

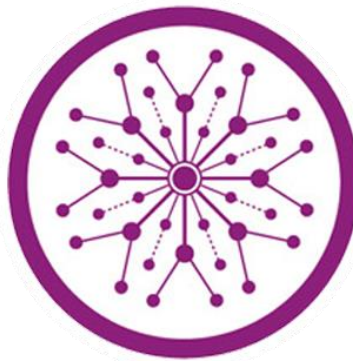
# Blockchain-based Land Record System

Final Year Project

Session 2019-2023

A project submitted in partial fulfillment of the degree of

BS in Computer Science



Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2023

Type (Nature of project)	<input checked="" type="checkbox"/> Development <input type="checkbox"/> Research <input type="checkbox"/> R&D			
Area of specialization				
FYP ID	FYP-BCSM-F22-010			
<b>Project Group Members</b>				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	Bcsm-f19-188	Ahmad Raza	bcsm-f19-188@superior.edu.pk	
(ii)	Bcsm-f19-206	Muhammad Hamees	bcsm-f19-206@superior.edu.pk	
(iii)	Bcsm-f19-207	Muhammad Usama	bcsm-f19-207@superior.edu.pk	

\*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

### Plagiarism Free Certificate

This is to certify that, I M. Usama S/D of Muhammad Saleem, group leader of FYP under registration no. Bcsm-f19-207 at Computer Science Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: \_\_\_\_\_ Name of Group Leader: Muhammad Usama Signature: \_\_\_\_\_

Name of Supervisor: Mam Hafiza Maria

Designation: Lecturer

Signature: \_\_\_\_\_

HoD: Dr. Irfan-ul-Din

Signature: \_\_\_\_\_

# Blockchain-based Land Record System

## Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Ahmad Raza, M. Hamees, M. Usama	1.0			

## APPROVAL

---

### PROJECT SUPERVISOR

Comments: \_\_\_\_\_

---

Name: \_\_\_\_\_

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

---

### PROJECT MANAGER

Comments: \_\_\_\_\_

---

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

### HEAD OF THE DEPARTMENT

Comments: \_\_\_\_\_

---

Date: \_\_\_\_\_ Signature: \_\_\_\_\_

## Dedication

*I offer my heartfelt thanks to Almighty Allah for His blessings, and I am indebted to my Parents and Supervisor, Mam Hafiza Maria, for her unwavering support, guidance, and inspiration. This project would not have been possible without her invaluable contributions, and I am truly grateful for the opportunity to work alongside such an exceptional mentor.*

## **Acknowledgments**

I am truly humbled and grateful for the blessings and guidance bestowed upon us by Almighty Allah. It is through His grace that we were granted the qualities of determination, knowledge, and courage, enabling us to successfully complete this important endeavor.

I would like to express my deepest appreciation to my supervisor, Mam Hafiza Maria, for her invaluable support and unwavering commitment throughout the entire process of developing this project. Her guidance and advice have been instrumental in shaping our ideas, refining our methodology, and ensuring the project's overall success.

Mam Hafiza Maria's dedication to our growth and learning has been evident in every interaction and milestone achieved. Her expertise, patience, and unwavering belief in our abilities have not only inspired us but also instilled within us a sense of confidence and motivation.

## Executive Summary

In the current era, the buying and selling of lands and transfer of lands are active daily processes in any city. Also, there is a lot of land conflict rise using the current system. The current system is on paper or on a digital system which creates conflict. Blockchain technology will overcome these kinds of problems. For the Land Record use case blockchain is best to build the system on the blockchain, it provides the immutability that no one can tamper with the data of land records. Decentralize system without anyone involvement of the third party, the citizen can buy and sell the property. The record is secure in the form of a hash and all data is stored in blocks and each block has different transactions that provide the facility of traverse to check the previous owners of the land. The Blockchain-based Records System will be a DApp (Decentralized Application) that government has functionalities that can register the property of the owner, view the land record history, buy, and sell property, and verify the transfer of land transactions. On the other side, a citizen can buy and sell property, mark the property for sale, and request for sale, and after successful transfer, tax is automatically deducted from both sides and it transfers to the government with detail. All the details of each transaction will be stored on a blockchain network that is secure, immutable, and transparent. By using blockchain for land record use cases the problem in the current system of land records will be solved, also save time cost and reduce expenses.

## Table of Contents

Blockchain-based Land Record System .....	i
Plagiarism Free Certificate .....	ii
Blockchain-based Land Record System .....	iii
Dedication .....	v
Acknowledgments .....	vi
Executive Summary .....	vii
Table of Contents .....	viii
List of Figures .....	x
List of Tables .....	xi
Chapter 1 .....	1
Introduction .....	1
1.1. Background .....	2
1.2. Motivations and Challenges .....	2
1.3. Goals and Objectives .....	2
1.4. Literature Review/Existing Solutions .....	3
1.5. Gap Analysis .....	3
1.6. Proposed Solution .....	4
1.7. Project Plan .....	4
1.7.1. Work Breakdown Structure .....	5
1.7.2. Roles & Responsibility Matrix .....	5
1.7.3. Gantt Chart .....	6
1.8. Empathy Map .....	6
Chapter 2 .....	7
Software Requirement Specifications .....	7
2.1. Introduction .....	8
2.1.1. Purpose .....	8
2.1.2. Document Conventions .....	8
2.1.3. Intended Audience and Reading Suggestions .....	8
2.1.4. Product Scope .....	9
2.1.5. References .....	9
2.2. Overall Description .....	9
2.2.1. Product Perspective .....	9
2.2.2. User Classes and Characteristics .....	9
2.2.3. Operating Environment .....	10
2.2.4. Design and Implementation Constraints .....	10
2.2.5. Assumptions and Dependencies .....	10
2.3. External Interface Requirements .....	10
6.1.1. User Interfaces .....	11
6.1.2. Hardware Interfaces .....	11
6.1.3. Software Interfaces .....	11
6.1.4. Communications Interfaces .....	12
6.2. System Features .....	12
6.2.1. System Feature 1 .....	12
6.2.1.1. Functional Requirements .....	12

6.3. Other Nonfunctional Requirements .....	15
6.3.1. Performance Requirements.....	15
6.3.2. Safety Requirements.....	15
6.3.3. Security Requirements.....	15
6.3.4. Software Quality Attributes.....	15
6.3.5. Business Rules.....	15
6.4. Portability Requirements.....	16
6.5. Domain Requirements.....	16
Chapter 3.....	17
Use Case Analysis.....	17
3.1. Use Case Model .....	18
3.2. Use Case Descriptions.....	18
Chapter 4.....	20
System Design .....	20
4.1. Architecture Diagram.....	21
4.2. Entity Relationship Diagram with data dictionary.....	23
4.1. Class Diagram .....	24
4.2. Sequence / Collaboration Diagram .....	25
4.1. Operation contracts .....	25
4.2. Activity Diagram.....	26
4.3. State Transition Diagram .....	27
4.4. Data Flow diagram .....	28
Chapter 5.....	30
Implementation .....	30
5.1. Important Flow Control/Pseudo codes.....	31
5.2. Deployment Environment .....	32
5.3. Tools and Techniques.....	32
5.4. Best Practices / Coding Standards.....	33
Chapter 6.....	34
Testing and Evaluation .....	34
6.1. Use Case Testing.....	35
6.2. Equivalence partitioning .....	37
6.3. Unit testing.....	38
6.4. Integration testing.....	40
6.5. Performance testing.....	41
6.6. Stress Testing .....	41
Chapter 7.....	42
Summary, Conclusion and Future Enhancements .....	42
7.1. Project Summary .....	43
7.2. Achievements and Improvements .....	43
7.3. Critical Review.....	44
7.4. Lessons Learnt.....	45
7.5. Future Enhancements/Recommendations .....	46

## List of Figures

1.1	Caption of first figure of first chapter	6
1.2	Caption of second figure of first chapter	7
2.1	Caption of first figure of second chapter	14
2.2	Caption of second figure of second chapter	22
2.3	Caption of third figure of second chapter	26
5.1	Caption of first figure of fifth chapter	49
5.2	Caption of second figure of fifth chapter	49

## List of Tables

1.1	label of first table of first chapter	6
1.2	label of second table of first chapter	7
2.1	label of first table of second chapter	14
2.2	label of second table of second chapter	22
2.3	label of third table of second chapter	26
5.1	label of first table of fifth chapter	49
5.2	label of second table of fifth chapter	49

# Chapter 1

## **Introduction**

# Chapter 1: Introduction

*In Land Records System a landowner can sell and request to buy properties, that working on the blockchain (Ethereum) and smart contracts using Dapp. The land inspector is the person who has access to the system to manage the properties, view requests, verify the transaction, and initiates the transfer and document validation and authenticity. It will reduce the time cost, payment cost, provide transparency, and keep the records secure. Once the transaction is done all the data record in the blockchain, and the hash is generated automatically, and it cannot change that is best for security purposes.*

## 1.1. Background

*Blockchain is an emerging technology nowadays and system developed on the blockchain a great impact to change the world industries working way. One famous example is cryptocurrency in the form of Bitcoin BTC that takes over paper currency. Blockchain collects information together in blocks and each block has different transactions. To utilize the blockchain to develop a system for the land records system that provides the facilities decentralize, secure, transparent, not changeable data and traceability. That is to empower the land record system.*

## 1.2. Motivations and Challenges

The best solution for the existing system problem is offered by the block chain-based Land Records System. It will be used by government agencies, property sellers, and buyers who want to purchase property. Because of block chain technology, this system will eventually offer the traceability function that verifies the prior and current landowner.

## 1.3. Goals and Objectives

The product is supposed to be open source, under the GNU General Public License. The product is DApp (Web) based system implementing a peer-to-peer model. The Blockchain based Land Records System provides a strong mechanism for citizens to buy and sell land, government can register new land and verify the transfer of land.

- The following are the main features that are included in Blockchain based Land Records System
- Manage Land Record: Government can manage their properties; Citizen can view the land record details.
- User Authentication: User access the DApp using login credentials, and newly user must have to register before login.
- Tax Wallet: On every successful transfer of land, the seller and buyer must pay tax and then tax transfer to Government Tax Wallet.
- Optimized Search: search is an optimized local search engine based on keywords.
- Complain Forum: Provides the user with a platform to complain about the issue to resolve.

#### **1.4. Literature Review/Existing Solutions**

*The current process of registering land records involves property verification, value estimation, stamp paper creation, stamp duty payment, and final transfer/registration without block chain technique. Although all the land register databases are going too connected to the land registry system, but these are under the threat of malware or hacker and cracker attacks.*

*This current system has so many disadvantages and has vulnerabilities such as, insecure. There are login-based authentications for every person in existing land-based record systems but these are not very much secure due to varieties of vulnerabilities.*

#### **1.5. Gap Analysis**

Even if the development of this project will help many people who are selling and buying block chain-based plots, the existing method has numerous drawbacks and security flaws. In the current land-based record systems, there are login-based authentications for everyone, but these are not very safe due to a number of flaws (secure).

We receive encouraging feedback when we discuss this idea with those in our immediate vicinity, leading us to believe that this project will probably succeed

## 1.6. Proposed Solution

*The successful solution for our proposal is to provide:*

- *A website for land sale-purchase system.*
- *Proper blockchain based land registry website. It will eliminate the vulnerabilities of the previous land registry systems by providing a secure method.*
- *To provide a website to have all the details about the land, including their history and previous owner's details.*

## 1.7. Project Plan

*Blockchain technology affecting the internet world with its powerful features and best apply to Land Record use case. This system will be used by the government and citizens. Users interact with this system by using DApp (Decentralized Application). This system will be available only in a specific country. A citizen can sell and buy their property without any involvement of a third party, it reduces the time cost, payment cost, and no one can tamper with the records of the user. When selling and buying of property is the process doing on Blockchain, DApp automatically deduces tax and transfer to the government. Government can add property and verify the records of the citizen. Smart contracts are contracts that can be completed without any human interaction. Every account has a public and private key. All the transactions stored on Blockchain in blocks. Each block containing the current hash and previous block hash that create the chain. Therefore, Government or citizen can check the previous owner of the land also it provides security to records if anyone tries to change the hash it cannot possible because when chain break it reject the transaction. Government can check the history of land records. The system also provides the citizen to calculate the inheritance by using the inheritance calculation system. In the centralized system, there are many drawbacks such as admin of system can be able to make illegal changes, no transparency, insecure. As all the land register databases are not connected to the land registry system, it reduces the efficiency and equality of the system.*

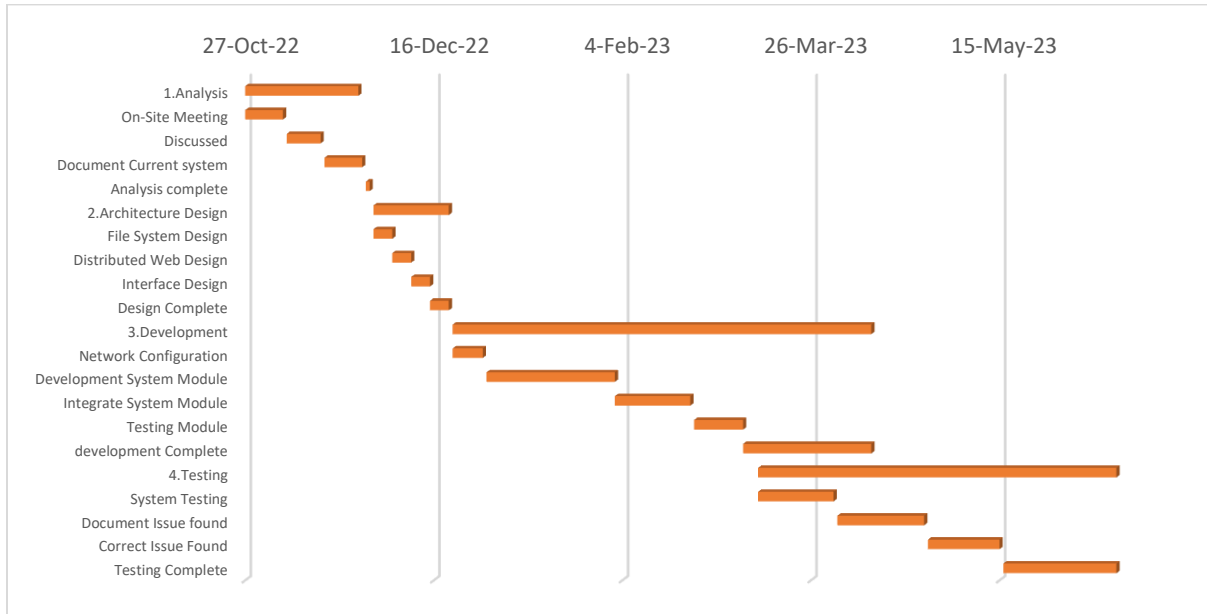
### 1.7.1. Work Breakdown Structure

- Requirement Gathering
- Designing
- Implementation
- Testing
- Conflict Resolving
- Deployment
- Marketing

### 1.7.2. Roles & Responsibility Matrix

WBS #	WBS Deliverable	Activity	Duration (Days)
<b>1.1</b>	<b>Analysis</b>	None	30
	On-Site Meeting	Analysis	10
	Discussed	On-Site Meeting	9
	Document Current system	Discussed	10
	Analysis complete	Document Current system	1
	<b>Architecture Design</b>	Analysis, Analysis complete	20
	File System Design	Architecture Design	5
	Distributed Web Design	File System Design	5
	Interface Design	Distributed Web Design	5
	Design Complete	Interface Design	5
	<b>Development</b>	Architecture Design, Design Complete	111
	Network Configuration	Development	8
	Development System Module	Network Configuration	34
	Integrate System Module	Development System Module	20
	Testing Module	Integrate System Module	13
	development Complete	Testing Module	34
	<b>Testing</b>	Development, development Complete	95
	System Testing	Testing	20
	Document Issue found	System Testing	23
	Correct Issue Found	Document Issue found	19
	Testing Complete	Correct Issue Found	30

### 1.7.3. Gantt Chart



### 1.8. Empathy Map

Says	Thinks
<ul style="list-style-type: none"> <li>• Need online land buying process</li> <li>• Trustworthy</li> </ul>	<ul style="list-style-type: none"> <li>• Need online blockchain System to get rid of stamping.</li> <li>• User friendly</li> </ul>
Does	Feels
<ul style="list-style-type: none"> <li>• Talking to different market person</li> <li>• Explore social media</li> </ul>	<ul style="list-style-type: none"> <li>• Worried</li> <li>• Confused</li> </ul>

# Chapter 2

## **Software Requirement Specifications**

## Chapter 2: Software Requirement Specifications

### 2.1. Introduction

#### 2.1.1. Purpose

As all the land register databases are not connected to the land registry system, it reduces the efficiency and equality of the system[1]. To provide automation, Ethereum Blockchain has smart contracts which are predefined rules in form of coding which tell that how, when, and where certain process happened. And Smart contracts are contracts that can be completed without any human interaction[2]. The main purpose of the Blockchain based Land Records System is to provide immutability of records which can be achieved through a decentralized system, where each node has the same authority no one can tamper with the record. It is more secure than the current system because each transaction packed in block is connected to the previous block with the encrypted hash value. Being public blockchain the system provides transparency; anyone can view the land details. The citizens can buy and sell land using our developed DApp. The government can verify the transfer of land, register new land with owner detail.

#### 2.1.2. Document Conventions

- Blockchain based Land Record System - BLRS
- Decentralized Application - DApp
- Smart Contracts - SC
- Data Flow Diagram - DFD
- Distributed Ledger -DL
- Interplanetary File System - IPFS
- Entity Relationship Diagram - ERD
- Graphic User Interface - GUI
- Procedural description language - PDL
- State Transition Diagram - STD
- Unified Modeling Language – UML

#### 2.1.3. Intended Audience and Reading Suggestions

This document is suggested for tech students, researchers, project managers, programmers, designers, developers, testers, documentation writers, & all type of educational institutional

person. This document consists of the various steps and procedures for the application update initiated with an open discussion with the Project manager. Following section describes the rest of the product function. Scope and other overall description. Finally, with the reference.

#### **2.1.4. Product Scope**

The buying and selling of lands have a proper system in all developed countries and taking it to an advanced level that is secure and immutable. The Blockchain based Land Records System provides the best solution for the current system problem. It will be used by government bodies, sellers who want to sell the property, and the buyer who want to buy the property. Also, it automates the tax to the government on buying and selling land by the citizen. In the Future this system provides the traceability feature that checks the previous and current landowner because of blockchain. Every block connects with the previous block. It will resolve the land conflicts and provide security that is not hackable, and minimize the scope of land disputes, records are transparent that checked by the public and private keys. Records data in digital ledger for transactions or contracts. The use of blockchain technology serves as the basis for a more reliable, cheaper, and more efficient land registry.

#### **2.1.5. References**

### **2.2. Overall Description**

#### **2.2.1. Product Perspective**

The product is supposed to be open source, under the GNU General Public License. The product is DApp (Web) based system implementing a peer-to-peer model. The Blockchain based Land Records System provides a strong mechanism for citizens to buy and sell land, government can register new land and verify the transfer of lan

#### **2.2.2. User Classes and Characteristics**

There are three kinds of users in our Land Record System. First one is Seller. Second one is buyer and third one is govt from where the land would be registered and all records would be kept in records.

### 2.2.3. Operating Environment

This DApp will work on every operating system Windows (7,8,10), Linux, and Mac OS, and can be supported by any type and version of browsers installed in your operating system and browser must have Meta Mask extension added. DApp also works on Android OS and IOS must have a browser with a Meta Mask extension. The operating system minimum has 4 GB RAM and needs good processing power to run the DApp for better results.

### 2.2.4. Design and Implementation Constraints

This system is provisioned to be built on the React framework which is highly flexible. For data storage, it uses Blockchain to store the data, that use personal blockchain Ganache and Test Network and for document storage that will use IPFS. The challenges in developing the product include deploying the DApp on Ethereum Mainnet. The DApp will be running 24 hours a day.

### 2.2.5. Assumptions and Dependencies

- User's Block chain addressed compromised can result in his data confidentiality and access to his profile.
- User's lack of knowledge as Block chain is a new technology and only little number of users will know about it in earlier stage.
- System security risks where unauthorized person may gain access to the system where they may violate data confidentiality, integrity and availability.
- The system may have nonfunctional requirements risk failure such as lack of multithreading functionality, lack of survivability functionality which may cause the system to crash if accessed by many users concurrently.

## 2.3. External Interface Requirements

### 3. Login/Sign up Interface:

Here the interface will be provided to the users to register them with the application. In case the user is already registered with the application a login page will be provided.

### 4. Home:

This will be the welcome interface of the application for all the users. There will be some navigation that will make user capable of navigate through all the application easily.

**5. Register land:**

Here the citizen will enter the details of property he wants to register such as city, property no, owner Block chain address.

**6. Search property:**

Here the citizen will be able to search the properties by property no and he can request the owner of that property for purchase if he likes that property.

**6.1.1. User Interfaces**

The following are the main UI features that are included in Blockchain based Land Records System

- Manage Land Record: Government can manage their properties; Citizen can view the land record details.
- User Authentication: User access the DApp using login credentials, and newly user must have to register before login.
- Tax Wallet: On every successful transfer of land, the seller and buyer must pay tax and then tax transfer to Government Tax Wallet.
- Optimized Search: search is an optimized local search engine based on keywords.
- Complain Forum: Provides the user with a platform to complain about the issue to resolve.

**6.1.2. Hardware Interfaces**

Since there's no direct interaction of this blockchain project with hardware because it's a web-based application and could be accessed only through mobile, tablets and desktops. So, it has no hardware interfaces

**6.1.3. Software Interfaces**

The software provides good graphical interface for all the different types of users like government admin, citizen whither they are seller or buyer so that they can easily operate the system for performing the required task such as login to their profiles, viewing the record of property records, buying/selling the property, taxation, payments etc.

#### **6.1.4. Communications Interfaces**

- The communication between the different parts of the system is important since they depend
- on each other. However, in what way the communication is achieved is not important for the
- system and is therefore handled by the underlying operating systems for Land Record System.

### **6.2. System Features**

- Land list
- Maintain Profile
- Home
- Register land
- Inheritance calculator
- Explore
- My account
- Land record history
- Manage properties
- Sign out

#### **6.2.1. System Feature 1**

##### **6.2.1.1. Functional Requirements**

The registration will have the following modules

REQ-SF2-1: Registration Module

REQ-SF2-2: Selling a Land-Plot

REQ-SF2-3: Buying a Land-Plot

REQ-SF2-4: Polyline API

REQ-SF2-5: Viewing Land Records History

REQ-SF2-7: Government Side

All the modules of Blockchain based DApp.

### **REGISTRATION MODULE**

- The user will log in using his account.
- In this system, registration is applied to all users.
- The government also acts as the user to buy and sell the property.
- Each user creates an account and gets a unique private key for transactions.

### **SELLING A LAND-PLOT MODULE**

- A user can sell the property and have owned multiple registered properties.
- The user has the option to mark the property as sellable.
- And user can display all properties, want to sell.
- Also, the user can set the price of a property.

### **BUYING A LAND-PLOT MODULE**

- Users can buy the properties, all the sellable properties searchable to the area the buyer wants to buy from.
- The buyer can view its price, current owner of the plot, and coordinates of the plot.
- The buyer sends the request to the seller by using the Dapp (Distributed App).
- The seller can view the request and able to accept and reject the request of the buyer.
- If the seller accepts then the piece of code called smart contract self-execute and checks the condition, then approval send to the buyer with a money transfer request.
- The buyer confirms and the land title is transferred to the buyer.
- The tax which is paid to the government can be directly transferred from the buyer's account to the government using the smart contract itself.

### **POLYLINE API MODULE**

- The plot or the land property can be mapped precisely using Polyline API from Google.
- This API gives the exact coordinates of the plot, thus calculating the area of the plot the user intends to sell.
- The location of the plot can also be easily determined anywhere on the globe, tracing the plot's coordinates.

- This provides to be a very strong verification feature and aims to solve issues related to the property area.

#### VIEW LAND RECORD HISTORY MODULE

- Land Records history option for the government organizations who want to check details about the owner of land property.
- They can view the history of the owners of the land by entering the ID of that property.
- Also, the officials can check the list of properties currently owned by a particular user by entering the address key corresponding to the account of that user.

#### GOVERNMENT SIDE MODULE

This module will be used by Government only.

<b>Identifier</b>	Government Side
<b>Title</b>	Manage Land Records
<b>Requirement</b>	Government must verify the transfer of land and ensure that no anomalies in the land record. Also, register new land with owner detail.
<b>Source</b>	From both citizen buyer and citizen seller
<b>Rationale</b>	To verify the transfer of land to avoid any fake selling and buying.
<b>Business Rule (if required)</b>	Citizen buyers must have the above money to requesting land value. Successful transfer of land deduce tax from both parties and transfer to government tax wallet.
<b>Dependencies</b>	Both citizen seller and buyer
<b>Priority</b>	Medium

## **6.3. Other Nonfunctional Requirements**

### **6.3.1. Performance Requirements**

The performance of the system is very good. It performs all the tasks very correctly and provides all the results quickly and accurately. When the contract between the seller and buyer the RPC on a peer-to-peer network that needs processing power to run the contract. The system provides a good response time to the user.

### **6.3.2. Safety Requirements**

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

### **6.3.3. Security Requirements**

There's no safety issue of DApp just because of high security. No one can change the data of others that's why it is safe and secure

### **6.3.4. Software Quality Attributes**

All the data inside the system and the system itself will be protected against malware attacks or unauthorized access. Each user has their login credentials to access the system. Datastore in the blockchain that is impossible to hack because of data store in the form of transaction in a block, each block connected with the previous block with the encrypted hash value. If anyone tries to change the data, then other nodes validate it if data change then the transaction failed automatically by the system.

### **6.3.5. Business Rules**

This blockchain app is easy to use. The UI of is user-friendly and highly graphical.

## 6.4. Portability Requirements

The DApp should run on every machine, Windows, Mac OS, Linux, Mobile Phones. and can be supported by any type and version of browsers installed in your operating system and the browser must have Meta Mask extension added. DApp also works on Android OS and IOS must have a browser with a Meta Mask extension. The operating system minimum has 4 GB RAM and needs good processing power to run the DApp for better results.

## 6.5. Domain Requirements

The functionality & operation performed by our application would be in Domain Requirements:

- System must have a new Registration and authentication component
- System must have a portal/page to login/register different types of users.
- System should also be able to behave like an e-commerce platform so that designers will be able to publish their articles to sell them to buyers.

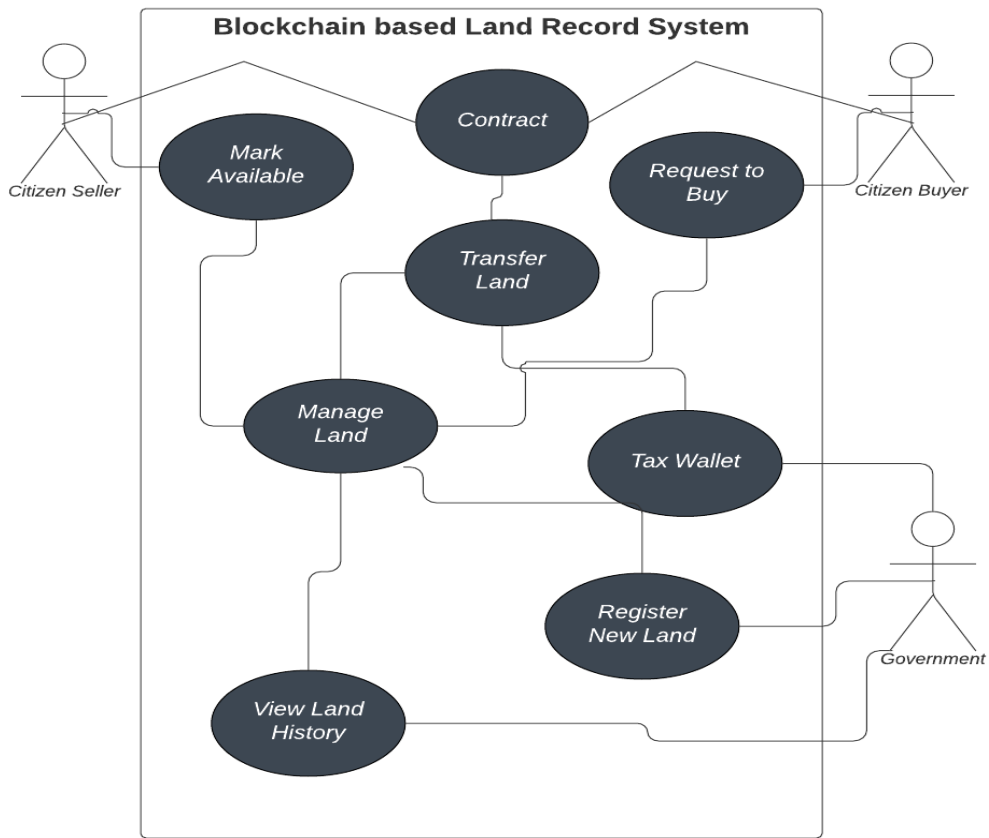
# Chapter 3

## Use Case Analysis

## Chapter 3: System Analysis

In this use case analysis, there's a graphical representation of data and functionalities of users i.e., how they are linked with each other logically

### 3.1. Use Case Model



### 3.2. Use Case Descriptions

The table below indicates a comprehensive use case template filled in with an example drawn from the BLOCHAIN BASED LAND RECORDS SYSTEM.

<b>Use Case ID:</b>	BLRS-1
<b>Use Case Name:</b>	BLOCKCHAIN BASED LAND RECORDS SYSTEM
<b>Actors:</b>	Primary Actors: Citizen Buyer, Citizen Seller. Secondary Actor: Government

<b>Description:</b>	<p>A Citizen can access the Blockchain based Land Record System by login credentials, and act as buyer and seller. Citizens can mark owned property for sale, request to buy, accept or reject the buyer request.</p> <p>Smart Contract that makes a contract between the buyer and seller, transfer initiate, and tax automatically deduces and update to Tax Wallet.</p> <p>Government can access DApp by their login credentials. And able to register new land, view land record history, manage the tax wallet.</p>
<b>Trigger:</b>	<p>A Citizen seller marks the owned property for sale and waits for the request from buyers. After getting the request, can accept or reject the request. If the seller accepts then the land transfer initiate, money transfer initiate, and tax automatically deducted.</p> <p>A Citizen buyer search for property to buy and send the request to buy. If the seller accepts then land ownership change and value of land deduced and transfer to the seller account.</p> <p>Government check and verify the transfer of land and view the tax wallet.</p>
<b>Preconditions:</b>	<p>PRE-1. All user Buyer, Seller, and Government are logged into Blockchain based Land Record System to use it.</p> <p>PRE-2. Both Buyer and Seller must be registered for proper usage of DApp and transfer of land into accounts.</p> <p>PRE-3. Government must verify the transfer of land and register new land with owner details.</p>
<b>Postconditions:</b>	<p>POST-1. For mark property to the sale must have ownership of the seller.</p> <p>POST-2. For sending buying requests to the seller, the buyer must have the above money to land value.</p> <p>POST-3. Government must verify the land before the transfer of land.</p>
<b>Normal Flow:</b>	<p>Citizen has to login for profile and view requests, view land.</p> <p>Citizen buyers send the request to buy land, and view requested, and Citizen sellers check the requests and accept or reject the request.</p> <p>The government views the land record history and Tax Wallet.</p>
<b>Alternative Flows:</b>	<p>Government can register the new land with owner details.</p> <p>Any user can calculate the inheritance by using an inheritance calculator.</p> <p>Citizen Seller can verify the buyer details and Buyer can verify the landowner details.</p>
<b>Exceptions:</b>	<p>DApp provides the facility to verify the land deed, contract between seller and buyer, automatically deduce tax.</p>
<b>Business Rules:</b>	<p>BR-1 Value of land varies on the in which region land belongs to.</p> <p>BR-2 Transfer of land done if the buyer has money above the value of the land.</p>
<b>Assumptions:</b>	<p>Assume that one hundred transfer land on each day. The system makes the record on daily basis.</p>

# Chapter 4

## System Design

## Chapter 4: System Design

### Incremental Model

The waterfall model has many drawbacks in the development of any product, to overcome these drawbacks incremental model is used in the development phase of our project. In this model, the product is developed in increments and module-wise one after one, each module contains more functionalities than before. These smaller pieces are then built and delivered to the client in increments. Quick response from clients. Each module is smaller than compared to the whole module. This model is used in our project.

### 4.1. Architecture Diagram

The structure of the system explains its core components, their relationships, and how they deal with each other. Blockchain Software architecture and design include several factors such as business logic, Ethereum Blockchain, DApp, and design of the system. In Architecture, nonfunctional decisions are cast and separated by the functional requirements. In Design, functional requirements are accomplished. The government module is very important. It is a major module in this module government have all rights to manage the property, verify property, land record history, and view tax generated. Anyone can visit, anyone can visit the main page related to the system and can get information and knowledge about our system that how to use to the system and how anyone can get their self-registered. Users can also calculate the inheritance by using the inheritance calculator on the system In the Citizen module, users can sell and buy property, also check the buyer request, and accept or reject the request. A combination of the modules makes up the system. We can use flowcharts to represent and illustrate the architecture.

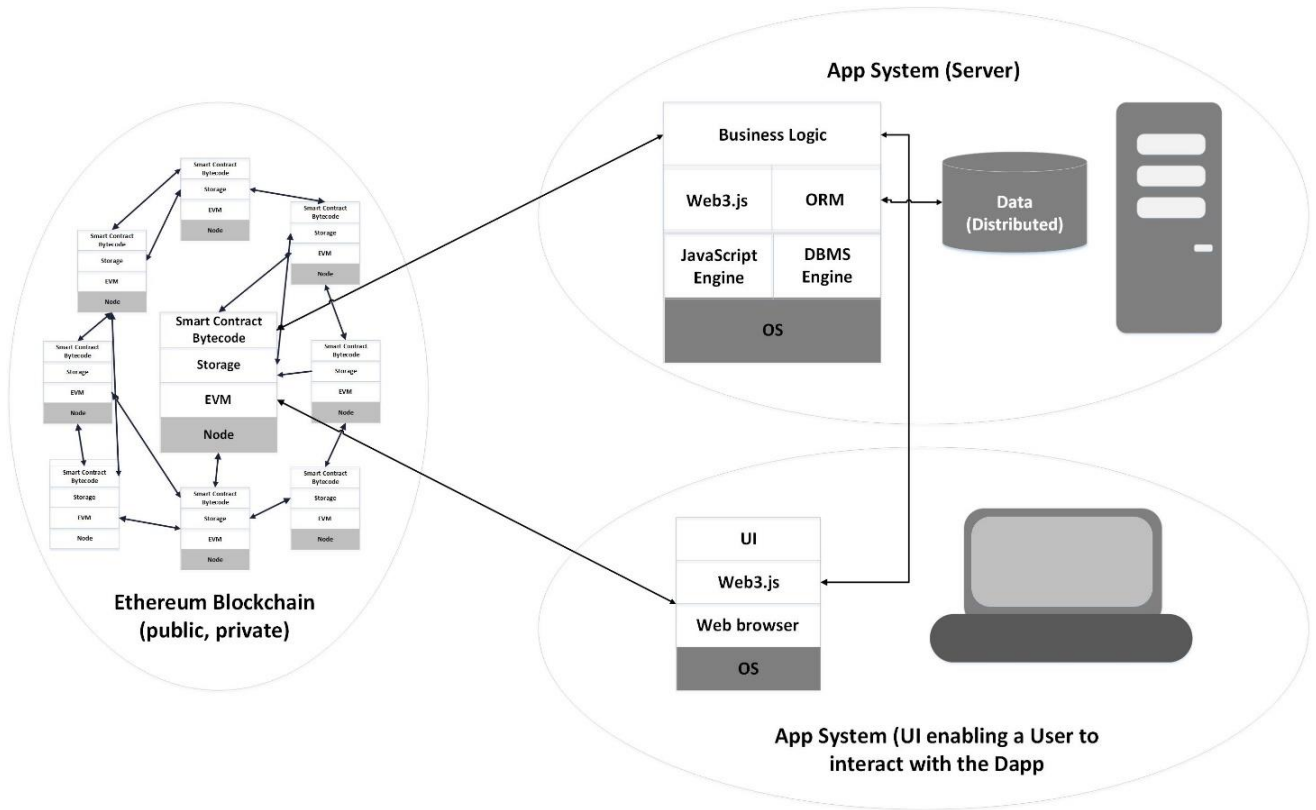


Figure 1: Architecture Design of DApp using Ethereum Blockchain

The **server** stores data that cannot be stored in the blockchain and performs business computations.

## Architecture Design System

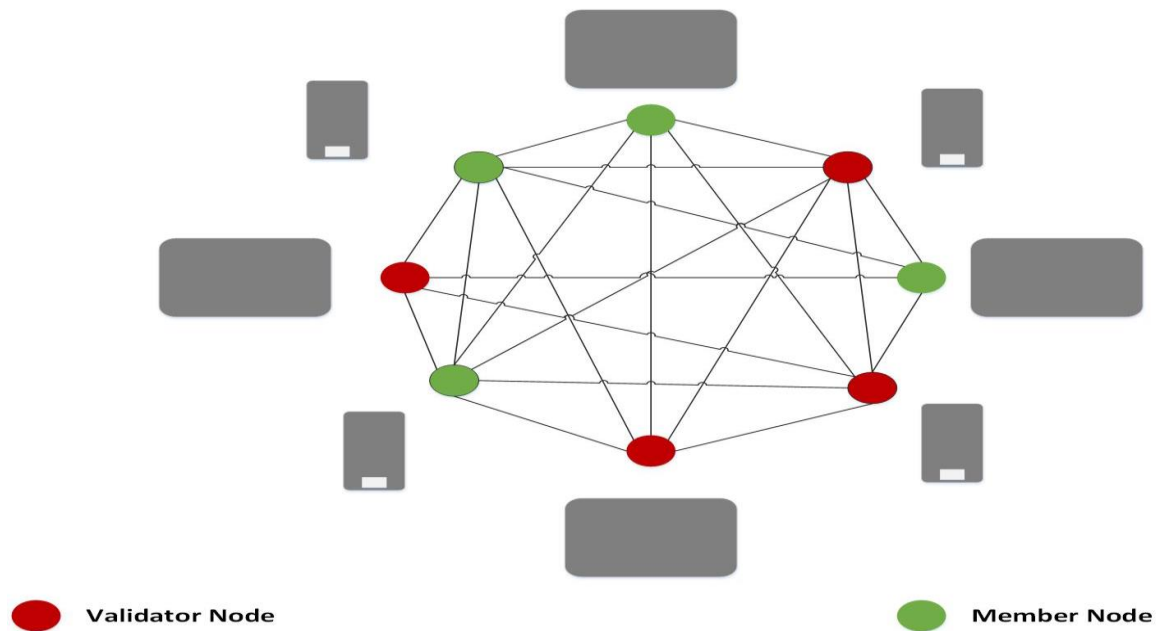


Figure 2: Architecture Design System

### 4.2. Entity Relationship Diagram with data dictionary

4. Blockchain based Land Record System - BLRS
5. Decentralized Application - DApp
6. Smart Contracts - SC
7. Data Flow Diagram - DFD
8. Distributed Ledger -DL
9. interplanetary File System - IPFS
10. Entity Relationship Diagram - ERD
11. Graphic User Interface - GUI
12. Procedural description language - PDL
13. State Transition Diagram - STD
14. Unified Modeling Language – UML

### 4.1. Class Diagram

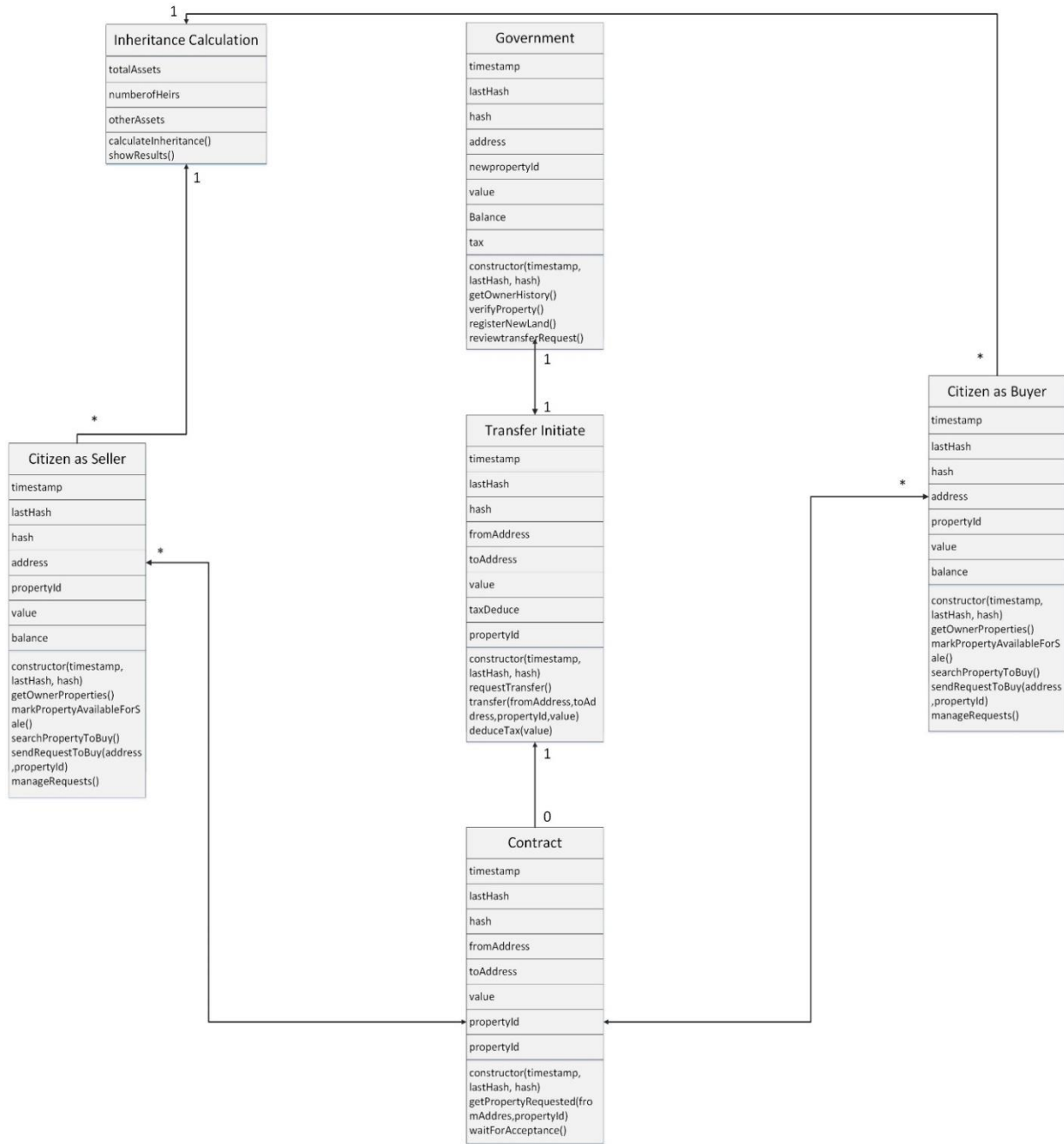


Figure: Class Diagram

## 4.2. Sequence / Collaboration Diagram

15. This sequence diagram shows that how the DApp will work, tasks after task sequence-wise. It gives a complete description of the working of our DApp.

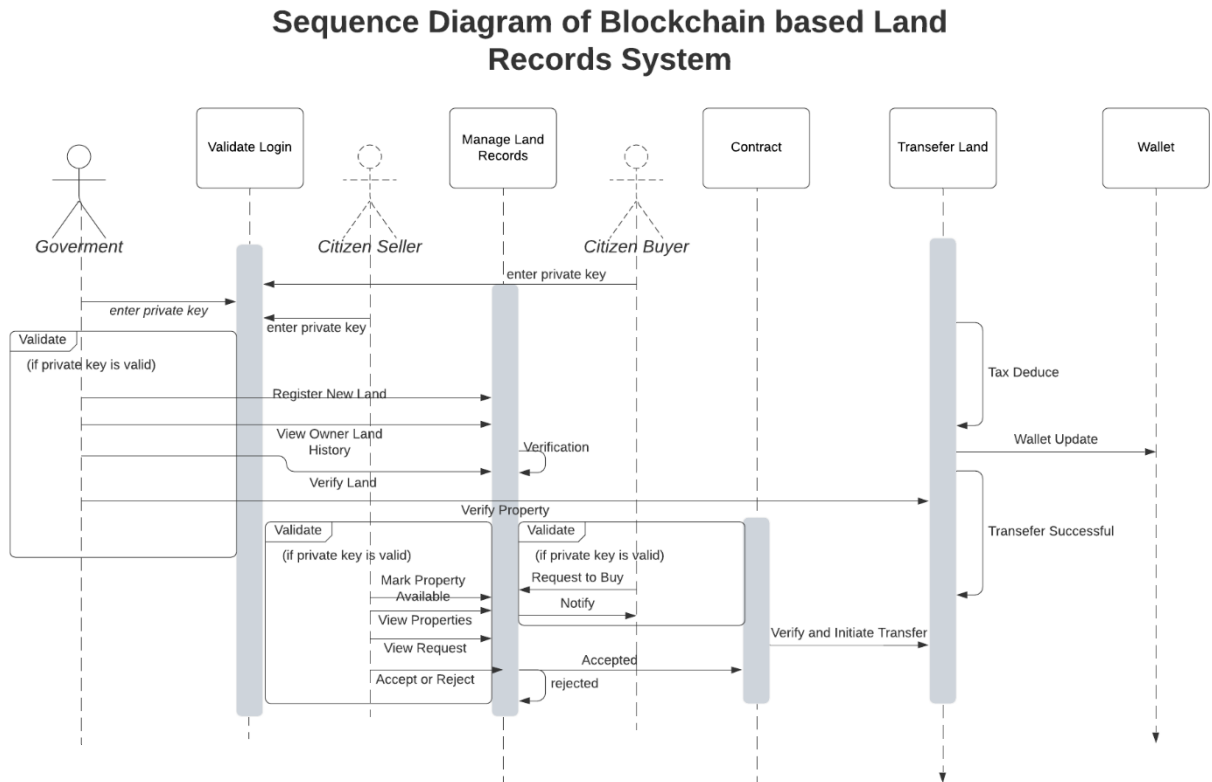
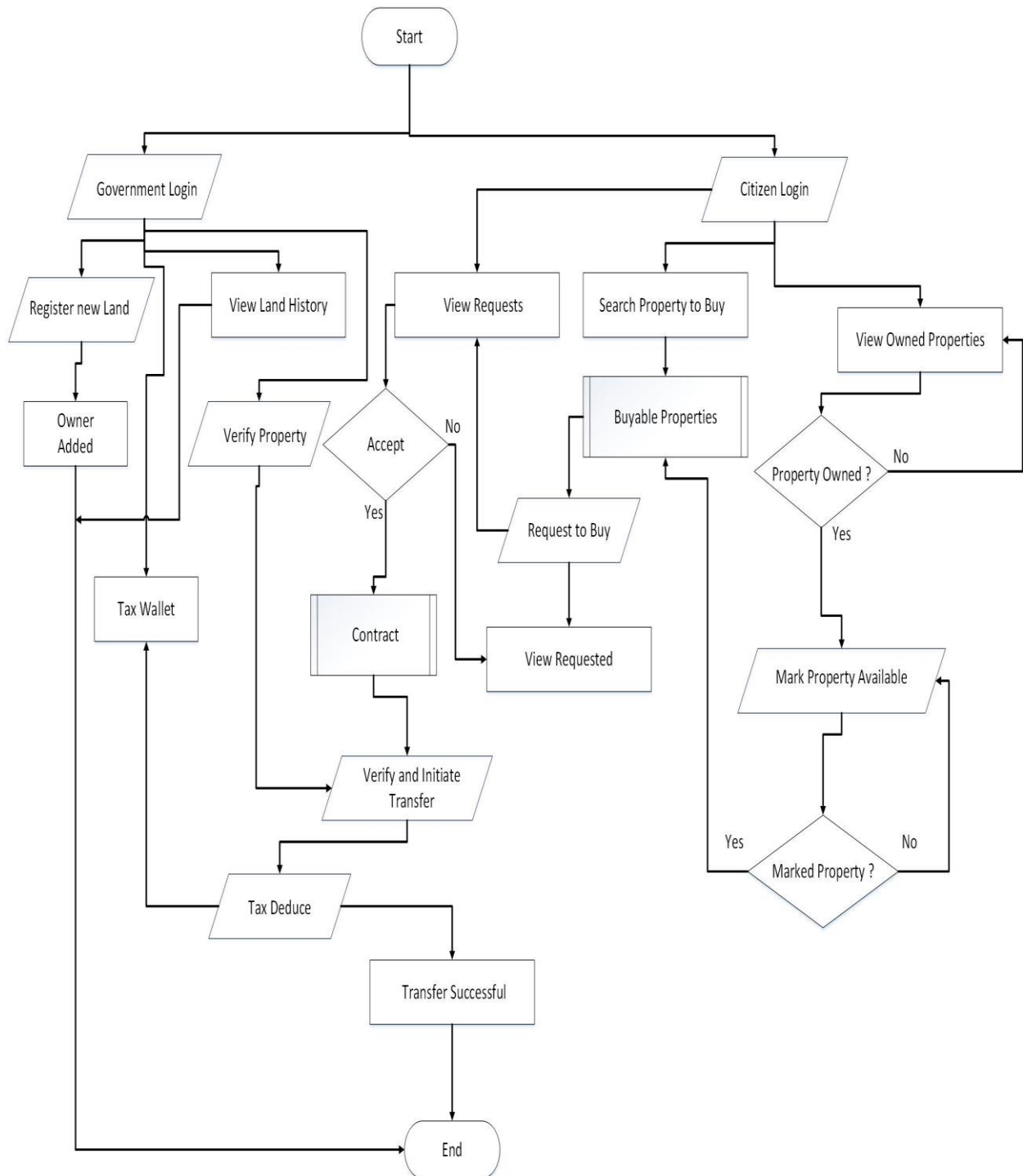


Figure: Sequence Diagram

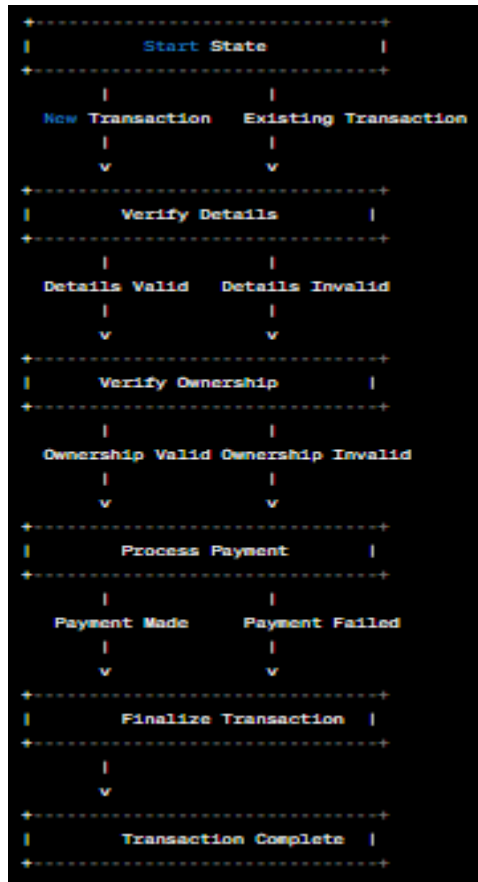
## 4.1. Operation contracts

<b>Preconditions:</b>	<p>PRE-1. All user Buyer, Seller, and Government are logged into Blockchain based Land Record System to use it.</p> <p>PRE-2. Both Buyer and Seller must be registered for proper usage of DApp and transfer of land into accounts.</p> <p>PRE-3. Government must verify the transfer of land and register new land with owner details.</p>
<b>Postconditions:</b>	<p>POST-1. For mark property to the sale must have ownership of the seller.</p> <p>POST-2. For sending buying requests to the seller, the buyer must have the above money to land value.</p> <p>POST-3. Government must verify the land before the transfer of land.</p>

## 4.2. Activity Diagram



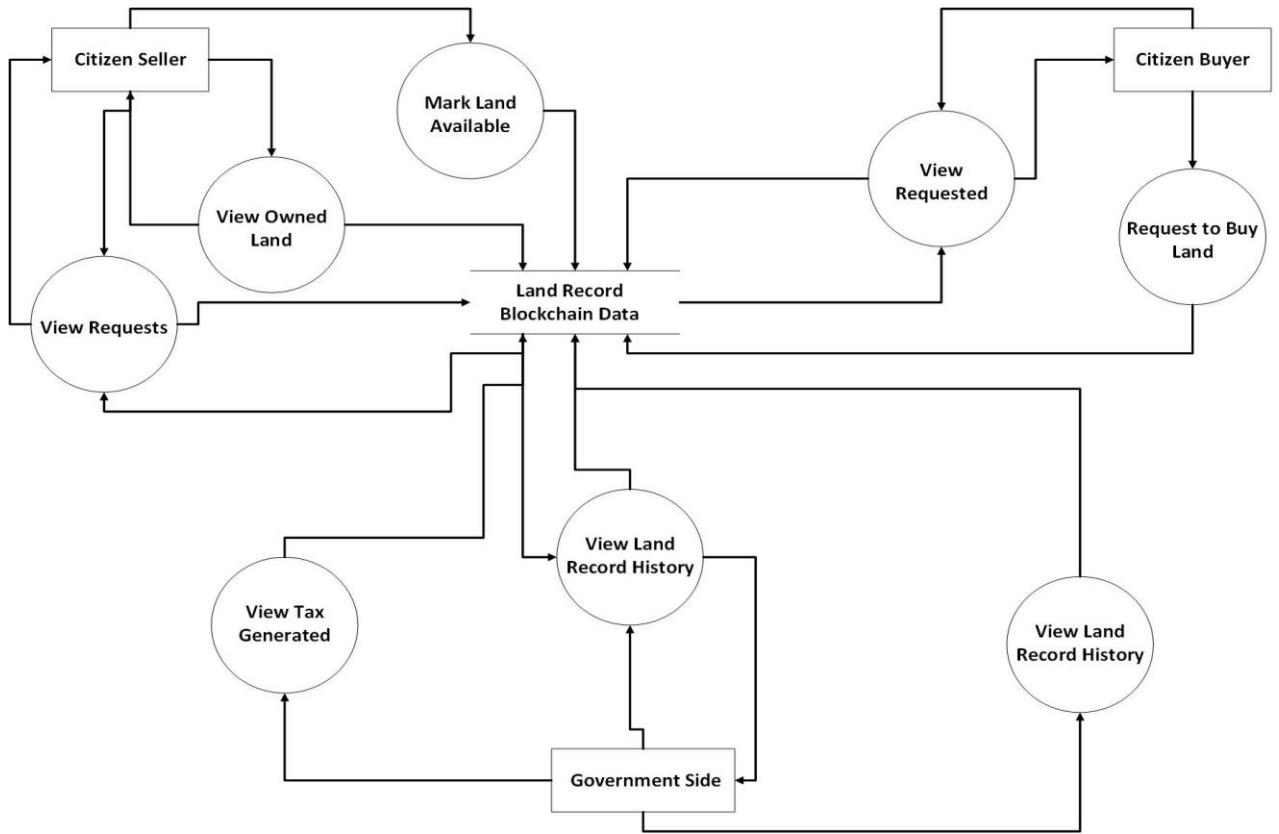
### 4.3. State Transition Diagram



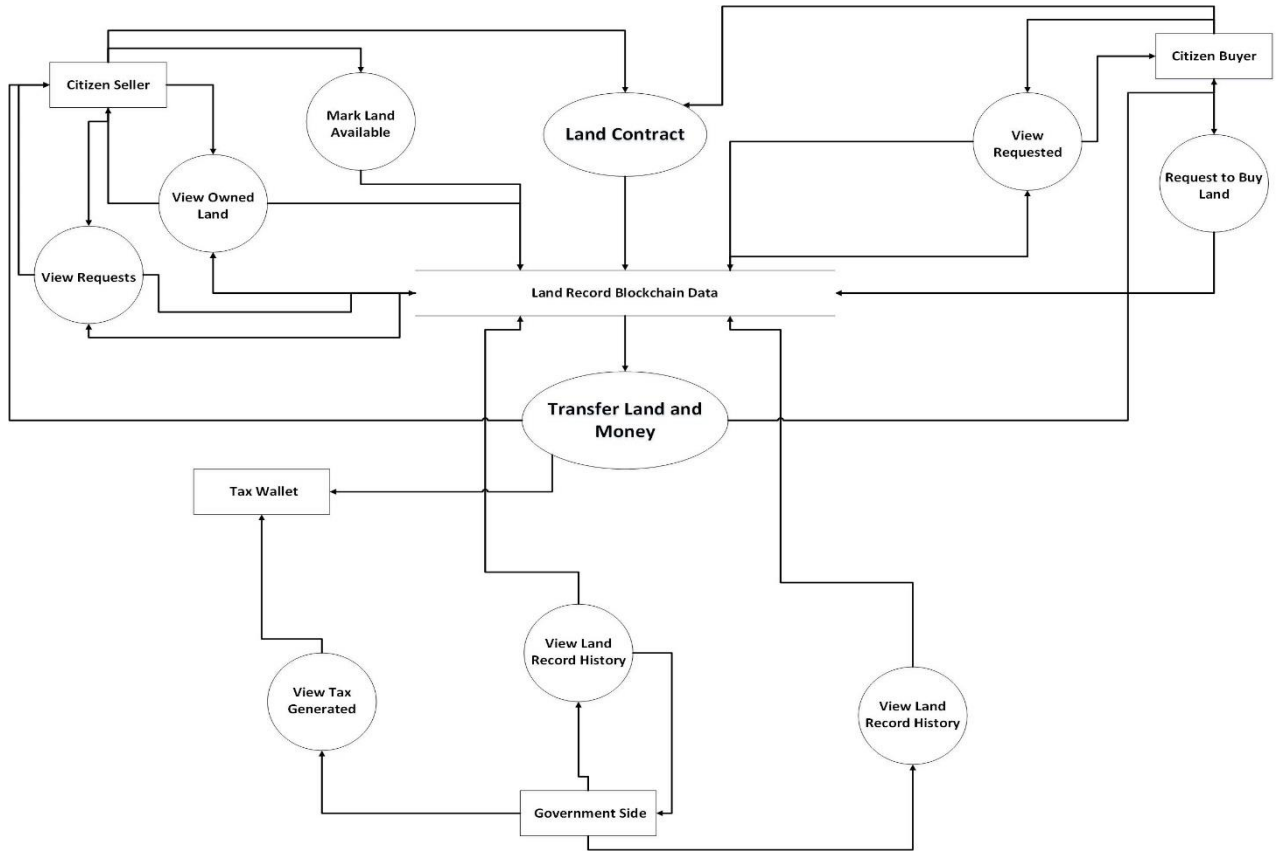
### 4.4. Data Flow diagram

This diagram represents the flow of data or process of the system. It gives complete information about the outputs and generated after providing the user’s inputs of each module and process itself.

**LEVEL 0:**



**LEVEL 1:**



# Chapter 5

## Implementation

## Chapter 5: Implementation

In the Blockchain, data design in such a way that is secure and immutable. During the data design process, smart contracts are specified along with the required and provided rules for the data. Each transaction is stored in a block; every block relates to the previous block with a hash value that makes a chain of the blocks. Data stored on the blockchain in the form of contracts and transactions. For Ethereum (the most popular blockchain for DApps), the library Web3.js is used to communicate with the blockchain. The way this works is that it uses something called the JSON-RPC protocol to query the blockchain. Government and Citizens act as a user to each other on a peer-to-peer network, everyone has the same authority. Government registers the new land and deeds of land stored on IPFS (interplanetary File System) is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. For testing, developing purposes, the Ganache is a personal blockchain that will be used in the system. Citizens can buy and sell land this system and tax automatically deduce and send to Government wallet when a transfer of land successful. Government can verify the land, register new land, and view tax generate.

### 5.1. Important Flow Control/Pseudo codes

- Smart Contracts are programs stored on the blockchain it automates the execution of an agreement.
- All transactions are stored on the blockchain; each block has different transactions. Each block connects with the previous block by encrypted hash. It provides security to the blockchain.
- The citizen must have to register to DApp.
- Sell and buy property.
- DApp has optimized search functionality, user can search the property to buy.
- The owner can mark the property for sale. Users can request the property to buy which available to buy.
- Users can check the owner of land that property buying.
- DApp automatically deduces the tax when the transfer of land from one owner to another.

- Government can register the land of the owner.
- Also, view land record history and tax generated by the transfer of lands.
- All the events happened in the form of transactions that will be store on the blockchain without third-party involvement.
- Blockchain makes DApp secure, transparent, immutable, and verifiable.
- Also, trace back to check the history of previous owners.

## 5.2. Deployment Environment

Screen objects containing the government side and citizen side. On the Government side, can register land, check land record history, verify the property and view tax generated. On the Citizen side, user can view property that own, mark property available to sale, search property to buy that available for sale and request to buy, verify the property owner.

## 5.3. Tools and Techniques

Tools	Version	Rationale
Visual Studio Code	2021	Coding and GUI
Ganache	2.5.4	Personal Blockchain for Ethereum
Git	2.26.2	For Version Controlling
Remix		Online IDE for building smart contracts
Rinkeby		Test Network
Ropsten		Test Network
Kovan		Test Network
MS Visio	2013	Gantt Chart, Use Case, Incremental Model Diagram, Flowchart, Activity Diagram.
MS Word for MSO 365	2102	Documentation
MS PowerPoint for MSO 365	2102	Presentation
Solidity	0.4.22	contract-oriented programming language for writing smart

		contracts
JavaScript		Programming language for business logic and GUI
Truffle		Development Framework provides (Ethereum Virtual Machine)
React	17.0.1	JavaScript library for building user interfaces
NodeJS	14.16.0	JavaScript run-time environment
IPFS	0.7	Distributed system for storing and accessing files
Hyperledger		Distributed ledger frameworks.
Web3.js	1.3.4	Ethereum JavaScript API
GitHub		Project Managing and Collaboration working as a team
Meta Mask	9.1.1	Accessing Ethereum enabled distributed applications
CSS	3	Styling

#### 5.4. Best Practices / Coding Standards

The DApp should run on every machine, Windows, Mac OS, Linux, Mobile Phones. and can be supported by any type and version of browsers installed in your operating system and the browser must have Meta Mask extension added. DApp also works on Android OS and IOS must have a browser with a Meta Mask extension. The operating system minimum has 4 GB RAM and needs good processing power to run the DApp for better results.

# Chapter 6

## Testing and Evaluation

## Chapter 6: Testing and Evaluation

### 6.1. Use Case Testing

- **Registration of Land Title:**

Test the ability to register a new land title on the blockchain-based system.

Verify that all required information, such as property details, owner information, and legal documentation, can be accurately recorded.

Test the validation of input data and ensure appropriate error messages are displayed for invalid or missing information.

- **Property Transfer:**

Test the process of transferring ownership of a property from one entity to another.

Verify that the system accurately updates the ownership records on the blockchain and reflects the new owner.

Test the integration with external systems (if applicable) to ensure proper synchronization of data during the transfer.

- **Title Verification:**

Test the functionality to verify the authenticity and validity of a land title.

Input various land title details and verify that the system can retrieve the corresponding record from the blockchain.

Validate that the retrieved information matches the input and ensure the system displays the correct title status (e.g., active, inactive, disputed).

- **Transaction History:**

Test the ability to view the transaction history of a specific land record.

Verify that the system displays a chronological list of all transactions related to a particular property.

Validate that the transaction details, such as date, time, parties involved, and transaction type, are accurately presented.

- **Smart Contract Execution:**

Test the functionality of smart contracts in automating processes, such as property transfers or title verification. Execute various scenarios to ensure that the smart contracts execute the predefined rules and conditions correctly. Verify that the system accurately reflects the outcomes of smart contract execution in the blockchain records.

- **Security and Access Control:**

Test the security measures implemented to protect sensitive data and prevent unauthorized access. Verify the effectiveness of encryption techniques used to secure land records and transactions. Test user authentication and authorization mechanisms to ensure proper access control for different user roles.

- **Performance and Scalability:**

Test the system's performance under different load conditions to ensure it can handle a significant number of concurrent users and transactions. Validate the response time for key operations and ensure they meet acceptable performance standards. Test scalability by adding a large volume of land records and transactions to verify that the system remains responsive and efficient.

- **Integration Testing:**

Test the integration between the blockchain-based land record system and any external systems, such as government databases or payment gateways.

Verify the seamless exchange of data and proper synchronization between systems during data updates or transactions.

Test data consistency and integrity across integrated systems.

- **Error Handling and Exception Testing:**

Test various error scenarios, such as invalid inputs, network disruptions, or system failures, and verify that the system handles these exceptions gracefully.

Validate that appropriate error messages are displayed to users, guiding them on how to rectify the issue or seek assistance.

- **User Experience Testing:**

Test the overall user experience of the web-based system, ensuring it is intuitive, user-friendly, and responsive. Validate that the user interface is consistent, visually appealing, and accessible

across different devices and browsers. Perform usability testing to gather feedback from users and make necessary improvements to enhance user satisfaction.

## 6.2. Equivalence partitioning

- **User Registration:**

Valid partitions:

- a) Valid email addresses with proper formatting.
- b) Valid passwords meeting the system's password policy.

Invalid partitions:

- a) Invalid email addresses (e.g., missing @ symbol, incorrect domain).
- b) Weak passwords that do not meet the system's password requirements.
- c) Empty or null values for email or password fields.

- **Property Details:**

Valid partitions:

- a) Properly formatted property addresses.
- b) Valid property ownership documents.

Invalid partitions:

- a) Property addresses with invalid characters or format.
- b) Invalid or expired ownership documents.
- c) Empty or null values for address or document fields.

- **Transaction Amount:**

Valid partitions:

- a) Positive numeric values representing valid transaction amounts.
- b) Decimal values with the appropriate precision required by the system.

Invalid partitions:

- a) Negative or zero values for transaction amounts.
- b) Non-numeric or non-decimal values.
- c) Empty or null values for the transaction amount field.

- **Land Title Verification:**

Valid partitions:

a) Existing land titles with correct identification details.

Invalid partitions:

a) Non-existent land titles or incorrect title identification.

b) Expired or revoked land titles.

c) Empty or null values for the land title identification field.

- **User Roles and Permissions:**

Valid partitions:

a) Authorized users with valid roles and corresponding permissions.

Invalid partitions:

a) Unauthorized users trying to access restricted functionalities.

b) Users with incorrect or expired roles.

c) Empty or null values for the user role field.

- **Blockchain Integration:**

Valid partitions:

a) Successful integration with the blockchain network.

Invalid partitions:

a) Failed or disrupted integration with the blockchain network.

b) Inconsistent data synchronization between the system and the blockchain.

### 6.3. Unit testing

- **Smart Contracts:**

Test the functionality of individual smart contracts by creating unit tests that cover different scenarios and contract methods.

Verify that smart contracts execute predefined rules and conditions accurately.

Test contract interactions and state changes, ensuring the desired outcomes are achieved.

- **Data Models and Entities:**

Test the data models and entities used in the system to ensure their integrity and proper functioning.

Verify that the properties and relationships between entities are correctly defined.

Test CRUD (Create, Read, Update, Delete) operations for each entity, ensuring data manipulation functions as intended.

- **User Authentication and Authorization:**

Test the user authentication and authorization mechanisms implemented in the system.

Verify that user credentials are validated correctly and access is granted or denied based on user roles and permissions.

Test different user scenarios, such as valid credentials, invalid credentials, or unauthorized access attempts.

- **Input Validation:**

Test the input validation functions to ensure that user inputs are properly validated and sanitized.

Verify that the system detects and handles invalid or malicious inputs, preventing security vulnerabilities or data corruption.

- **Error Handling:**

Test the system's error handling capabilities by intentionally triggering different error scenarios.

Verify that the system detects and handles errors gracefully, providing appropriate error messages or fallback actions.

- **Integration with Blockchain Network:**

Mock or simulate interactions with the blockchain network to test the integration between the system and the blockchain.

Verify that data is correctly sent to and retrieved from the blockchain.

Test different scenarios, such as network disruptions or failures, to ensure the system handles such situations properly.

- **System Configuration and Dependencies:**

Test the system's configuration settings and dependencies to ensure they are correctly set up.

Verify that the system integrates properly with external dependencies, such as databases or blockchain networks.

Test different configurations or dependency versions to ensure compatibility and stability.

- **Performance and Resource Usage:**

Test the performance and resource usage of individual components or units of code.

Verify that units of code perform optimally, meet performance requirements, and do not have excessive resource consumption.

## 6.4. Integration testing

- **Smart Contract Integration:**

Test the integration between the web application and smart contracts to ensure proper communication and data exchange.

Verify that the web application can invoke smart contract methods and retrieve data from the blockchain network accurately.

Test scenarios that involve different contract interactions, such as creating a new land record, updating ownership, or retrieving transaction history.

- **Blockchain Network Integration:**

Test the integration between the web application and the blockchain network to ensure seamless communication and data consistency.

Verify that transactions initiated from the web application are properly processed and recorded on the blockchain.

Test scenarios involving multiple nodes and validating the synchronization of data across the network.

- **User Authentication and Authorization:**

Test the integration between the web application and the user authentication and authorization mechanisms.

Verify that user roles and permissions are correctly enforced when accessing different system features and performing actions.

Test scenarios involving authenticated and unauthorized users, ensuring that access control is functioning as intended.

## 6.5. Performance testing

- **Performance and Scalability:**

Test the system's performance and scalability during integration.

Verify that the system can handle a large number of concurrent users and transactions.

Test performance metrics such as response time, throughput, and resource utilization.

## 6.6. Stress Testing

- **High User Load:**

Simulate a large number of concurrent users accessing the system simultaneously.

Measure the system's response time and throughput under the heavy user load.

Observe if the system can handle the expected number of users without significant performance degradation.

- **Transaction Volume:**

Generate a significant number of land record transactions within a short period.

Assess how the system handles the increased transaction volume.

Monitor the blockchain network to ensure that transactions are processed and recorded without delays or failures.

- **Resource Exhaustion:**

Simulate scenarios where the system's resources, such as CPU, memory, or network bandwidth, are heavily utilized.

Evaluate the system's behavior and performance under resource exhaustion conditions.

# Chapter 7

## Summary, Conclusion and Future Enhancements

## Chapter 7: Summary, Conclusion & Future Enhancements

### 7.1. Project Summary

In the current era, the buying and selling of lands and transfer of lands are active daily processes in any city. Also, there is a lot of land conflict rise using the current system. The current system is on paper or on a digital system which creates conflict. Blockchain technology will overcome these kinds of problems. For the Land Record use case blockchain is best to build the system on the blockchain, it provides the immutability that no one can tamper with the data of land records. Decentralize system without anyone involvement of the third party, the citizen can buy and sell the property. The record is secure in the form of a hash and all data is stored in blocks and each block has different transactions that provide the facility of traverse to check the previous owners of the land. The Blockchain-based Records System will be a DApp (Decentralized Application) that government has functionalities that can register the property of the owner, view the land record history, buy, and sell property, and verify the transfer of land transactions. On the other side, a citizen can buy and sell property, mark the property for sale, and request for sale, and after successful transfer, tax is automatically deducted from both sides and it transfers to the government with detail. All the details of each transaction will be stored on a blockchain network that is secure, immutable, and transparent. By using blockchain for land record use cases the problem in the current system of land records will be solved, also save time cost and reduce expenses.

### 7.2. Achievements and Improvements

**Improved Transparency:** The blockchain-based land record system has brought increased transparency to the land registration process. All transactions and ownership transfers are recorded on the blockchain, ensuring an immutable and auditable history of land records.

**Enhanced Security:** The system leverages the inherent security features of blockchain technology, such as cryptographic hashing and decentralized consensus. This provides a higher level of security and prevents unauthorized modifications to land records.

**Efficient Transaction Processing:** By using smart contracts and blockchain technology, the system automates and streamlines the land registration process. This reduces the need for manual paperwork, eliminates intermediaries, and accelerates transaction processing times.

**Reduced Fraud:** The immutable nature of blockchain prevents fraudulent activities such as tampering with land records, forging ownership documents, or unauthorized changes to transaction history. This helps to safeguard the integrity of land ownership and reduce fraudulent practices.

**Improved Accessibility:** The web-based interface of the system makes land records easily accessible to authorized users from anywhere at any time. This enhances convenience and eliminates the need for physical visits to government offices or land registry departments.

### 7.3. Critical Review

#### Strengths

**Transparency:** The use of blockchain technology brings transparency to the land record system by providing an immutable and auditable history of transactions and ownership transfers. This helps to mitigate fraud and disputes.

**Security:** Blockchain's cryptographic mechanisms and decentralized nature enhance the security of land records, preventing unauthorized modifications and ensuring data integrity.

**Efficiency:** Automating the land registration process through smart contracts and blockchain technology reduces paperwork, eliminates intermediaries, and speeds up transaction processing, resulting in greater efficiency.

**Accessibility:** The web-based interface of the system allows authorized users to access land records conveniently from anywhere, eliminating the need for physical visits to government offices.

## Weaknesses

**Scalability Challenges:** Blockchain networks can face scalability issues due to the resource-intensive consensus mechanisms. As transaction volumes increase, the system may experience slower processing times and higher transaction costs.

**User Adoption:** Adoption of the system by all relevant stakeholders, including government agencies, land registry offices, and users, may pose a challenge. Training and education programs may be required to ensure widespread acceptance and usage.

**Integration Complexity:** Integrating the blockchain-based land record system with existing government systems, external services, and legacy databases can be complex and time-consuming. Ensuring seamless data exchange and interoperability may present challenges.

## 7.4. Lessons Learnt

**Thorough Requirement Analysis:** Conduct a comprehensive analysis of the requirements and expectations of all stakeholders involved in the land record system. This ensures that the project addresses their specific needs and aligns with their goals.

**Clear Governance and Decision-Making:** Establish a clear governance framework for the blockchain network and decision-making processes. Define roles and responsibilities, consensus mechanisms, and dispute resolution procedures to ensure smooth operation and effective management.

**User-Centric Design:** Prioritize user experience and usability throughout the project. Involve end-users in the design process, gather feedback, and continuously iterate to improve the system's interface and functionality.

## 7.5. Future Enhancements/Recommendations

**Mobile Application:** Develop a mobile application alongside the web interface to provide users with convenient access to land records on their smartphones or tablets. This will further improve accessibility and user experience.

**Geospatial Integration:** Integrate geospatial data into the system to provide users with visual representations of land records on maps. This can enhance the usability and understanding of land information, especially for surveyors, land developers, and urban planners.

**Digital Signatures:** Implement digital signature capabilities to ensure the authenticity and integrity of land record documents. This can streamline the process of document signing and verification, making it more secure and efficient.

**Blockchain Interoperability:** Explore interoperability solutions to connect with other blockchain networks or land record systems. This can facilitate seamless data exchange and collaboration between different jurisdictions, enabling the system to scale and serve a wider user base.

# References And Bibliography

## References

S. Humdullah, S. H. Othman, M. N. Razali, and H. K. Mammi, "Secured data storage framework for land registration using blockchain technology," in 2021 3rd International Cyber Resilience Conference (CRC), pp. 1–6, IEEE, 2021

F. Ullah and F. Al-Turjman, "A conceptual framework for blockchain smart contract adoption to manage real estate deals in smart cities," *Neural Computing and Applications*, vol. 1-22, 2021.

M. Abdullah, W. A. L. T. E. R. De Vries, and Z. Ali, "Assessing the performance of land administration system in Punjab after land records computerization," in Conference: 2020 World Bank Conference On Land and Poverty At: The World Bank-Washington DC, pp. 1–24, Washington, 2020.

## Plagiarism Report:

