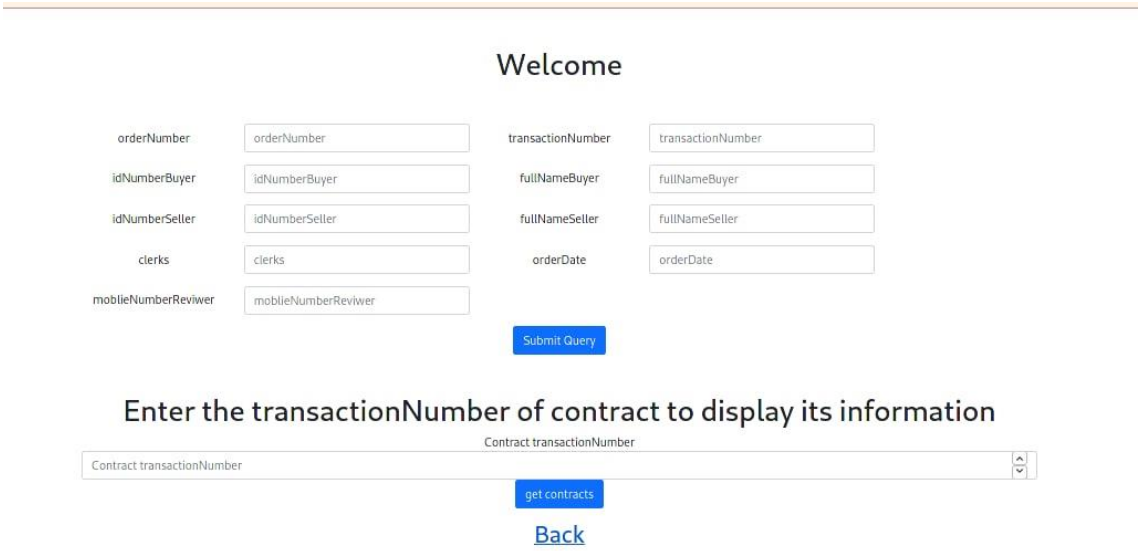



**Figure 5-16: React Local Test Interface**

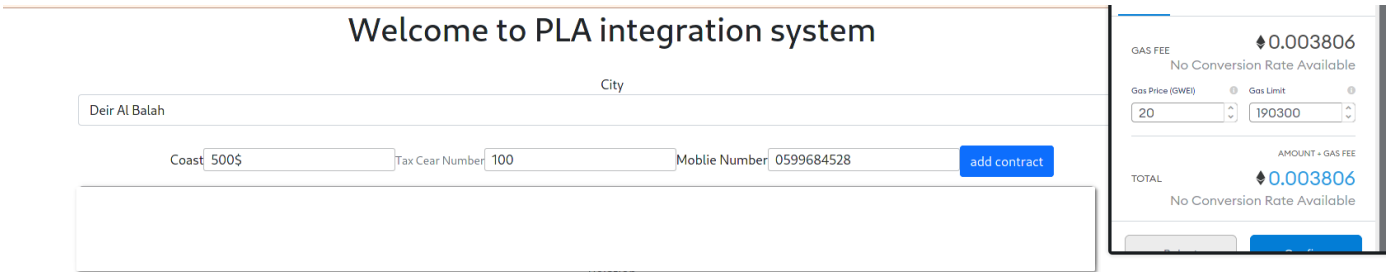



**Figure 5-18: Add Contract of Sell Land**


**Figure 5-17: Add Seller**


**Figure 5-20: Add Content Contract**


**Figure 5-19: Add Buyer in Blockchain**

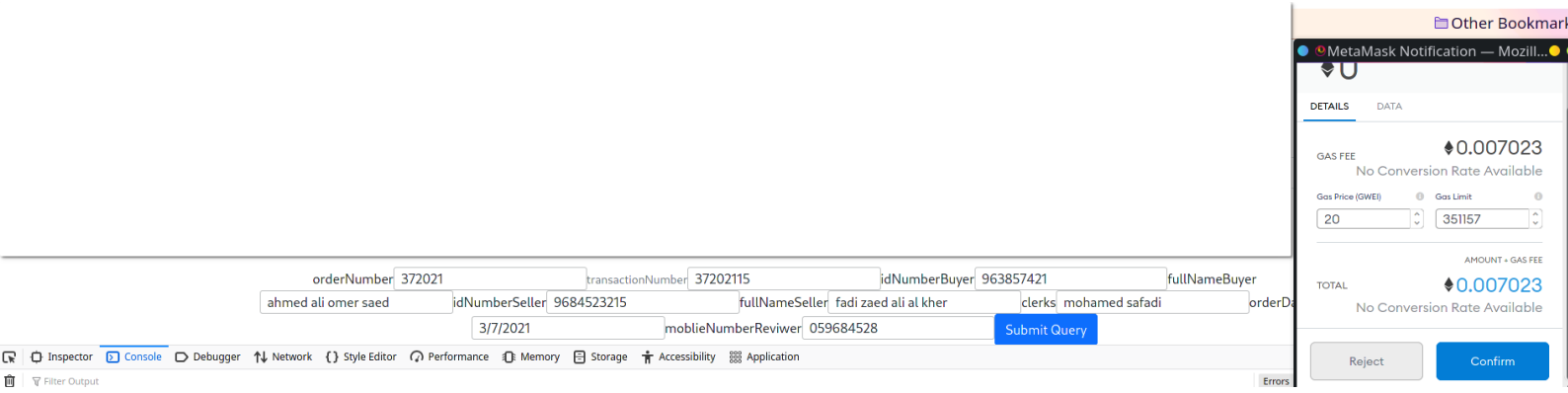



**Figure 5-21: Add Final Contract**

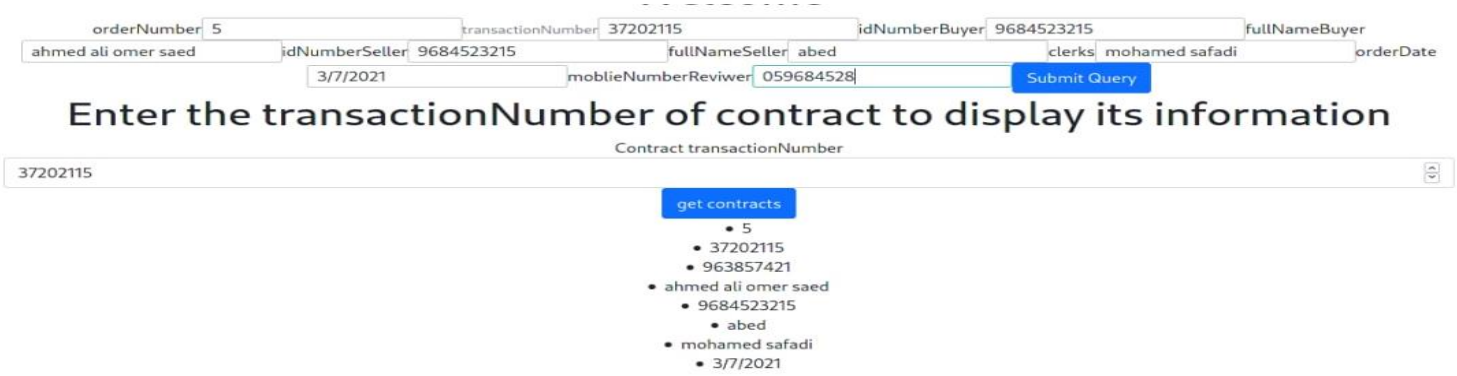



**Figure 5-22: Retrieve Final Contract by Transaction Number**

### 5.4.2 API Backend Testing

Test if the API code works correctly, which used to verify the server running well and correctness of the work of each endpoint (Add, Retrieve, and Validate). The Postman software was used to send a POST request with the JSON file containing the values of our Functions shows the Backend server listen on port 6000 waiting for a request

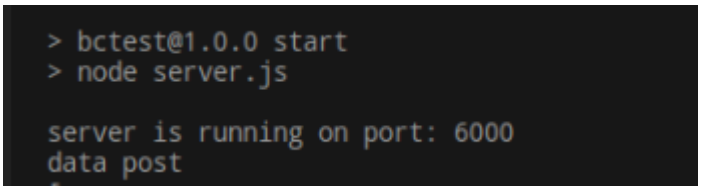



**Figure 5-23: API Server Running**

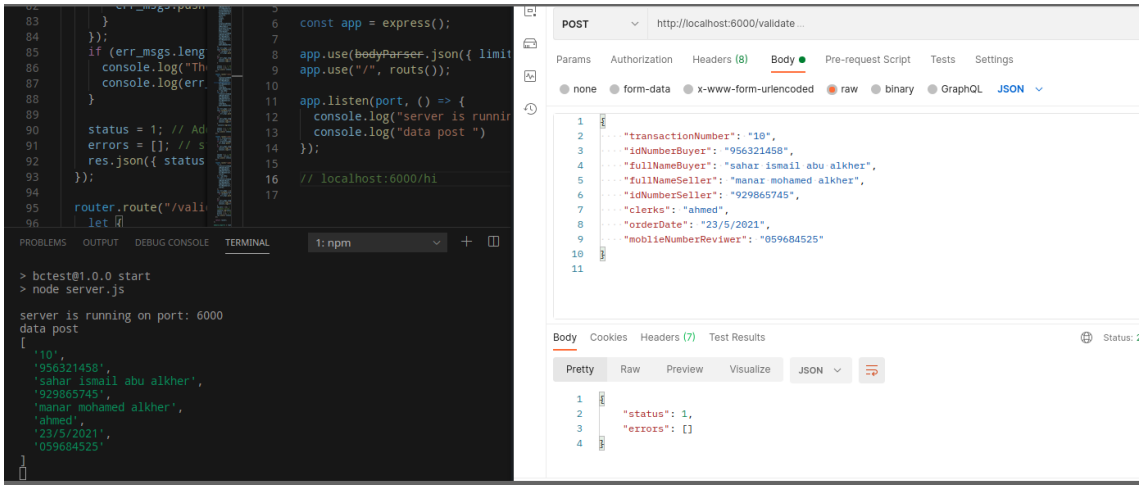



**Figure 5-24: API Test Function**

### 5.4.3 Verifications Testing

In this section, we explain the practical results of the verification process, as well as the experiments we conducted

**First experience:**

When we validated the paper contract, we uploaded the contract and hashed it using IPFS Assuming that the contract is complete and in its correct form then we retrieved the stored contract from the Blockchain through the transaction number and we took it as an image and made a hash for it and compare the hash for both contracts but this experiment did not work

**Second experience:**

In this experiment, we created a paper contract data entry form, and then made a hash of the input data .Then we retrieved the nodes stored in the Blockchain and hashed these retrieved data Then we compare the hash of data retrieved from the Blockchain with the hash of data from paper nodes.

This operation worked, but in our view, there is a more effective method that we explained in the next section, but we did not implement it because we did not get permissions from PLA

**Our Assumption:**

Here we made the following hypothesis So that the paper contract information is stored in the database, where we will make a request On Validate Function located in Web Services And then we're going to do request on the Blockchain.

The Web Service will make a comparison between Database and Blockchain automatically by comparing the data Here we will be able to detect whether the contract is forged or not.

In this way, we can ensure the safety of the Database and preserve the integrity of the data Thus, we can print a valid and valid contract

### 5.4.4 Blockchain Testing

In this section, we explained a series of practical experiments to prove the strength of the Blockchain where we have changed the settings related to Ganache which represents local Blockchain where through Figure 5-25 we modified the mining block time and noticed that we were unable to add any operations due to the time difference.

We also changed the server settings and manipulated them according to the Figure 5-26

The result is our inability to modify and deal with the Blockchain

This explains how successful the Blockchain is in detecting any modification or change process.

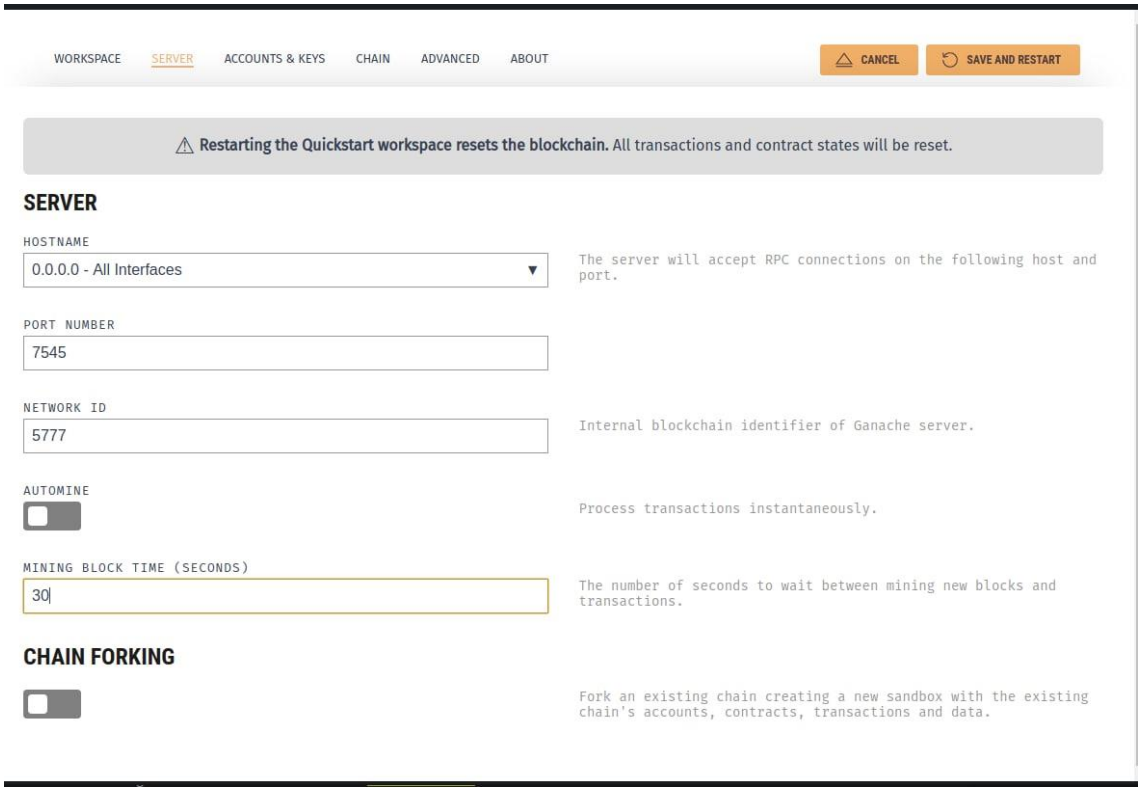



**Figure 5-25: Changing the mining block time**

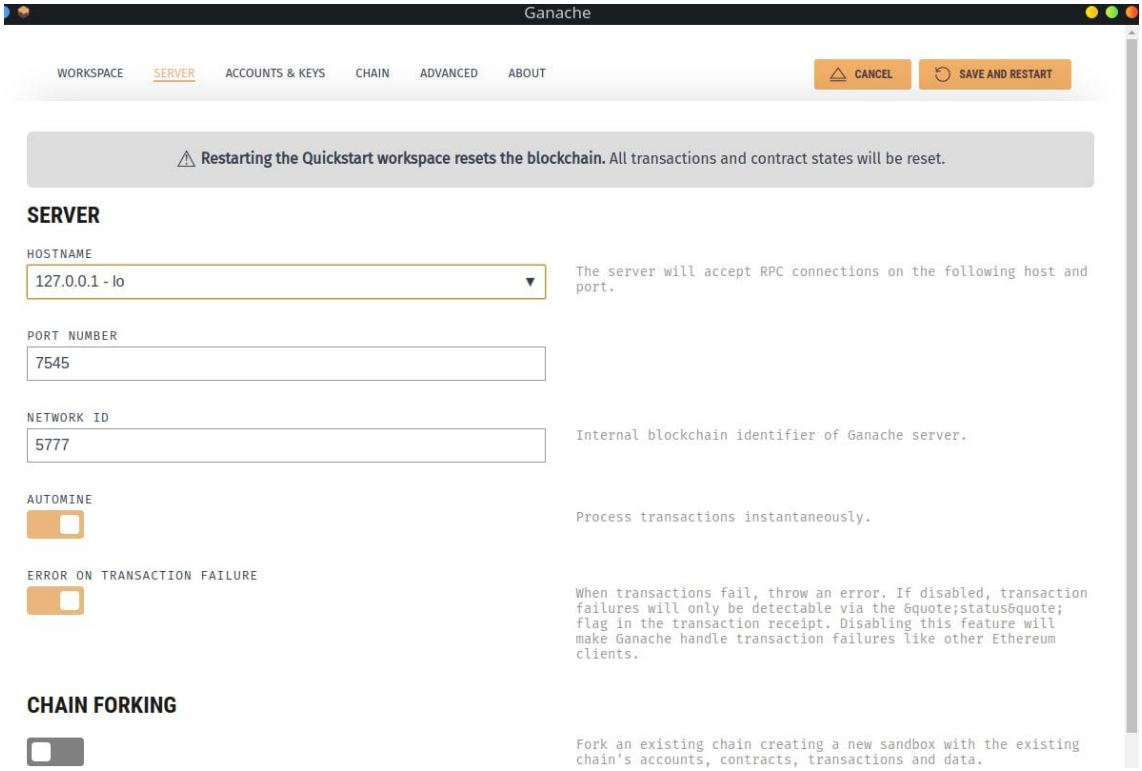



**Figure 5-26: Changing the Server Configuration**

## 5.5 SUMMARY

In this chapter, we talked about the practical procedures of our project, through which we explained how to build smart contract code through illustrative diagrams of how the functions work.

We also explained the structure of the code, where we explained the variables and functions through an explanatory diagram that expresses this

Then we moved on to talk about the deployment process, through which we show the effectiveness and validity of the code, and explained the nature of the tools used in the deployment process.

Then we talked about the process of building an API and running its server, where we explained how to build and run it

Then we moved to talk about the connection process, which was the last stage of our project, through which work was done to hand it over to the Palestinian Land Authority.

At the end of the chapter, we talked about running testing where we did a local test where we simulated the power interfaces to make sure that our system is working properly

While the PLA test, we explained it over the next captor by measurement evaluation

*Chapter Six*

# 6 EVALUATION

In this chapter, we talked about the most important results of this study based on information collected from questionnaires for employees of the Palestinian Land Authority.

The evaluation criteria section reviews three measures in order to evaluate the proposed system performance.

- Usability Evaluation
- Functional Evaluation
- Performance Evaluation

## 6.1 USABILITY EVALUATION

In this section, the proposed solution is measured according to its effectiveness and whether it performs the required functions. As those who answered these questions are the employees of the Palestinian Land Authority

The Following tables display ratings for each person who independently filled out the survey

**Table 6-1:Statistics1 from Clacker X**

| Question | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Is the service provided smooth to deal with? | | | * | | |
| Are you performing the required functions? | | | | | * |
| Is the service difficult to use? | | * | | | |
| Was the service provided quick responsive | | | | * | |

**Table 6-2: Statistics2 from Clacker Y**

| Question | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Is the service provided smooth to deal with? | | * | | | |
| Are you performing the required functions? | | | | | * |
| Is the service difficult to use? | | * | | | |
| Was the service provided quick responsive | | | | | * |

**Table 6-3:Statistics3 from Reception X**

| Question | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Is the service provided smooth to deal with? | | * | | | |
| Are you performing the required functions? | | | | | * |
| Is the service difficult to use? | | | * | | |
| Was the service provided quick responsive | | | | | * |

**Table 6-4: Statistics3 from IT Employee X**

| Question | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Is the service provided smooth to deal with? | | | | * | |
| Are you performing the required functions? | | | | | * |
| Is the service difficult to use? | * | | | | |
| Was the service provided quick responsive | | | | | * |

**Table 6-5: Statistics3 from IT Employee X**

| Question | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Is the service provided smooth to deal with? | | | | * | |
| Are you performing the required functions? | | | | * | |
| Is the service difficult to use? | * | | | | |
| Was the service provided quick responsive | | | | | * |

**Table 6-6: Average Score (Out of 5)**

| Question | Average Score |
|---|---|
| Is the service provided smooth to deal with? | 3 |
| Are you performing the required functions? | 4.8 |
| Is the service difficult to use? | 1.4 |
| Was the service provided quick responsive | 4.8 |

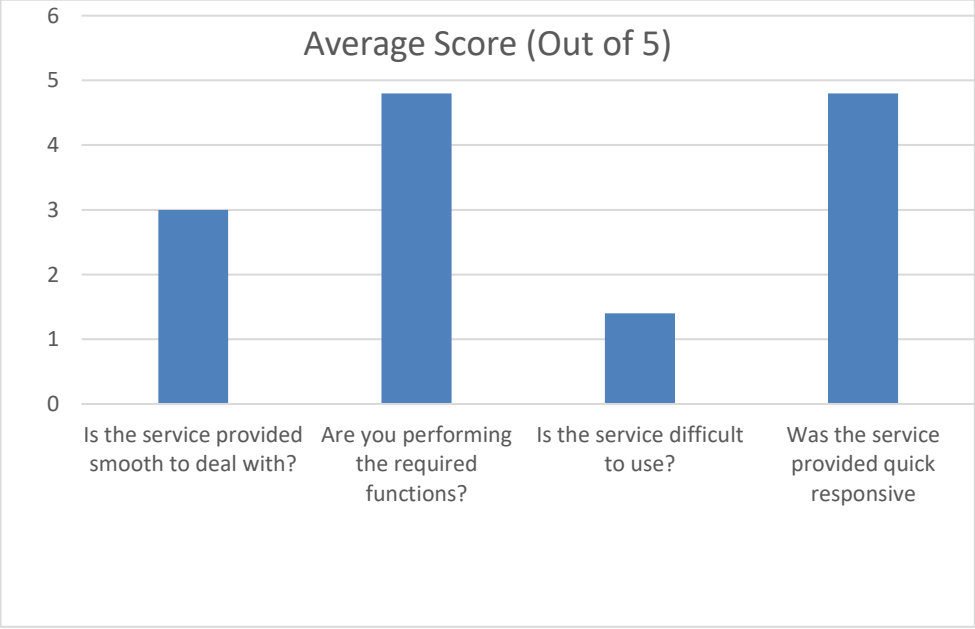



**Figure 6-1: Average Score (Out of 5)**

## 6.2 FUNCTIONAL EVALUATION

In this section, we have customized the measurement in terms of functionality, and also introduced it to PLA employees.

This is a special questionnaire to ensure that the jobs are performing the required jobs. The Following table display ratings for each person who independently filled out the survey

**Table 6-7: Statistics1 from Clacker X and Y**

| | Question | Yes | No |
|---|---|---|---|
| Add Functions | Add Contract Of Sell Land | * | |
| | Add Buyer | | |
| | Add Seller | | |
| | Add Content contract | | |
| | Add Attach Document | | |
| Retrieve Functions | Retrieve Contract Of Sell Land | * | |
| | Retrieve Buyer | | |
| | Retrieve Seller | | |
| | Retrieve Content contract | | |
| | Retrieve Attach Document | | |

**Table 6-8: Statistics1 from Reception X and Y**

| | Question | Yes | No |
|---|---|---|---|
| Add Functions | Add Transaction Number | * | |
| | Add Order Number | | |
| | Add Final Contract | | |
| Retrieve Functions | Retrieve Final Contract | * | |

**Table 6-9: Statistics1 from IT Employee X**

| | Question | Yes | No |
|---|---|---|---|
| Add Functions | Add Contract Of Sell Land | | |
| | Add Buyer | | |

| | | | |
|---|---|---|---|
| | Add Seller | * | |
| | Add Content contract | | |
| | Add Attach Document | | |
| Retrieve Functions | Retrieve Contract Of Sell Land | * | |
| | Retrieve Buyer | | |
| | Retrieve Seller | | |
| | Retrieve Content contract | | |
| | Retrieve Attach Document | | |

**Table 6-10: Statistics1 from Reception IT Employee**

| | Question | Yes | No |
|---|---|---|---|
| Add Functions | Add Transaction Number | * | |
| | Add Order Number | | |
| | Add Final Contract | | |
| Retrieve Functions | Retrieve Final Contract | * | |

## 6.3 PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed solution and the current system of the PLA In terms of time and cost.

Note: read process from Ethereum network free, JUST write process need cost

**Table 6-11: Statistics to show the cost for each process**

| | Time in Seconds | Cost in Dollar | Cost in Gas[1] |
|---|---|---|---|
| Add Contract Of Sell Land | 5-10 | 0.023461 | 0.003806 |
| Add Buyer | 5-15 | 0.031578 | 0.005126 |
| Add Seller | 5-15 | 0.031578 | 0.005126 |
| Add Attachment Document | 10-20 | 0.023461 | 0.003806 |
| Add Content Contract | 10-20 | 0.023009 | 0.003753 |
| Add Final Contract | 15-25 | 0.043265 | 0.003806 |
| Total | ~70 | 0.176352 | 0.025423 |

Table 6-12 shows the performance comparison between the current system and the proposed solution

**Table 6-12: Performance comparison between current system and proposed solution**

| | Current system | Our Solution |
|---|---|---|
| Time | 1 -2 Weeks | 1 hour |
| Probability of fraud | High | Very low |
| Security | Middle | High |

[1] 1 GAS = 6.114976 Dollar

## 6.4 SUMMARY

In this section, we talked about the effectiveness of our system, as we made questionnaires and submitted them to Palestinian Land Authority employees after using the web services.

Measured based on:

- Usability Evaluation
- Functional Evaluation
- Performance Evaluation

In using the system, the result differed because the writer and the future dealt with technology and weak techniques, while with the It employee His/her dealings were flexible.

The result of these measurements showed the efficiency of the proposed solution in saving the sequence of the buying and selling process and then storing it in the Blockchain.

*Chapter Seven*

# 7 CONCLUSION AND FUTURE WORK

In this chapter, and according to our in-depth study of the proposed solution, we have clarified the future directions that we aspire to implement to improve the performance and quality of the existing proposed solution.

Where we mentioned four future plans focused on developing the solution and improving its quality.

Then, at the conclusion of this report, we moved to clarify the most important consequences, as we clarify the relationship between the proposed solution and between research papers close to our system and between the proposed solution and the current system in the Palestinian Land Authority

## 7.1 CONCLUSION

Blockchain in the land registry has been an essential aspect of today's world. Once the land transfer task is completed, the information is automatically updated and saved on this Blockchain platform, and this process is the most secure and tamper-free mode of the operating system. No one can change the legal right of ownership, no one can harm the origin of the data others cannot make a change to that transaction and ownership

The idea behind this solution is to develop a model based on Blockchain technology to store and validate contracts (case study of direct purchase and sale of land in PLA).

Our Solution explored the methods of storing contracts in the Blockchain and examining the integrity of the data in them, which is done in a fast and secure manner compared to the system currently used in the Palestinian Land Authority.

**Table 7-1: Conclusion Result between PLA System and Proposed Solation**

| | PLA | Contracts Validity Based On Blockchain |
|---|---|---|
| **Reliability** | Contracts are stored in a paper archive for the Land Authority, while keeping electronic copies by photocopying them and saving images electronically, and the data can be changed easily.<br><br>Save contract data in Database | Contract is stored in multiple places in the Blockchain based system. Consensus mechanisms ensure that only information changes when all relevant parties agree |
| **Ease of access** | Access to contracts is difficult because they are stored in paper in the archive. The contract is searched by searching for the contract serial number. | Contracts stored in multiple nodes, promote easy and fast access<br><br>index search method<br>And it is fast in the case of inquiring about contracts |
| **Reducing corruption** | A person, whether from internal or external parties, can falsify or view the contract. | Distributed ledgers storage allows for preventing corruption in land ownership due to the secure and reliable mechanism of transfer and changing ownership records |
| **Data integrity and higher quality** | A person, whether from internal or external parties, can falsify or view the contract. | Not being accessed by hackers or unauthorized users means the authenticity of the saved data |

| | | |
|---|---|---|
| | Any attack on Database can modify the data without the knowledge of the administrators | |
| **Reliability** | Access to contracts is difficult because they are stored in paper in the archive. The contract is searched by searching for the contract serial number. | Contracts stored in multiple nodes, promote easy and fast access |

Through the results of this study, it was found that this proposed solution has outperformed the similar projects presented in the second chapter in terms of accuracy, performance and implementation time. However, the proposed system has not been deployed in the real Blockchain environment, because it needs a specific budget to write the transactions used to store the contracts and their data.

**Table 7-2: Conclusion Result between Similar Related Work and Proposed Solation**

| | **Blockchain-based framework for secure and reliable land registry system** | **Contracts Validity Based On Blockchain** |
|---|---|---|
| **The aim** | System based on Blockchain technology to verify the validity of contracts (the study case is direct buying and selling of land). | A secure and reliable framework for a land registry system using Blockchain. The proposed framework uses the concept of a smart contract at different stages of the land registry and gives an algorithm for prior agreement. |

| | | |
|---|---|---|
| **Verification method in the traditional system** | Verified by checking contract serial number | verified by checking physical registers |
| **Issues with Traditional System** | The existence of many contracts forgery, whether<br>from internal or external parties | In the existing land registry system, the land registrar cannot verify any existing dues on the land and registrar will act assuming that all the land dues are clear. |
| **Benefits of using Blockchain smart contracts<br>(strength point)** | Transparency<br><br>Increased trust<br><br>Reliability: Data is stored in multiple places in the Blockchain based system. Consensus mechanisms ensure that only information changes when all relevant parties agree<br><br>Security: Data is stored at multiple systems using encryption methods that stop the data from being altered without proper authentication.<br>Ease of access: Information stored at multiple nodes, enhances easy and speedy access<br><br>Integrity<br>Reducing corruption<br><br>Error reduction | The decentralized standard system for land registration records will reduce the role of intermediaries, decrease the time and cost of the process and the system will strengthen the process and will build the trust between the transacting parties. Recording rights of property through Blockchain will help in annual cost savings and provide tamper-proof ledger book. Blockchain usually reduces the expenditure and time of registration process and will help Judiciary to settle down most of the civil cases related to property. |

| | | |
|---|---|---|
| **Weak points** | When adding the attachments related to the contract, we assume that they are correct So that we will take a hash of the attachments after scanning them. | The system focused on the security of contract payment methods and they did not focus on keeping the contract from forgery. |

## 7.2 FUTURE WORK

### 7.2.1 Use Private Blockchain

In our solution, the experiment was conducted on the Ethereum platform, where we noticed during the operations we carry out, specifically the addition operations, that it costs in the transactions process

Found the most suitable solution to use Private Blockchain for the following reasons:

- Transactions are free
- Lesser nodes verify transactions, verification by just owner of the Blockchain.
- The owner (PLA Employees) can control who has access to Blockchain. This ensures that the servers of the institution will do the mining process.

### 7.2.2 Improve the process of saving images and attachments

In our proposed solution, when adding the attachments to the contract, we attach them as pictures.

Where we use IPFS to make a hash of the image and then store the hash in the Blockchain we have proven our success in this, but we do not guarantee the attached pictures that changes will occur over time (fold the paper for example).

Therefore, we suggest that the attachments be dealt with using Image Processing to ensure that the extracted information is correct, as we will extract the fields for the images and store them in the Blockchain.

### 7.2.3 Local IPFS

In the part of the attached documents, we deal with IPFS so that we upload the attachments related to the contract

Then you will IPFS make a hash of these attachments, the hash of the attachments will be stored in the Blockchain.

In order to retrieve these attachments, we will deal with Online IPFS So that the hash of the attachments stored in the Blockchain will be taken and placed in the form of https://ipfs.infura-ipfs.io/ipfs/ (hash of attachments in Blockchain) Then Online IPFS search for this hash and then it will display these attachments.

Therefore, we suggest using local IPFS this mean install IPFS and configure it in local pc

### 7.2.4 Expand the Scope of the Proposed Solution Operations

In this paper, we have customized the process used in the proposed solution, as we have customized the direct buying and selling process as there are many processes in the PLA which require a deep understanding of each process.

After our success in the direct buying and selling process, we saw the expansion of the solution to include other operations. Where the sales process alone includes different branches [Selling by agency, Selling on condition of living]

Therefore, we will expand the project to include various aspects of the sales process and other transactions that occur in the Palestinian Land Authority

# 8 REFERENCES


[1] J. Luke Conway, “Blockchain Explained,” Investopedia, 1 June 2021. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp. [Accessed 2021 June 26].

[2] A. Rosic, “What is Blockchain Technology? A Step-by-Step Guide For Beginners,” Blockgeeks, 25 11 2020. [Online]. Available: https://blockgeeks.com/guides/what-is-blockchain-technology/. [Accessed 25 March 2021].

[3] R. T. V. Y. K. H. T. Keerthana1, “Blockchain and Smart Contract,” in *Integration of Digital Certificate Blockchain and Overall Behavioural Analysis using QR and Smart* , Issue-3, Volume-2, 2019, p. 267.

[4] A. Lastovetska, “Blockchain Architecture Basics: Components, Structure, Benefits & Creation,” MLSDev, 5 January 2021. [Online]. Available: https://mlsdev.com/blog/156-how-to-build-your-own-blockchain-architecture. [Accessed 25 March 2021].

[5] J. M. LUKE CONWAY, “Investopedia,” Investopedia, 1 June 2021. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp. [Accessed 26 June 2021].

[6] L. Conway, “Blockchain Explained,” Investopedia, 1 June 2021. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp. [Accessed 25 March 2021].

[7] “When To Use Blockchain Technology?,” 101 Blockchains, 29 June 2020. [Online]. Available: https://101blockchains.com/when-to-use-blockchain/. [Accessed 25 March 2021].

[8] J. Jake Frankenfield, “Consensus Mechanism (Cryptocurrency),” Investopedia, 29 July 2021. [Online]. Available: https://www.investopedia.com/terms/c/consensus-mechanism-cryptocurrency.asp. [Accessed 26 March 2021].

[9] N. JOSHI, “8 blockchain consensus mechanisms you should know about,” Allerin, 23 April 2019 . [Online]. Available: https://www.allerin.com/blog/8-blockchain-consensus-mechanisms-you-should-know-about. [Accessed 26 March 2021].

[10] C. GONG, “5 Types of Blockchain Consensus Mechanisms,” LOGIC SOLUATIONS, 16 2018 2018. [Online]. Available: https://www.logicsolutions.com/5-types-blockchain-consensus-mechanisms/. [Accessed 26 MARCH 2021].

[11] D. Dog, “Proof-of-Work vs. Proof-of-Stake,” ALEXANDRIA, 10 September 2020. [Online]. Available: https://coinmarketcap.com/alexandria/article/proof-of-work-vs-proof-of-stake. [Accessed 26 March 2021].

[12] C. GONG, “5 Types of Blockchain Consensus Mechanisms,” LOGIC SOLUATIONS, 16 April 2018. [Online]. Available: https://www.logicsolutions.com/5-types-blockchain-consensus-mechanisms/. [Accessed 26 March 2021].

[13] N. JOSHI, “8 blockchain consensus mechanisms you should know about,” Allerin, 23 April 2021. [Online]. Available: https://www.allerin.com/blog/8-blockchain-consensus-mechanisms-you-should-know-about. [Accessed 26 March 2021].

[14] N. JOSHI, “8 blockchain consensus mechanisms you should know about,” Allerin, 23 April 2018. [Online]. Available: https://www.allerin.com/blog/8-blockchain-consensus-mechanisms-you-should-know-about. [Accessed 26 March 2021].

[15] N. JOSHI, “8 blockchain consensus mechanisms you should know about,” Allerin, 23 April 2019. [Online]. Available: https://www.allerin.com/blog/8-blockchain-consensus-mechanisms-you-should-know-about. [Accessed 26 March 2021].

[16] M. A. ,. R. ,. M. Mohammad Shojafar, “Types Of Blockchain,” in *Blockchain for Cybersecurity and Privacy*, London , New York, CRC Press, 2020, pp. 176-177.

[17] C. Press, “Types of Blockchain Cybersecurity and Privacy,” in *Blockchain for* , London, New York, CRC Press, 2020, pp. 176-177.

[18] C. Parizo, “What are the 4 different types of blockchain technology?,” TechTarget, 21 May 2021. [Online]. Available: https://searchcio.techtarget.com/feature/What-are-the-4-different-types-of-blockchain-technology. [Accessed 27 March 2020].

[19] co-author, “Types of Blockchains,” Types of Blockchains, 2020. [Online]. Available: https://devopedia.org/types-of-blockchains. [Accessed 2020 March 27].

[20] co-author, “TOP BLOCKCHAIN PLATFORMS OF 2021,” LeewayHertz, 2021. [Online]. Available: https://www.leewayhertz.com/blockchain-platforms-for-top-blockchain-companies/. [Accessed 27 March 2021].

[21] T. T. A. D. L. C. Ji Wang, “BLOCKBENCH: A Framework for Analyzing Private Blockchains,” ResearchGate, March 2017. [Online]. Available: https://www.researchgate.net/publication/314943090_BLOCKBENCH_A_Framework_for_Analyzing_Private_Blockchains. [Accessed 27 2020 March].

[22] co-author, “What Is a Database?,” ORACLE, 2021. [Online]. Available: https://www.oracle.com/database/what-is-database/. [Accessed 27 March 2021].

[23] G. Iredale, “Blockchain Vs Database: Understanding The Difference,” 101 Blockchain, 30 December 2020. [Online]. Available: https://101blockchains.com/blockchain-vs-database-the-difference/. [Accessed 27 March 2020].

[24] co-author, “Palastine Land Authority,” PLA, 2021. [Online]. Available: http://www.pla.gov.ps/ar/. [Accessed 1 January 2021].

[25] J. Frankenfield, “Decentralized Applications – dApps,” Investopedia, 22 June 2021. [Online]. Available: https://www.investopedia.com/terms/d/decentralized-applications-dapps.asp. [Accessed 1 May 2021].

[26] T. Economy, “Decentralized Applications – dApps,” Blockchainhub Berlin, July 2019. [Online]. Available: https://blockchainhub.net/decentralized-applications-dapps/. [Accessed 1 May 2021].

[27] ryangrunest, “INTRODUCTION TO DAPPS,” Ethereum, 11 June 2021. [Online]. Available: https://ethereum.org/en/developers/docs/dapps/. [Accessed 1 May 2021].

[28] co-author, “What are smart contracts on blockchain?,” IBM, 2021. [Online]. Available: https://www.ibm.com/topics/smart-contracts. [Accessed 1 May 2021].

[29] L. Mearian, “What's a smart contract (and how does it work)?,” ComputerWorld, 29 July 2019. [Online]. Available: https://www.computerworld.com/article/3412140/whats-a-smart-contract-and-how-does-it-work.html. [Accessed 1 May 2021].

[30] co-authors, “What are smart contracts on blockchain?,” IBM, 2021. [Online]. Available: https://www.ibm.com/topics/smart-contracts. [Accessed 1 May 2021].

[31] wslyvh, “INTRODUCTION TO SMART CONTRACTS,” Ethereum, 18 June 2021. [Online]. Available: https://ethereum.org/en/developers/docs/smart-contracts/. [Accessed 1 May 2021].

[32] S. Mehdi, “Smart Contracts vs. DApps: What's the difference?,” LinkedIn, 18 December 2017. [Online]. Available: https://www.linkedin.com/pulse/smart-contracts-vs-dapps-whats-difference-shams-mehdi. [Accessed 2 May 2021].

[33] J. Frankenfield, “Ethereum,” Investopedia, 4 June 2021. [Online]. Available: https://www.investopedia.com/terms/e/ethereum.asp. [Accessed 2 May 2021].

[34] co-author, “Jump start your crypto portfolio,” coinbase, May 2021. [Online]. Available: https://www.coinbase.com/. [Accessed 2 May 2021].

[35] co-author, “CMC,” What is ethereum?, 2021. [Online]. Available: https://www.cmcmarkets.com/en/learn-cryptocurrencies/what-is-ethereum. [Accessed 2 May 2021].

[36] co-author, “What is the Ethereum platform?,” AR+, 2020. [Online]. Available: https://ar.cryptonews.com/guides/what-is-ethereum.htm. [Accessed 10 May 2020].

[37] co-author, “What is the Ethereum platform?,” 2020. [Online]. Available: https://ar.cryptonews.com/guides/what-is-ethereum.html. [Accessed 10 May 2020].

[38] co-auther, “WHAT IS ETHER (ETH)?,” Ethereum, 2021. [Online]. Available: https://ethereum.org/en/eth/. [Accessed 2 May 2021].

[39] J. Frankenfield, “Ether (ETH),” Jake Frankenfield, 24 Feb 2021. [Online]. Available: https://www.investopedia.com/terms/e/ether-cryptocurrency.asp. [Accessed 2 May 2021].

[40] co-author, “Ethereum,” wikipedia, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Ethereum. [Accessed 8 May 2021].

[41] J. Frankenfield, “Gas (Ethereum),” Investopedia, 26 May 2021. [Online]. Available: https://www.investopedia.com/terms/g/gas-ethereum.asp. [Accessed 3 May 2021].

[42] wackerow, “GAS AND FEES,” Etheteum, 22 June 2021. [Online]. Available: https://ethereum.org/en/developers/docs/gas/. [Accessed 3 May 2021].

[43] Emiliano, “L16_Dapp_Ecosystem,” GIOVANNI MAROTTA - UNIVERSITY OF CATANIA, DMI , 2020.

[44] co-author, “Solidity,” Solidity, 2021. [Online]. Available: https://soliditylang.org/. [Accessed 3 May 2021].

[45] co-author, “Solidity Tutorial,” tutorialspoint, 2021. [Online]. Available: https://www.tutorialspoint.com/solidity/index.htm. [Accessed 3 May 2021].

[46] co-author, “Welcome to Remix’s documentation!,” Remix, 2021. [Online]. Available: https://remix-ide.readthedocs.io/en/latest/. [Accessed 3 May 2021].

[47] co-author, “A crypto wallet & gateway to blockchain apps,” Metamask, 2021. [Online]. Available: https://metamask.io/. [Accessed 3 May 2021].

[48] W. Vermaak, “What Is MetaMask?,” Alexendria, 31 March 2021. [Online]. Available: https://coinmarketcap.com/alexandria/article/what-is-metamask. [Accessed 3 May 2021].

[49] co-author, “TRUFFLE SUITE,” TRUFFLE OVERVIEW, 2021. [Online]. Available: https://www.trufflesuite.com/docs/truffle/overview. [Accessed 3 May 2021].

[50] https://www.upgrad.com/blog/what-is-truffle-suite/, “What is Truffle Suite? Features, How to Install, How to Run Smart Contracts,” upGrade blog, 24 March 2021. [Online]. Available: https://www.upgrad.com/blog/what-is-truffle-suite/. [Accessed 3 May 2021].

[51] co-author, “What is IPFS?,” IPFS, 2021. [Online]. Available: https://docs.ipfs.io/concepts/what-is-ipfs/#decentralization. [Accessed 3 May 2021].

[52] co-author, “What are Web Services? Architecture, Types, Example,” GURV99, 2021. [Online]. Available: https://www.guru99.com/web-service-architecture.html. [Accessed 3 May 2021].

[53] T. Cleo, “What Are Web Services? Easy-to-Learn Concepts with Examples,” Cleo, 2021. [Online]. Available: https://www.cleo.com/blog/knowledge-base-web-services. [Accessed 3 May 2021].

[54] A. D. ByCamille Siegel, “What is an API?,” API Friends, 28 October 2020. [Online]. Available: https://apifriends.com/api-management/what-is-an-api/. [Accessed 3 May 2021].

[55] D. Gavilanes, “What is BlazeMeter, What is Postman, and How to Get Started with API Testing,” BlazeMeter, 16 March 2021. [Online]. Available: https://www.blazemeter.com/blog/what-is-postman-and-how-to-use-it-for-api-testing/. [Accessed 3 May 2021].

[56] co-author, “What is an API? A Digestible Definition with API Examples for Ecommerce Owners,” BIGCOMMERCE, 2021. [Online]. Available: https://www.bigcommerce.com/blog/what-is-an-api/#what-is-an-api. [Accessed 3 May 2021].

[57] co-author, “6 Differences between Web Services vs API (SOAP & REST examples),” Test Automation Resources, 21 May 2018. [Online]. Available: https://testautomationresources.com/api-testing/differences-web-services-api/. [Accessed 3 May 2021].

[58] R. Staff, “API vs Web Service,” Last Call, 20 April 2021. [Online]. Available: https://rapidapi.com/blog/api-vs-web-service/. [Accessed 2020 May 12].

[59] co-author, “json — JSON encoder and decoder,” python, 2021. [Online]. Available: https://docs.python.org/3/library/json.html. [Accessed 3 May 2021].

[60] co-author, “Working with JSON,” MDN WEB DOCS, 2021. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON. [Accessed 3 May 2021].

[61] co-author, “About Node.js,” Node Js , 2021. [Online]. Available: https://nodejs.org/en/about/. [Accessed 3 May 2021].

[62] co-author, “Node.js - Introduction,” tutorialspoint., 2021. [Online]. Available: https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm. [Accessed 3 May 2021].

[63] I. Feoktistov, “Why and When to Use Node.js?,” RELEVANT, 2021. [Online]. Available: https://relevant.software/blog/why-and-when-to-use-node-js/. [Accessed 3 May 2021].

[64] I. Fekotistov, “Why and When to Use Node.js?,” RELEVANT, 2021. [Online]. Available: https://relevant.software/blog/why-and-when-to-use-node-js/. [Accessed 3 May 2021].

[65] T. Hardy, “Top reasons to use Node.js for web application development,” Jaxenter, 29 March 2016. [Online]. Available: https://jaxenter.com/top-reasons-to-use-node-js-for-web-application-development-125144.html. [Accessed 3 May 2021].

[66] T. Hardy, “Top reasons to use Node.js for web application development,” Jaxenter, 29 March 2016 . [Online]. Available: https://jaxenter.com/top-reasons-to-use-node-js-for-web-application-development-125144.html. [Accessed 3 May 2021].

[67] co-author, “Tutorial: Intro to React,” React, 2021. [Online]. Available: https://reactjs.org/tutorial/tutorial.html. [Accessed 7 May 2021].

[68] N. Pandit, “What And Why React.js,” C#Corner, 10 Feb 2021. [Online]. Available: What And Why React.js. [Accessed 7 May 2021].

[69] co-author, “Express4.17.1,” Express, 2021. [Online]. Available: https://expressjs.com/. [Accessed 7 May 2021].

[70] co-author, “What is Express.js?,” Besant Technologies, 2021. [Online]. Available: https://www.besanttechnologies.com/what-is-expressjs. [Accessed 7 May 2021].

[71] co-author, “Express/Node introduction,” MDN Web Docs, 2021. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction. [Accessed 7 May 2021].

[72] co-author, “What is npm?,” w3schools, 2021. [Online]. Available: https://www.w3schools.com/whatis/whatis_npm.asp. [Accessed 7 May 2021].

[73] co-author, “What is npm?,” w3schools, 2021. [Online]. Available: https://www.w3schools.com/whatis/whatis_npm.asp. [Accessed 7 May 2021].

[74] D. Gavilanes, “What is BlazeMeter, What is Postman, and How to Get Started with API Testing,” BlazeMeter, 16 March 2021. [Online]. Available: https://www.blazemeter.com/blog/what-is-postman-and-how-to-use-it-for-api-testing. [Accessed 7 May 2021].

[75] co-author, “Postman Tutorial,” Javatpoin, 2021. [Online]. Available: https://www.javatpoint.com/postman. [Accessed 7 May 2021].

[76] co-author, “Postman Tutorial: How to use Postman Tool for API Testing,” GURV99, 2021. [Online]. Available: Postman Tutorial: How to use Postman Tool for API Testing. [Accessed 7 May 2021].

[77] E. T. I. Kongu Engineering College, “BlockChain enabled digitization of land Registration,” 2021.

[78] S. M. D. S. A. ,. W. Z. K. Mohammed Shuaib1, “Blockchain-based framework for secure and reliable land registry system,” 2021.

[79] M. ,. Y. K. ,. T. A. ,. G. K. Vinay Thakura, “Land records on Blockchain for implementation of Land Titling in India,” 2021.

[80] I. A. R.HasanbRajaJayaramanaKhaledSalahbMohammedOmara, “Implementing decentralized auctions using blockchain smart contracts,” 2021.

[81] A. Pinheiro and E. D. Canedo, “Monitoring File Integrity Using Blockchain and Smart Contracts,” 2020.

[82] F. Ullah, “A conceptual framework for blockchain smart contract adoption to manage real estate deals in smart cities,” 2020.

[83] co-author, “RUNNING MIGRATIONS,” trufflesuite., 2021. [Online]. Available: https://www.trufflesuite.com/docs/truffle/getting-started/running-migrations. [Accessed 2 June 2021].

[84] Dapp_Ecosystem, “UNIVERSITY OF CATANIA,” GIOVANNI MAROTTA, DMI, 2020.

[85] “Advantages of Blockchain,” Investopedia, 1 June 2021. [Online]. Available: https://www.investopedia.com/terms/b/blockchain.asp. [Accessed 25 March 2021].

[86] co-author, “web3.js - Ethereum JavaScript API,” web3.js, 2021. [Online]. Available: https://web3js.readthedocs.io/en/v1.3.4/. [Accessed 3 May 2021].

[87] co-author, “WEB3,” Dapp_Ecosystem, 2021. [Online]. Available: Dapp_Ecosystem. [Accessed 3 May 2021].

[88] S. E. Sommerville, “System Modeling,” CS 410/510 - Software Engineering, 2021. [Online]. Available: https://cs.ccsu.edu/~stan/classes/CS410/Notes16/05-SystemModeling.html. [Accessed 1 June 2021].

[89] Emiliano, “L16_Dapp_Ecosystem,” GIOVANNI MAROTTA - UNIVERSITY OF CATANIA, DMI, 2020.

[90] G. MAROTTA, “Dapp_Ecosystem,” UNIVERSITY OF CATANIA, DMI, 2020.

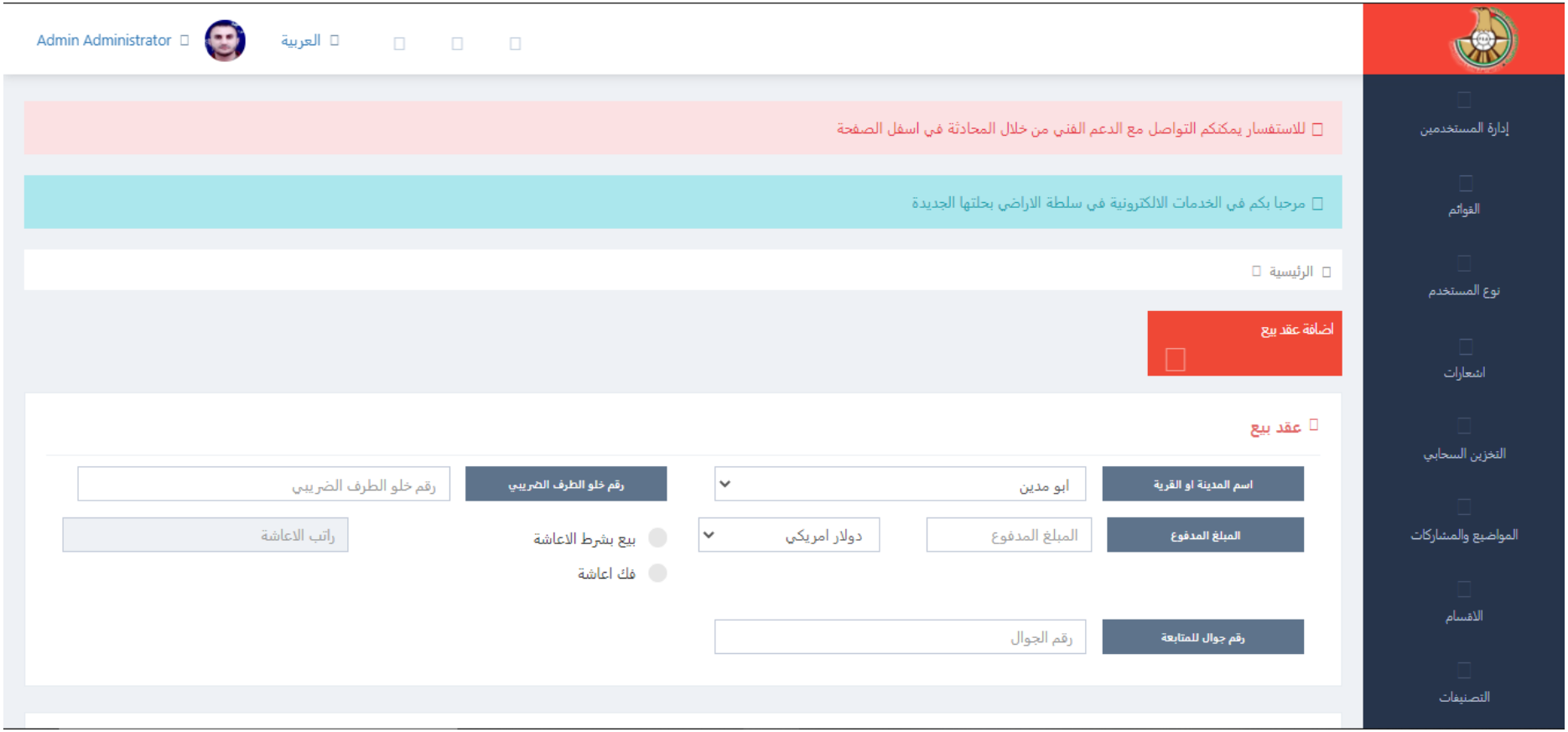
[91] G. MAROTTA, “L16_Dapp_Ecosystem,” UNIVERSITY OF CATANIA, DMI, 2020.
[92] G. MAROTTA, “Dapp_Ecosystem,” L16_Dapp_Ecosystem, DMI, 2020.
[93] G. MAROTTA, “Dapp_Ecosystem,” UNIVERSITY OF CATANIA, DMI, 2020.
[94] G. MAROTTA, “Dapp_Ecosystem,” UNIVERSITY OF CATANIA, DMI, 2020.

# APPENDICIES

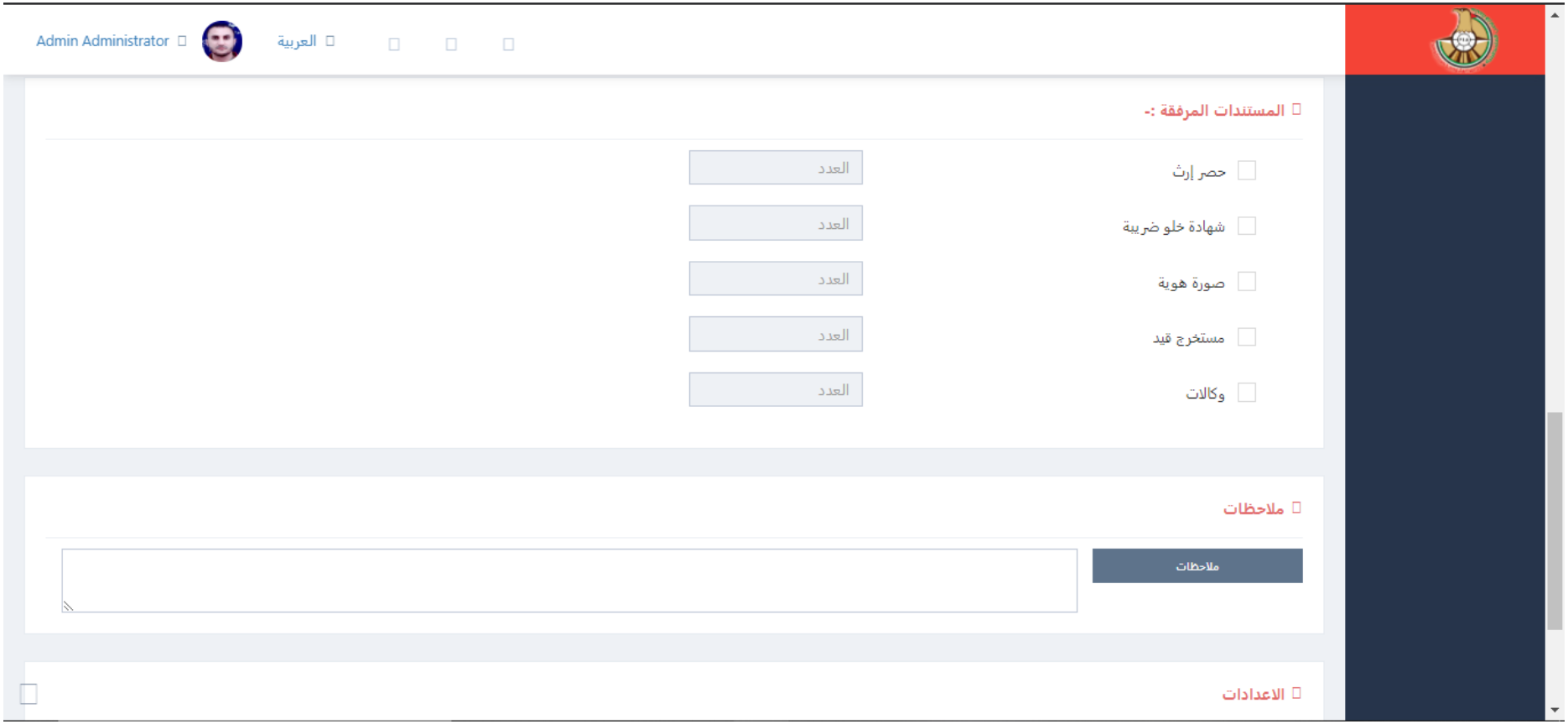
## Appendix A) UML For PLA Integrity Smart Contract

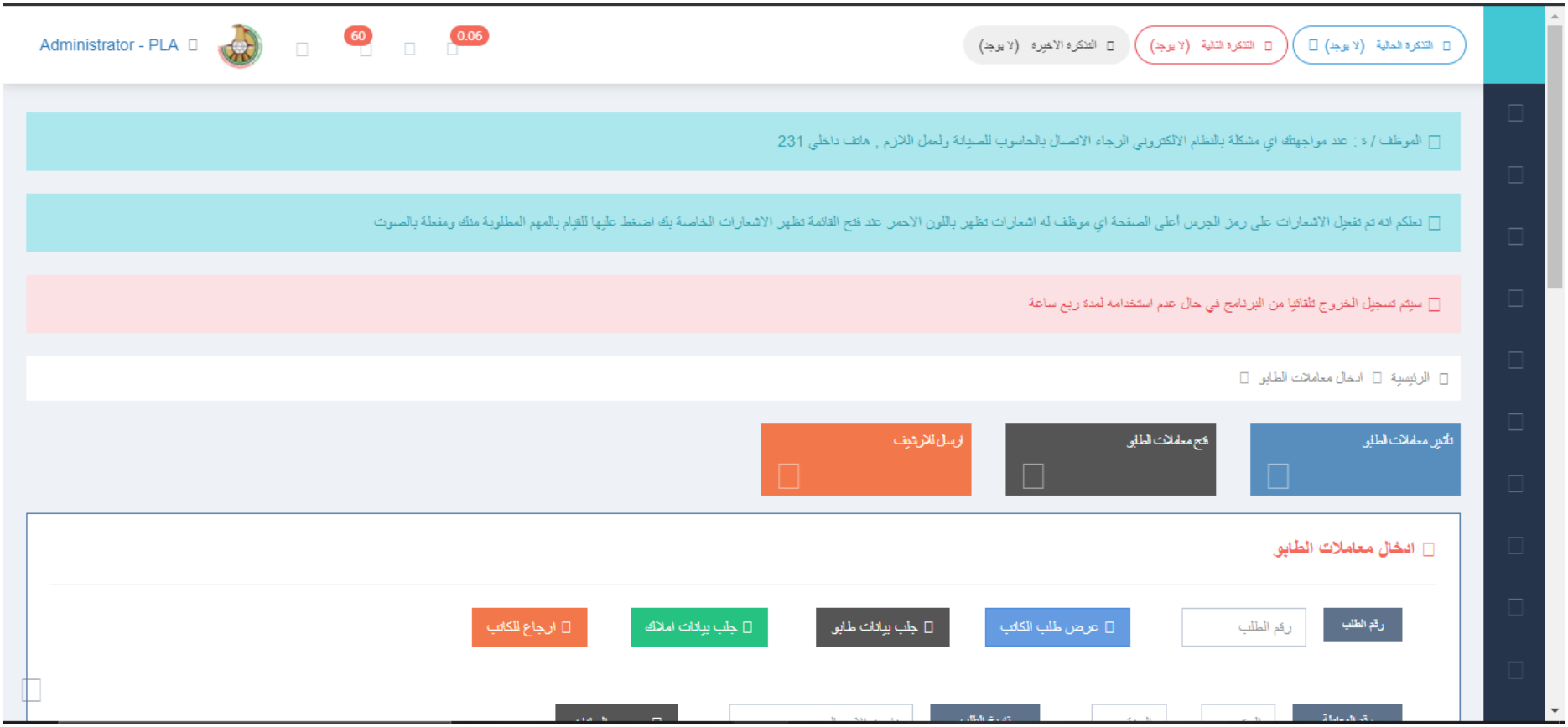
## Appendix B) PLA Interface


**Figure 8-1:Figure 4 8:UML For PLA Integrity Smart Contract**


Admin Administrator
العربية
للاستفسار يمكنكم التواصل مع الدعم الفني من خلال المحادثة في اسفل الصفحة
مرحبا بكم في الخدمات الالكترونية في سلطة الاراضي بحلتها الجديدة
الرئيسية
اضافة عقد بيع
عقد بيع
اسم المدينة او القرية
ابو مدين
رقم خلو الطرف الضريبي
رقم خلو الطرف الضريبي
المبلغ المدفوع
المبلغ المدفوع
دولار امريكي
بيع بشرط الاعاشة
راتب الاعاشة
فك اعاشة
رقم جوال للمتابعة
رقم الجوال
إدارة المستخدمين
الفوائم
نوع المستخدم
اشعارات
التخزين السحابي
المواضيع والمشاركات
الاقسام
التصنيفات


Admin Administrator
العربية
مخصص
خدمات المواطنين
الاستعلامات
الطلبات
المعاملات
اللجان
التنبيهات
قائمة الاسماء
يشهد هذا العقد أن :-
البائع *
هوية البائع
الاسم
الاب
الجد
العائلة
أخرى
اضافة بائع
قبل أن ينقل إلى :-
المشتري *
هوية المشتري
الاسم
الاب
الجد
العائلة
أخرى
توحيد حصص
اضافة مشتري
نص العقد :-
الملك المبين في الجدول الآتي وذلك لقاء مبلغ ---------دينار اردني المدفوع من قبل المشترين والذي يعترف البائعين بموجب هذا العقد بقبضه تم أن البائعيين يعلن بهذا أن صاحب الملك المذكور بالجدول أدناه وأنه بخلاف ما هو مذكور في الجدول فأن الملك خال من جميع حقوق الغير وأن البائعيين والمشتريين يعلنان بهذا العقد أن المبلغ المشار إلى سداده بهذا العقد هو عبارة عن كل المبلغ المدفوع لقاء البيع. .
القطعة
القسيمة
المساحة
ملاحظات
اضافة قطعة وقسيمة


Admin Administrator
العربية
المستندات المرفقة :-
حصر إرث
العدد
شهادة خلو ضريبة
العدد
صورة هوية
العدد
مستخرج قيد
العدد
وكالات
العدد
ملاحظات
ملاحظات
الاعدادات


Administrator - PLA
60
0.06
التذكرة الحالية (لا يوجد)
التذكرة التالية (لا يوجد)
التذكرة الاخيرة (لا يوجد)
الموظف / ة : عند مواجهتك اي مشكلة بالنظام الالكتروني الرجاء الاتصال بالحاسوب للصيانة ولعمل اللازم , هاتف داخلي 231
تعلكم انه تم تفعيل الاشعارات على رمز الجرس أعلى الصفحة اي موظف له اشعارات تظهر باللون الاحمر عند فتح القائمة تظهر الاشعارات الخاصة بك اضغط عليها للقيام بالمهم المطلوبة منك ومفعلة بالصوت
سيتم تسجيل الخروج تلقائيا من البرنامج في حال عدم استخدامه لمدة ربع ساعة
الرئيسية
ادخال معاملات الطابو
تأشير معاملات الطابو
فتح معاملات الطابو
ارسال الارشيف
ادخال معاملات الطابو
رقم الطلب
رقم الطلب
عرض طلب الكاتب
جلب بيانات طابو
جلب بيانات املاك
ارجاع للكاتب