**Distributed Land Management System Over**

**Blockchain**



**Tauqeer Hassan**

**Inayat Ullah**

**Arslan Hayat**

*Supervised By:*

**Dr. Basit Shahzad**

*Submitted for the partial fulfillment of BS Software Engineering degree to the Faculty of Engineering and Computer Science*

**NATIONAL UNIVERSITY OF MODERN LANGUAGES**

**ISLAMABAD**

**DECEMBER, 2019**

**ABSTRACT**

Land management and registration system is the system for storing land information and managing the records involving land trade and transformation of ownership. Due to the sensitivity of land’s data issues, land management and registration system should be strong to avoid any document forgery, available all the time, and take a short time to complete tasks. The data in the current land management system is not secured. The data can be tempered with and has a single point of failure. On the other hand, currently, trade of land is done with the help of real estate agents that cover a very small market and specific area only, which limits the users to few options. Our system covers all these limitations and problems. The proposed system is a blockchain-based land system that will allow users to register, verify and trade lands. Trading includes buying and selling of lands. Land records of individuals will be private and they can control the privacy of their land information. The system will also include a digital map that will cover the map that patwaris have (with specific terms/jargons such as Khassra, Khevat numbers). The system will be a web-based application. Record management of land and individuals will be based on hyperledger fabric blockchain.

TABLE OF CONTENTS

[CHAPTER 1 3](#_Toc26830195)

[INTRODUCTION 3](#_Toc26830196)

[1.0 Introduction 4](#_Toc26830197)

[1.1 Motivation 4](#_Toc26830198)

[1.2 Problem statement 4](#_Toc26830199)

[1.3 Scope of the System 5](#_Toc26830200)

[1.4 Process Model 6](#_Toc26830201)

[1.4.1 Requirement Gathering 6](#_Toc26830202)

[1.4.2 Identifying Modules 6](#_Toc26830203)

[1.4.3 Designing 6](#_Toc26830204)

[1.4.4 Coding 6](#_Toc26830205)

[1.4.5 Testing 7](#_Toc26830206)

[1.4.6 Deployment 7](#_Toc26830207)

[1.4.7 Why the Incremental Model 7](#_Toc26830208)

[1.5 Proposed System 7](#_Toc26830209)

[1.6 Features 8](#_Toc26830210)

[1.6.1 Land Registration 8](#_Toc26830211)

[1.6.2 Land Verification 8](#_Toc26830212)

[1.6.3 Land Trade 8](#_Toc26830213)

[1.6.4 Buy the Land 9](#_Toc26830214)

[1.6.5 Sell the Land 9](#_Toc26830215)

[1.6.6 Digital Map 9](#_Toc26830216)

[1.7 Nature of The Project 9](#_Toc26830217)

[1.8 Summary 9](#_Toc26830218)

[CHAPTER 2 11](#_Toc26830219)

[BACKGROUND AND EXISTING WORK 11](#_Toc26830220)

[2.0 Introduction 12](#_Toc26830221)

[2.1 Existing System 12](#_Toc26830222)

[2.1.1 Land registration system in hard copies 12](#_Toc26830223)

[2.1.2 Digitalized Land Registration System 12](#_Toc26830224)

[2.1.3 Zameen.com 13](#_Toc26830225)

[2.2 Limitations of existing systems 13](#_Toc26830226)

[2.2.1 Temperable 13](#_Toc26830227)

[2.2.2 Non-Friendly 13](#_Toc26830228)

[2.2.3 Time-consuming 13](#_Toc26830229)

[2.2.4 Disputed 14](#_Toc26830230)

[2.2.5 Hard to verify 14](#_Toc26830231)

[2.2.6 Privacy issues 14](#_Toc26830232)

[2.2.7 Real estate agents 14](#_Toc26830233)

[2.3 Solutions to limitations of the existing system 14](#_Toc26830234)

[2.3.1 Non-Temperable 15](#_Toc26830235)

[2.3.2 Friendly 15](#_Toc26830236)

[2.3.3 Less time consuming 15](#_Toc26830237)

[2.3.4 Zero Disputes 15](#_Toc26830238)

[2.3.5 Verification 15](#_Toc26830239)

[2.3.6 Secure and private 16](#_Toc26830240)

[2.3.7 Eliminating Real Estate Agent 16](#_Toc26830241)

[2.4 Comparison of existing systems and proposed system 16](#_Toc26830242)

[2.5 Summary 17](#_Toc26830243)

[CHAPTER 3 19](#_Toc26830244)

[REQUIREMENTS SPECIFICATION 19](#_Toc26830245)

[3.0 Introduction 20](#_Toc26830246)

[3.1 Interface Requirements 20](#_Toc26830247)

[3.1.1 Hardware Interface Requirements 20](#_Toc26830248)

[3.1.2 Software Interface Requirements 21](#_Toc26830249)

[3.2 Functional Requirements 21](#_Toc26830250)

[3.2.1 Land Registration 21](#_Toc26830251)

[3.2.2 Land Verification 21](#_Toc26830252)

[3.2.3 Buy the Land 21](#_Toc26830253)

[3.2.4 Sell the Land 22](#_Toc26830254)

[3.2.5 Digital Map 22](#_Toc26830255)

[3.3 Use Case Diagram 22](#_Toc26830256)

[3.3.1 Verify the Land 26](#_Toc26830257)

[3.3.2 Buy the Land 28](#_Toc26830258)

[3.3.3 Sell the Land 30](#_Toc26830259)

[3.3.4 Use the Digital Map use case 32](#_Toc26830260)

[3.4 Non-Functional Requirements 33](#_Toc26830261)

[3.4.1 Performance Requirements 34](#_Toc26830262)

[3.4.2 Security 34](#_Toc26830263)

[3.4.3 Flexibility 34](#_Toc26830264)

[3.4.4 Testability 34](#_Toc26830265)

[3.4.5 Maintainability 34](#_Toc26830266)

[3.5 Resource Requirements 34](#_Toc26830267)

[3.5.1 Blockchain 34](#_Toc26830268)

[3.5.2 Tools 35](#_Toc26830269)

[3.6 Project Feasibility 36](#_Toc26830270)

[3.6.1 Technical Feasibility 36](#_Toc26830271)

[3.6.2 Operational Feasibility 36](#_Toc26830272)

[Legal & Ethical Feasibility 37](#_Toc26830273)

[3.7 Summary 37](#_Toc26830274)

[CHAPTER 4 38](#_Toc26830275)

[SYSTEM MODELLING 38](#_Toc26830276)

[4.0 Introduction 39](#_Toc26830277)

[4.1 System Design 39](#_Toc26830278)

[4.2 Design Approach 39](#_Toc26830279)

[4.2.1 Top-Down Design Approach 39](#_Toc26830280)

[4.3 Interface Design 41](#_Toc26830281)

[4.3.1 High Fidelity Prototype 41](#_Toc26830282)

[4.4 4+1 View Model of Architecture 44](#_Toc26830283)

[4.4.1 Logical View 45](#_Toc26830284)

[4.5 Process View 47](#_Toc26830285)

[4.5.1 Activity Diagram 47](#_Toc26830286)

[4.5.2 Sequence Diagram 49](#_Toc26830287)

[4.6 Development View 54](#_Toc26830288)

[4.6.1 Component Diagrams 55](#_Toc26830289)

[4.7 Physical View 56](#_Toc26830290)

[4.7.1 Deployment Diagram 56](#_Toc26830291)

[4.8 The use case view 56](#_Toc26830292)

[4.9 Summary 57](#_Toc26830293)

[5.0 Introduction 59](#_Toc26830294)

[5.1 Modules Our System 59](#_Toc26830295)

[5.1.1 Land Registration 59](#_Toc26830296)

[5.1.2 Land Verification 59](#_Toc26830297)

[5.1.3 Land Trade 60](#_Toc26830298)

[5.1.4 Digital Map 60](#_Toc26830299)

[5.2 Tools and Technology 60](#_Toc26830300)

[5.2.1 React.js 60](#_Toc26830301)

[5.2.2 Blueprint.js 60](#_Toc26830302)

[5.2.3 Node.js 61](#_Toc26830303)

[5.2.4 Hyperledger Fabric 61](#_Toc26830304)

[5.2.5 IPFS 61](#_Toc26830305)

[5.2.6 GO 61](#_Toc26830306)

[5.2.7 AWS/EC2 61](#_Toc26830307)

[5.2.8 Micro services 62](#_Toc26830308)

[5.2.9 Docker 62](#_Toc26830309)

[5.2.10 Google geolocation API 62](#_Toc26830310)

[5.3 Summary 62](#_Toc26830311)

[CHAPTER 6 63](#_Toc26830312)

[TESTING, ANALYSIS, AND VALIDATION 63](#_Toc26830313)

[6.0 Introduction 64](#_Toc26830314)

[6.1 Test Bed 65](#_Toc26830315)

[6.2 System Test Case 65](#_Toc26830316)

[6.3 Test Cases 65](#_Toc26830317)

[6.3.1 Registration of the Land 66](#_Toc26830318)

[6.3.2 Verification of the land 67](#_Toc26830319)

[6.3.3 Buy the land 68](#_Toc26830320)

[6.3.4 Sell the land 70](#_Toc26830321)

[6.3.5 Digital Map 71](#_Toc26830322)

[6.4 Summary 73](#_Toc26830323)

[CHAPTER 7 74](#_Toc26830324)

[CONCLUSION AND FUTURE WORK 74](#_Toc26830325)

[7.0 Overview 75](#_Toc26830326)

[7.1 System overview 75](#_Toc26830327)

[7.2 Milestones Achieved 75](#_Toc26830328)

[7.2.1 Land Registration 75](#_Toc26830329)

[7.2.2 Land Verification 75](#_Toc26830330)

[7.2.3 Land Trade 76](#_Toc26830331)

[7.2.4 Digital Map 76](#_Toc26830332)

[7.3 Limitations 76](#_Toc26830333)

[7.4 Future Work 77](#_Toc26830334)

[7.5 Summary 77](#_Toc26830335)

[8.0 References 79](#_Toc26830336)

**LIST OF FIGURES**

[Figure 3. 1 Use case diagram of the distributed land management system 23](#_Toc26830637)

[Figure 3. 2 Use case diagram of register the land 24](#_Toc26830638)

[Figure 3. 3 Use case diagram of verifying the land 26](#_Toc26830639)

[Figure 3. 4 Use case diagram of buy the land 28](#_Toc26830640)

[Figure 3. 5 Use case diagram sell the land 30](#_Toc26830641)

[Figure 3. 6 Use case diagram of digital map 32](#_Toc26830642)

[Figure 4. 1 Top-down approach view of the system---------------------------------------------- 40](#_Toc26830658)

[Figure 4. 2 Class diagram of a system 46](#_Toc26830659)

[Figure 4. 3 Activity diagram of a system 48](#_Toc26830660)

[Figure 4. 4 Sequence diagram of register the land 50](#_Toc26830661)

[Figure 4. 5 Sequence diagram of verifying the land 51](#_Toc26830662)

[Figure 4. 6 Sequence diagram of buy the land 52](#_Toc26830663)

[Figure 4. 7 Sequence diagram of sell the land 53](#_Toc26830664)

[Figure 4. 8 Sequence diagram of digital map 54](#_Toc26830665)

[Figure 4. 9 Development diagram of a system 55](#_Toc26830666)

[Figure 4. 10 Component diagram of the system 56](#_Toc26830667)

**LIST OF TABLES**

[Table 1. 1 Scope of the project 5](#_Toc26830762)

[Table 2. 1 Comparison of an existing system and proposed system features -------------------17](#_Toc26830768)

[Table 3. 1Register the land-use case-----------------------------------------------------------------25](#_Toc26830781)

[Table 3. 2Verify the land-use case 27](#_Toc26830782)

[Table 3. 3 Buy the land-use case 29](#_Toc26830783)

[Table 3. 4 Sell the land-use case 31](#_Toc26830784)

[Table 3. 5 Digital map use case 32](#_Toc26830785)

[Table 6. 1 Registration of the land test case-------------------------------------------------------- 66](#_Toc26830786)

[Table 6. 2 Verification of the land 67](#_Toc26830787)

[Table 6. 3 Buy the land test case 69](#_Toc26830788)

[Table 6. 4 Sell the land test case 70](#_Toc26830789)

[Table 6. 5 Digital Map Test Case 71](#_Toc26830790)

# 

# CHAPTER 1

# INTRODUCTION

# Introduction

The land is a very large source of economy for any country. Especially when we talk about Pakistan which has an agriculture-based economy, the importance of lands becomes very high. Unfortunately, in Pakistan, we are still hanging in physical registers for the registration and management of records of lands which cause many issues to the related people.

In order to address these problems, we are composing a web-based application for the management of land records. The user will be able to register his land’s record and verify his record. This system will also provide the selling and purchasing of the land. The user will get his desired plot (a piece of land) by putting the information about the plot he wants to purchase. Similarly, the user will put information about his plot which he wants to sell. This system will also consist of a map where the user will check its land by entering the required information about his land.

We will do it through blockchain technology which means records will be not be tempered. It will be secured and easily verifiable. It will almost finish the dispute regarding the ownership of the lands. Ownership will be easily verifiable by putting the required information.

## Motivation

The biggest motivation to do this project is the current system of land management in our country, which is corruptible and easy to temper with. Pakistan is an agrarian country and it should have a safe and secure system of management of land. Keeping all the limitations of the current systems in mind we are motivated to develop a safe and secure land management system by using blockchain technology.

## Problem statement

In our country, land management is done physically which means that the data is stored in hard copies (in registers of patwari). In some districts, the government has shifted this to digital systems but still, the data is insecure. The data can be tempered with and has a single point of failure. On the other hand, currently, trade of land is done with the help of real estate agents that cover a very small market and specific area only which limits users to few options.

## Scope of the System

Table 1. 1 Scope of the project

|  |  |
| --- | --- |
| **SCOPE OF THE PROJECT** | |
| **Title** | Land Management and Information System Over Blockchain. |
| **Description** | This system is a web-based application that allows the user to register his land, verify his registered land and trade the land by using this system |
| **Justification** | The purpose of this project is to facilitate the common man in registering his land and reduce the efforts in the verification of the land. Land trade and the change of ownership has been made easier. |
| **Constraints** | The complexity of blockchain technology |
| **Assumption** | Availability of tools, Availability of Internet connection. |
| **Stakeholders** | User, Admin, Developer, Tester, Evaluation Team. |
| **Risk** | The complexity of blockchain technology |
| **Deliverables** | Web-based application, Documentation of the project. |

## Process Model

We have followed the incremental process model for the development and implementation of the Land Management and Information System Over Blockchain. The incremental process model splits the project into its modules, then these modules are developed and delivered iteratively. In incremental process model phases of the waterfall model is followed for the development of each module. Increment process model includes the following steps in a cycle;

1. Requirement Gathering
2. Identifying modules.
3. Designing
4. Coding
5. Testing
6. Deployment

### Requirement Gathering

In this phase, requirements and specifications that are needed for the development of “Land Management and Information System over Blockchain” are collected.

### Identifying Modules

In this phase, requirements are split into modules such as Land Registration, Land Trade, Land Verification, and Digital Map.

### Designing

In the designing phase, solutions regarding each module are identified and high-end functions have been designed in this phase. This phase has been repeated during the development of each module.

### Coding

In this phase, coding has been done on the bases of the proposed solution. This phase has also been repeated in the development of each module.

### Testing

The testing of each module has been done in this phase, to assure the quality of the product.

### Deployment

In the Deployment phase, we have a standalone working module that can be deployed.

### Why the Incremental Model

Reason for choosing an incremental process model is that:

1. We have done modular development in our project.
2. Modular development is flexible and improvements can be done within the module.
3. Errors can be identified easily.
4. This model can be followed by the less skillful team.

## Proposed System

We have proposed the system to provide a trustworthy solution to the stated problems. We will build a web application that will allow registration of land of people, one can verify registered lands from our platform if he is allowed by the owner of the land. Once the land is registered, its future record and trade will be maintained on our platform. We will use hyper ledger Fabric in node.js as a backend platform and smart contract to declare land as a digital value which can be further divided and traded in the form of transactions.

The owner of land can set his lands (if he has more than one) visibility to either public or private or only visible to some specific users. This will keep his privacy and will also allow 3rd party verifications. We will allow land’s registration and searching of land through google maps so that it can be easy for users to find a different area or land, registered lands will appear on our personal google map(or if we use some other service) [1]**.**

## Features

The following are the four main features of our platform.

### Land Registration

This feature will allow users of the system to register his land on the system. The user will enter his valid CNIC number and proof of ownership of the land to register it on the system.

The process of land registration will have the following steps:

1. Documents of land are verified, and its record is verified from the patwari office.
2. After a successful verification record of land is put on our system against a unique ID (signature) of the owner of Land.
3. Further records of that land can only be changed through that unique ID (signature) that’ll be only with the Owner of land.
4. The record of any user can be verified by a government signature that will be held by some single government authority.

### Land Verification

This feature of the system will allow users to verify them and verify another land. The user can verify by entering the specific id of land which was generated by the system at the time of registration and the id of the owner of the land only if the owner has allowed a user to view his land. The user can only read the information on land.

### Land Trade

One main feature of our system is trading land. We will provide a platform to trade land as we will be registering land as well. So, our platform will provide only authentic lands and fast land trade instead of the local and conventional physical trade system.

### Buy the Land

This feature of the system will allow users to buy the land. A user will enter the name of a city, location, and the area of land, property type (plot or cultivable land) and his range of price. The system will show all the suitable lands for that search.

### Sell the Land

This feature of the system will allow users to sell the land. The user needs to register his land first at our platform and then enter required basic information about his lands like the name of a city, location, the area of land, property type (plot or cultivable) and his demand price.

### Digital Map

This feature of the system will allow users to check the location of the land.  The user will enter the information about the land and the map will locate the desired land. Our map will replace the register and map patwaris have, with specific terms Khasra number, Kheevat, khatooni, etc

## Nature of The Project

Land Management and Information System over Blockchain is a web-based application. This system is developed by using blockchain and hyperledger fabric.

## Summary

In this chapter, the highlighted need is to develop a distributed land management system. Land management is done physically in most parts of our country which is easy to temper. In some districts, the government has digitalized the register system, but still, data is corruptible with a single point of failure. Then we discussed the proposed system by keeping all these problems in mind. The proposed system is based on blockchain, the immutable, decentralize and transparent nature of blockchain almost finishes the chances of tempering the data, unauthorized access, and disputes regarding lands. The system has the feature of registration of the land, verification of the land. Furthermore, our system contains the features of buying and selling of the properties & maps of registered lands.

As the proposed system is based on blockchain technology so this system will do a lot of good to people especially underprivileged and poor people who become a victim of a corrupt land system which still exists in majority part of the country.

# CHAPTER 2

# BACKGROUND AND EXISTING WORK

# Introduction

In this chapter, we are going to discuss the existing systems of land management and land trade. At first land management used to be done physically means data were stored in a register of patwari (which is still exists in many cities of our country). Then the government of Punjab and KP has digitalized the land records management. So that data is stored in computers instead of physical registers. This digital system “Punjab Land Records Authority” provides a feature of land registration & verification. Another existing system is Zameen.com which has the features of buying and selling of the property. These previously build systems have many limitations like they mutable, disputed, time-consuming and have privacy issues. To cover all the limitations of the existing systems we have proposed a web-based land management system using blockchain, as the immutability, decentralization, and transparency of blockchain will assure that tempering the data, unauthorized access, and disputes regarding lands should be avoided.

## Existing System

The following are some existing systems.

### Land registration system in hard copies

Currently, all the record of Pakistan’s land is managed physically on hard copies by patwaris who are designated in each village or council. They keep the record on their registers, once the land is traded, they verify the owners of that land from their register and after the verification process, they update that transaction of land on a new page or the register. This process is quite time consuming and has a central point of failure.

### Digitalized Land Registration System

Punjab and KPK governments are shifting land records from hard copies to computers that are a good step and helps in verifying and finding information about the land. But this also has a single point of failure as well as the database is centralized and mutable which means anyone has enough power to effect PLRA (Punjab Land Records Authority) and can modify a record of a system. And it only allows registration of a piece of land not the update on trade of land [2].

### Zameen.com

Zameen.com is the property trade platform in Pakistan which provide the facility of buying and selling of the property. It also provides the facility of taking property on rent giving property on rent too. It displays the trend of buying and selling in famous housing societies of the country[3].

## Limitations of existing systems

The following are the problems in the conventional LRMS (Land record management system).

### Temperable

The records stored in the system are easily temperable.

### Non-Friendly

The reading and writing procedure of record or new transaction either new entry of record or changing the owner of the previous land/property is very unfriendly. The current system is based on registers and patwaris offices. Now the government is trying to digitalize the system replacing registers with computers to store records, but this also comes with more new problems like hacking, permanent data loss, the central point of failure & security issues, etc.

### Time-consuming

Purchasing or selling land/property in the current system consumes a lot of time due to the non-efficient ways and long procedures. Mostly time varies for more than 2 weeks.

### Disputed

Due to temperable a centralized and non-trusted organization either the patwaris or a specific government institute records can be modified easily by any other person so single property/land can have multiple owners, transfer of record from one owner to another is quite easy without involving the owner of the property. It causes disputes between people for ownership of the land.

### Hard to verify

Sometimes, Banks, police or other 3rd party investigators need to verify a piece of land for some person, only this process of reading and verifying a property wastes time of these 3rd party verifiers.

### Privacy issues

Land and property information of individuals can be easily retrieved because it is insecure, unsecured here does not mean DB hacks or system hacks, but data is stored non-cryptographically in DB’s. So, anyone who can access the DB can easily read personal data, land information and a number of properties of specific individuals.

### Real estate agents

Purchasing and selling of land is not that easy and cheap process in this era. To get information about properties, people go-to real estate agents who charge fees and they only cover local properties (small market) have fewer sellers and buyers which means they can entertain fewer people.

## Solutions to limitations of the existing system

The following are the solutions provide by our system to address the limitation of the existing systems.

### Non-Temperable

One of the main benefits of using blockchain is the overcoming of tempering issue. Blockchain does not support UD (update and deleting) operations which means data once written on blockchain ledger can’t be deleted or updated [4].

### Friendly

We will digitalize the system put the record from a register to computers which will help to retrieve and writing data easily and swiftly. We will use google maps as the main feature to search and view lands that are registered on our platform. We will have an interactive UI as compared to previous systems.

### Less time consuming

Time consumption of the current system can be overcome by digitalizing the whole procedure. Contracts and agreement for transferring land will be made digital and signed digitally with digital signatures of involved parties (such as a government body, seller & buyer).

### Zero Disputes

The disputation problem will be overcome once the land is verified and shifted to the blockchain, after that each transfer, division or modification operation on that piece of land will be stored as a transaction in Blockchain that can’t be removed which means once can easily verify all the history of the land. And can buy undisputed land easily [4].

### Verification

Verifying properties of individuals and owners of a property can be made easy and a swift process once they are digitalized. Reading and verifying data would be very easy from a computer with respect to patwari offices and finding a record for a piece of land from patwari’s registers.

### Secure and private

Data stored in our system will be stored through secure hashing algorithms sha256, etc. And the owner of the property will decide and grant permissions of reading his record to a specific user [4].

### Eliminating Real Estate Agent

Real estate agents only cover a small specific area for land trade. Our solution will cover the whole country and provide users a platform to trade, sell and buy land/property easily, as we will also be verifying lands/properties so cases of disputes and untrusted properties would be zero.

Currently, Zameen.com and OLX have a very unfriendly UI and users buying property on their platform must verify that land themselves.

## Comparison of existing systems and proposed system

A comparison between the proposed system and the existing systems is given below in the form of a table.

Table 2. 1 Comparison of an existing system and proposed system features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Hard Registers (Patwaris) | PLRA (Punjab Land records authority) | Zameen.com | Proposed system |
| Digital record storing | Description: Close | Description: Checkmark | Description: Close | Description: Checkmark |
| Secure record storing (No-single point of failure) | Description: Close | Description: Close | Description: Close | Description: Checkmark |
| Land trade | Description: Close | Description: Close | Description: Checkmark | Description: Checkmark |
| Digital map (like patwaris map) | Description: Close | Description: Close | Description: Close | Description: Checkmark |
| Record privacy | Description: Close | Description: Close | Description: Close | Description: Checkmark |

## Summary

In this section, we will summaries the chapter as some similar systems that were designed earlier for a similar purpose but they did not achieve significant success due to many reasons which we have discussed. In this chapter, it is discussed, that various systems have been developed which help users. There are existing systems such as “Punjab Land Records Authority”, “Zameen.com” available with different functionalities. These existing systems have a single point of failure and temperable database, which causes many conflicts in society.

The most important aspect of any land management system is that it should be immutable and transparent at any cost because without that the purpose to build the land management system dies. In our proposed system transparency and immutability have the highest preference so that can help the society to get better.

# CHAPTER 3

# REQUIREMENTS SPECIFICATION

# Introduction

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems [5]. The installation problems may cause problems in installing the device or application, whereas the performance problems may cause a product to malfunction or perform below expectations or even to hang or crash. In this chapter, the system requirements of our system have been discussed. System requirements of our system include interface requirements, hardware requirements, software requirements, database requirements, functional requirements, and non-functional requirements. The feasibility of our project has also been discussed in this chapter.

## Interface Requirements

Interface requirements state the mandatory things that we need to have to interface the different components of the system with themselves to make these communicate easily and compile the whole system [5]. In this section, we also need to know what we need to have to get the system communicating with the other environment. Since the project contains hardware and software both and including an online interface as well as a mobile application platform. Therefore, the section should be divided into the respective categories

### Hardware Interface Requirements

These requirements include the minimum processor speed, memory, and disk space required to install Windows. In almost all cases, you will want to make sure that your hardware exceeds these requirements to provide adequate performance for the services and applications running on the server [5].Our system is a web-based application it can easily be used on the computer with a dual-core 1.7GHz intel core i3 processor, 500 GB hard disk and 4 GB RAM.

### Software Interface Requirements

As our system is a web-based application so the user needs a browser running on its computer with a strong connection of the internet.

## Functional Requirements

Those requirements which are explicitly stated during the phase of requirement gathering are called functional requirements of the system. These are the requirements that are a must for the user. The following are the functional requirements of the system.

### Land Registration

1. The system should get the CINC number of the user.

2. The system should get proof of ownership of the user.

3. The system should record the entered data with the time stamp.

4. The data should be distributed to all peers.

### Land Verification

1. A user should be able to search his land record by entering CNIC.

2. The system should display the searched record to verify.

3. The system should not show record to anybody other than the user.

4. The system should only allow another person to view record if the user gives his

The system generated an ID.

### Buy the Land

1. The user should be able to enter the required information about the land he/she wants to

buy.

2. The system should be able to display the available lands according to requirements

entered by the user.

3. The user should be able to get the contact information of the landowner.

### Sell the Land

1. The user should be able to post his land for sale.

2. The user should be able to enter all the required information about his land.

3. The system should be able to show his land to untrusted users.

### Digital Map

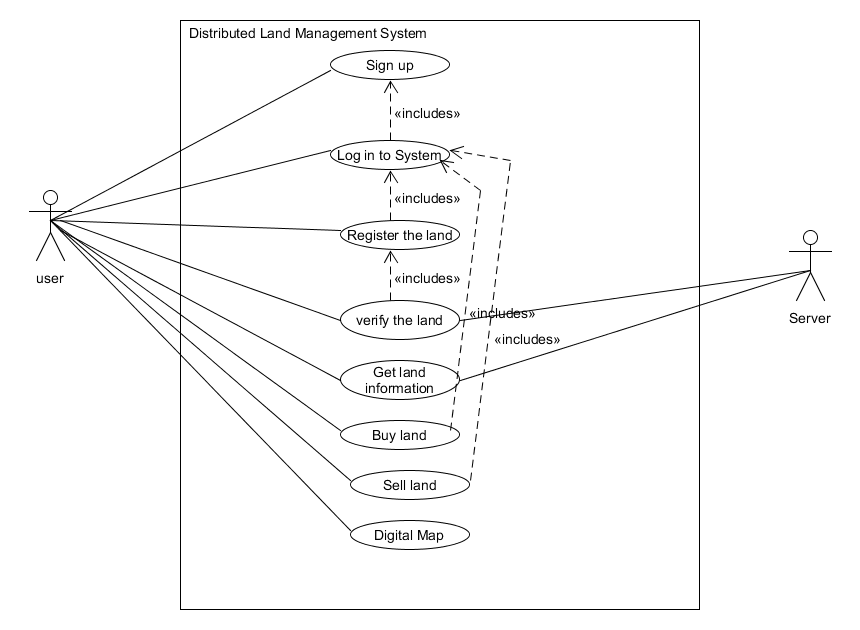
1. The user should be able to enter the information on his land.

2. The system should be able to locate his land on the basis of the entered information.

## **Use Case Diagram**

The use case diagram shows the interaction between the user and the system to complete the required task. The user can be a human or and outside the system as well. The actors, usually users of the system are defined according to their roles. It also shows the relationship between and among the actors and the use cases as well. There are many functions for the user in the System. The below figure 3.1 shows the use case diagram of "Distributed Land Management System over blockchain”. The System first provides an interface that is Sign up and log in. This provides two options so that the user can sign up if he is not registered to the system otherwise, he can log in to the system. After a successful login to the system, it takes the user to the home page. On the home page, users have many options. He/she can register the land by uploading an authorized document of the land. He can verify his registered land by entering a system id which was generated at the time of land registration. He can do the trade of the land; he can buy and sell the land. He can view his registered land by using a digital map.

The table specifies use case scenarios about overall operations which are included in the system. System actors are a user who are using and the System. Pre-condition of the particular use case is that the system should be in running condition and all the components are working properly which are shown in the figure below:

Figure 3. 1 Use case diagram of the distributed land management system

The use case of the Land Registration is shown in figure 3.2. The use case begins as the user uploads the authorized document of land and enters the district and tehsil name. And in response, the system shows the successful registration’s message.

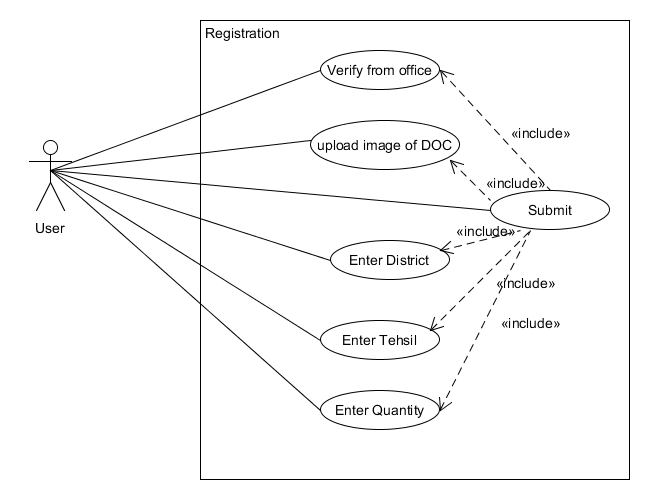


Figure 3. 2 Use case diagram of register the land

Table 3. 1Register the land-use case

|  |  |
| --- | --- |
| Use case Id | UC-1 |
| Use Case Name | Register the land |
| References Requirement | Requirement No 1 |
| Actors | The user of the System |
| Purpose | To register the land |
| Overview | The user first uploads the authorize document of the land. Then he enters the name of district and tehsil |
| Type | Primary and essential |
| Pre-condition | User must be logged in |
| Post-condition | The System registers the land successfully |
| Typical Course of event | |
| Actors Action | System Response |
| 1. Use Case begins as the user uploads the authorized document and enters the name of district and tehsil. | 1. The system responds by checking all input fields. And shows the message of successful registration. |
| Alternative flows | |
| 1a. The user left any input field empty. | 1b. The system does not forward the user in the next phase. |

### Verify the Land

Figure 3.3 shows the Verify the Land-use case. The use case begins as the user enters the System id which was generated at the time of registration of the land. And in response, the system takes the user to validate the information page.

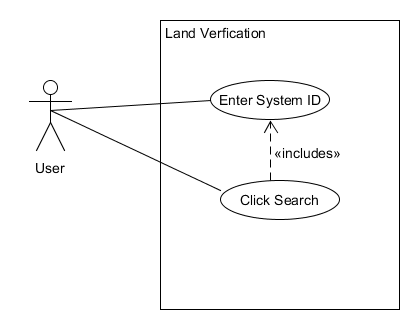


Figure 3. 3 Use case diagram of verifying the land

Table 3. 2Verify the land-use case

|  |  |
| --- | --- |
| Use case Id | UC-2 |
| Use Case Name | Verify the Land. |
| References Requirement | Requirement No 2 |
| Actors | A user of the System. |
| Purpose | To Verify the registered land. |
| Overview | The user enters the system id which was generated at the time of registration of the land. in response the system forwards the user to choose to verify or update the information pop up window. |
| Type | Primary and essential. |
| Pre-condition | Users should be able to verify the Land. |
| Post-condition | The land is successfully verified. |
| Typical Course of event | |
| Actors Action | System Response |
| 1. The user enters the system id which was generated at the time of registration of the land. | 1. The system shows the pop-up window to verify or update the information on the land. |
| Alternative flows | |
| 2a. The user does not enter the id. | 2b. The system does not forward the user to the next phase and shows the message of missing inputs. |

### Buy the Land

Figure 3.4 shows the Buy the Land Use Case diagram. The use case begins as the user enters the type of property which he wants to buy. The system responds by showing available properties.

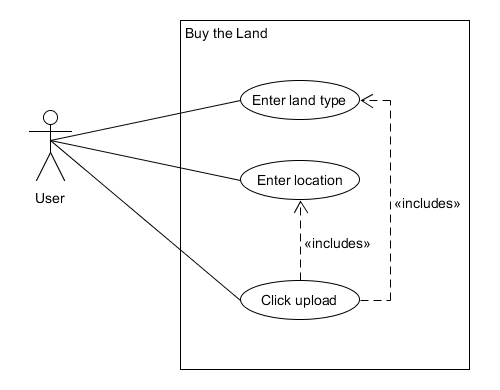


Figure 3. 4 Use case diagram of buy the land

Table 3. 3 Buy the land-use case

|  |  |
| --- | --- |
| Use case Id | UC-3 |
| Use Case Name | Buy the Land |
| References Requirement | Requirement No 3 |
| Actors | A user of the system. |
| Purpose | To choose an image. |
| Overview | The user enters the type of property which he wants to buy in response is able then to get output. |
| Type | Primary and essential. |
| Pre-condition | User must be logged in to the system. |
| Post-condition | Users should be able to get available properties. |
| Typical Course of event | |
| Actors Action | System Response |
| 1. Users enter the property type and location. | 1. The system shows the available property in the desired location. |
| Alternative flows | |
| 2a. The user does not enter the Property type or Location. | 2b. The system redirects to the same page and does not take him further. |

### Sell the Land

Figure 3.5 shows the sell the Land Use Case diagram. The use case begins as the user enters the type of property which he wants to sell. The system responds by uploading the land for sale.

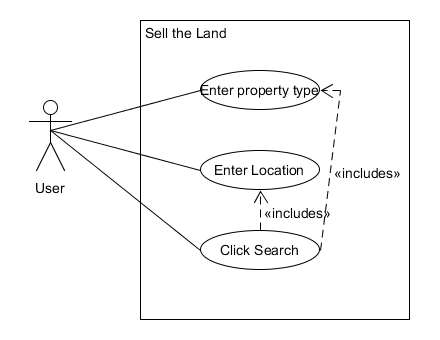


Figure 3. 5 Use case diagram sell the land

Table 3. 4 Sell the land-use case

|  |  |
| --- | --- |
| Use case Id | UC-4 |
| Use Case Name | Sell the Land |
| References Requirement | Requirement No 4 |
| Actors | A user of the system. |
| Purpose | To Upload the land on the system for sale. |
| Overview | The user enters the type of property which he wants to sell in response is able to upload the land on the system for sale. |
| Type | Primary and essential. |
| Pre-condition | User must be logged in to the system. |
| Post-condition | User should be able to upload the land on the system for sale |
| Typical Course of event | |
| Actors Action | System Response |
| 1. Users enter the property type and location. | 1. The system responds by uploading the land for sale. |
| Alternative flows | |
| 2a. The user does not enter the Property type or Location. | 2b. The system redirects to the same page and does not take him further. |

### Use the Digital Map use case

Figure 3.6 shows “Use the Digital Map” Use Case diagram. The use case begins as the user enters the location of the property which he wants to see on the map. The system responds showing his desired land on the map.

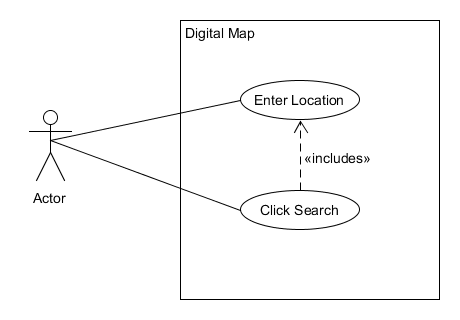


Figure 3. 6 Use case diagram of digital map

Table 3. 5 Digital map use case

|  |  |
| --- | --- |
| Use case Id | UC-5 |
| Use Case Name | Use the Digital Map |
| References Requirement | Requirement No 5 |
| Actors | The user of the system |
| Purpose | To See the desired land on the map |
| Overview | A user enters the location of the land and the system shows the land on the map. |
| Type | Primary and essential. |
| Pre-condition | Users must have registered the land on the system. |
| Post-condition | Users should be able to see the land on the map. |
| Typical Course of event | |
| Actors Action | System Response |
| 1. User enters. Location of land. | 1. The system shows the land on the map. |
| Alternative flows | |
| The system runs the text but is failed in producing output. | |

## Non-Functional Requirements

Those requirements which are not explicitly stated during the phase of requirement elicitation are called non-functional requirements of the system. These are the requirements that are considered foremost when we talk about the quality of the system. These are the requirements that are almost present in all the systems, the quality of our system will improve by keeping these requirements in mind during the phase of development of the system.

### Performance Requirements

It basically is the response of the system in a time when the user interacts with the system. So the system should be quick enough to respond when the user interacts with it.

### Security

The system should be secured and it must not allow unauthorized access, accidental or unintended use of the system and provide access to authorized users only.

### Flexibility

The system should be flexible when it's required to add some new features according to the change in requirements of the user.

### Testability

The system should be able to get tested in order to check the performance of the system.

### Maintainability

The system should be maintainable, and it should be retained in its original form, and to be restored to that form in case of a failure

## Resource Requirements

A list of hardware or software items needed to accomplish a task. The following tools are used in this project.

### Blockchain

The blockchain is an incorruptible digital ledger of transactions that can be programmed to record virtually everything of value. Such as creation and transfer of values. An asset can be anything e.g. currency, property, document, etc.

Blockchain does not have CRUD operations but only CR (create & read). This means only a new transaction can be generated or previous can be read and once a transaction is stored in Blockchain, it can’t be removed or updated to make the ledger corrupt. In short, we can say to perform every action on blockchain query of that specific action is stored in blockchain as a unique hash for record purpose and that query can’t be deleted or updated which means for every change in a value blockchain will be keeping a history of all the changes from the start [6]. The main purpose of the blockchain is to allow fast, secure and transparent transactions.

### Tools

We will be using the following languages, architectures, and tools for developing our system.

* **Html/Sass:** For Web interface.
* **Reactions**: As a client-side JavaScript library supported and founded by the Facebook community.
* **NodeJS:** As a server-side language by JavaScript community. NodeJS has a vast community as is very fast, modern and reliable as compared to PHP, .NET, etc.
* **Hyperledger Fabric**: For writing the smart contract for the whole logic of the application. Hyperledger Fabric is an organization Blockchain founded by Linux foundation and then continued by IBM.
* **IPFS:** For storing data and image files in a decentralized manner. IPFS is being supported by the blockchain community, Ethereum, Mozilla and Linux developers.
* **GO:** Go was founded by Google as a somewhat replacement for C/C++. It’s an easy low-level language as compared to C/C++; Go is used in native Blockchain implementation. And has an SDK for Hyperledger fabric.
* **AWS/EC2:** EC2 is an AWS service, a Linux machine to deploy a variety of applications. We will be using Nginx to deploy our Web App client and server both.
* **Microservices:** Instead of conventional monolithic architecture. We will be using microservices architecture dividing the whole app in separate services. Each service will communicate only where it is required. And the crash of one service will prevent the crash of the whole app.
* **Docker:** Docker is a computer program that performs operating-system-level virtualization. Docker is used to running software packages called containers. We will use Docker for deploying our app where the whole environment is not configured.

## Project Feasibility

Moving to a blockchain-based property registration from the current digital system would affect the whole process. Many people will lose their jobs because the blockchain removes the need for middlemen like patwaris and brokers etc. The Shift to the blockchain-based land registration system is not an overnight process. Now the time has come to upgrade our most lands records system to the latest technology to achieve a high level of transparency and security. This project is the step in the right direction to achieve a high level of security and to lessen the difficulties of the user to register his land’s records [7].

### Technical Feasibility

Technically this project is feasible as the distributed ledger system of blockchain is a more secure method of registration of the land and transferring the rights of the land’s ownership. Land’s trade by using the distributed ledger system of blockchain is more secure and efficient. The main modules of our systems are land registration and land trade which are technically feasible with the distributed ledger system of blockchain.

### Operational Feasibility

Operational feasibility is a measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. Our system is performing all its operations accurately. All functions including land registration, land verification, and land trade are working accurately.

### Legal & Ethical Feasibility

Legal and ethical feasibility is an assessment of whether the project meets the legal requirements that exist for implementation. Our system is legally and ethically feasible as:

1. It does not go against any law of the country.
2. It is purely designed for the assistance of people, and to reduce their difficulties in the management of land’s records.
3. Data of the user is secure and it cannot be accessed by anyone without his consent.

## Summary

In this chapter, the system requirements were discussed. It includes interface requirements, hardware requirements, software requirements, database requirements, functional requirements, and non-functional requirements. In interface requirements, we discussed how the system’s interface interacts or communicate with another interface and we also need to know what we need to have to get the system communicating with the other environment. We briefly discussed the functional and non-functional requirements of the system. All use cases with complete descriptions were also discussed in this chapter. At the end feasibility of the system is discussed that the system is very much feasible within time and budget. The system’s technical, operational, and legal feasibility is positive as it doesn’t operate against and rule and regulation of the country.

# CHAPTER 4

# SYSTEM MODELLING

# Introduction

System modeling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. It is about representing a system using some kind of graphical notation, which is now almost always based on notations in the Unified Modeling Language (UML). The system’s modeling of our system is done in this chapter. 4+1 view model of the architecture of our system is described in this chapter. Which is all about presenting the view of different stakeholders related to the system. The 4 + 1 view model describes software architecture using five concurrent views, each of which addresses a specific set of concerns [8]. All five views with their respective diagrams are discussed in the chapter. Apart from the 4+1 view model of architecture design approaches are also discussed.

## System Design

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

## Design Approach

Generally, there are two basic design approaches in software engineering.

* Top-Down Design Approach
* Bottom-Up Design Approach

### Top-Down Design Approach

We know that a system is consists of the sub-systems and it contains many components. Further, these sub-systems and components may have their onset of sub-systems and components and create hierarchical structure in the system.

The top-down design takes the whole software system as one entity and then decomposes it to achieve more than one subsystem or component based on some characteristics. Each sub-system or component is then treated as a system and decomposed further. This process keeps on running until the lowest level of the system in the top-down hierarchy is achieved.

Top-down design starts with a generalized model of a system and keeps on defining the more specific part of it. When all components are composed the whole system comes into existence [9].The top-down approach of our system is given below in the figure.

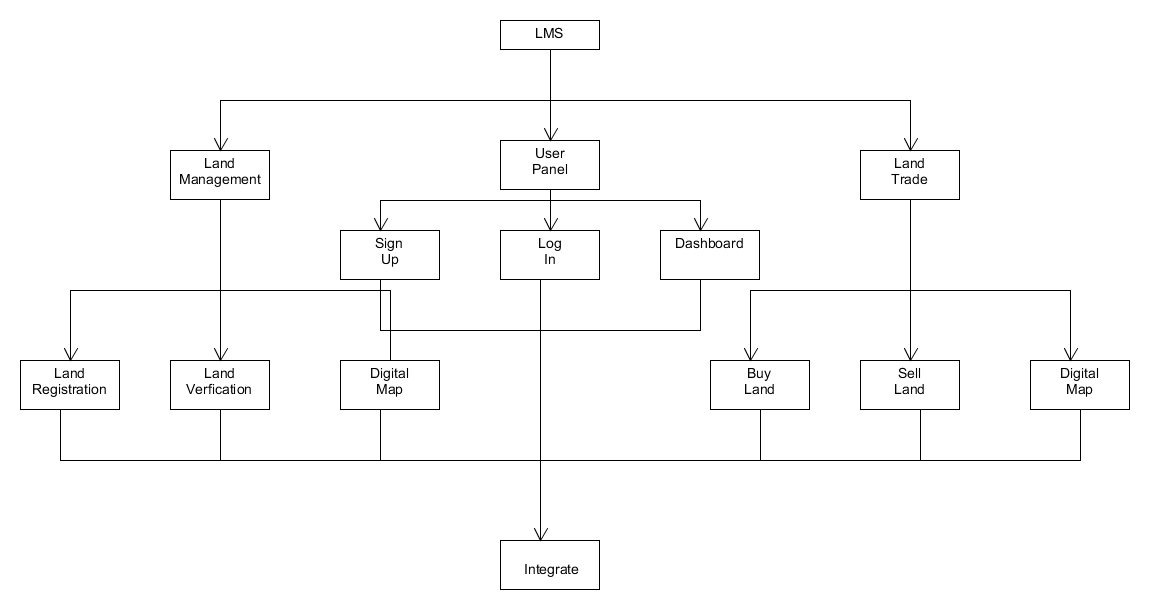


Figure 4. 1 Top-down approach view of the system

## Interface Design

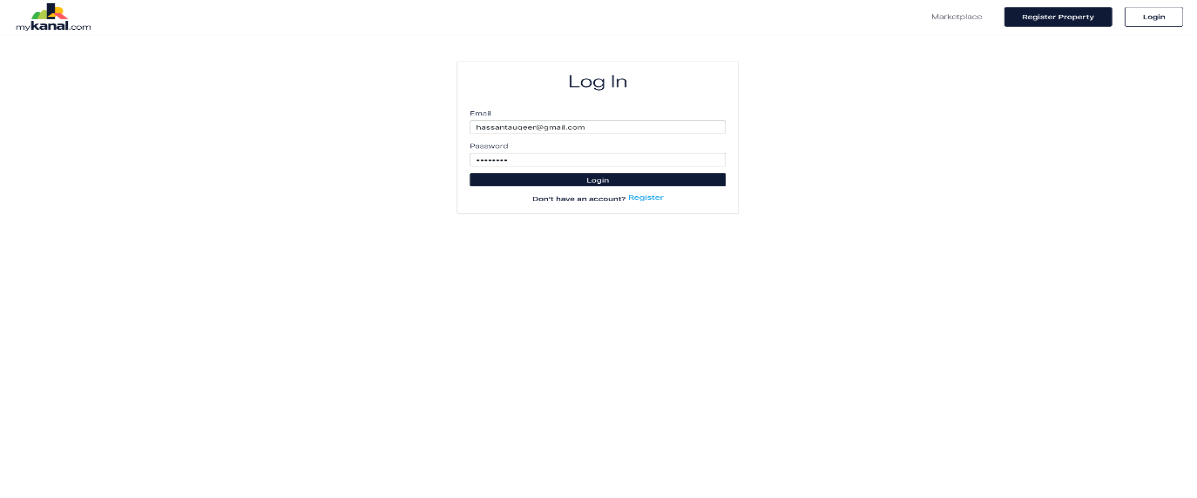
User interface design or UI design generally refers to the visual layout of the elements that a user might interact with a website or technological product. This could be the control buttons of a radio or the visual layout of a webpage. User interface designs must not only be attractive to potential users but must also be functional and created with users in mind.

### High Fidelity Prototype

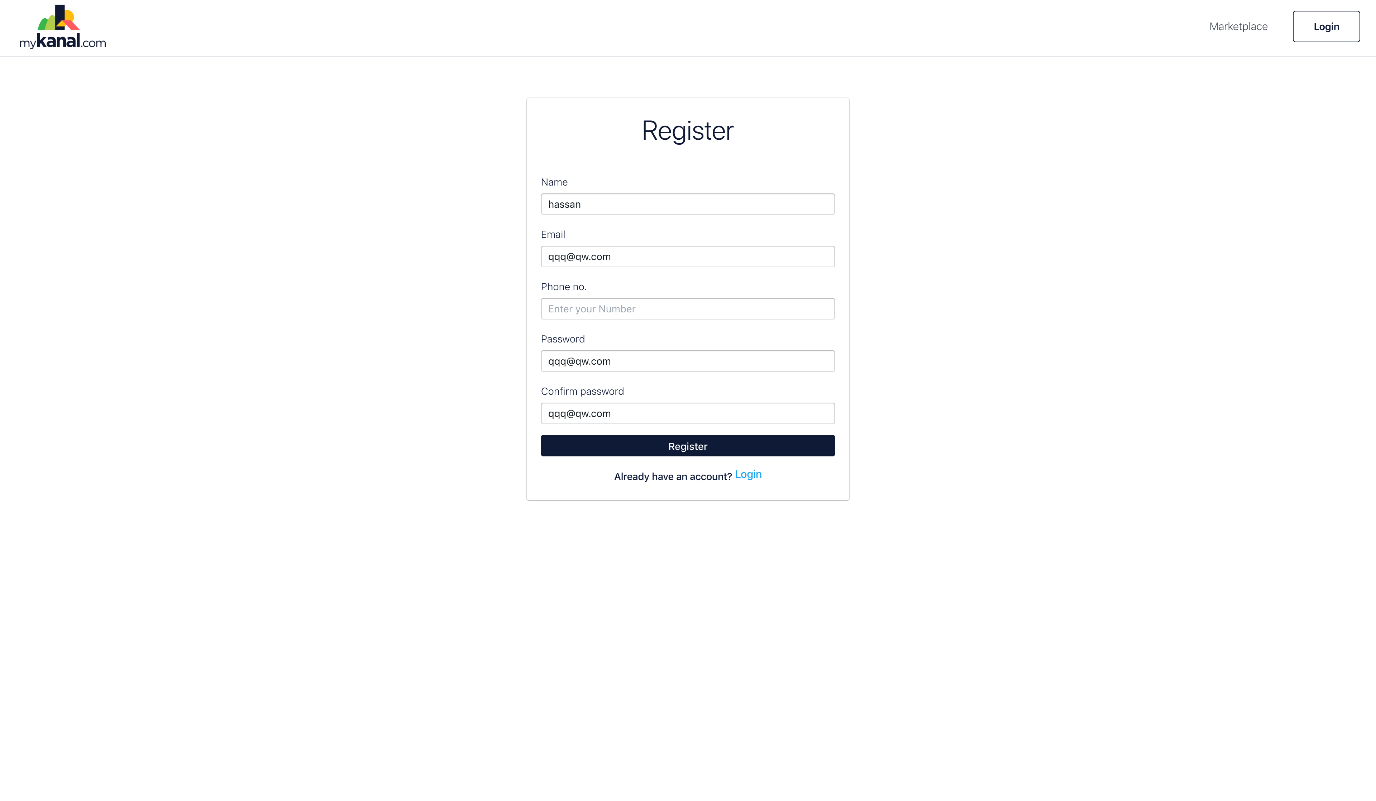
**High fidelity prototypes** are highly functional and interactive. They are very close to the final product, with most of the necessary design assets and components developed and integrated. Hi-fi prototypes are often used in the later stages to test usability and identify issues in the

workflow.

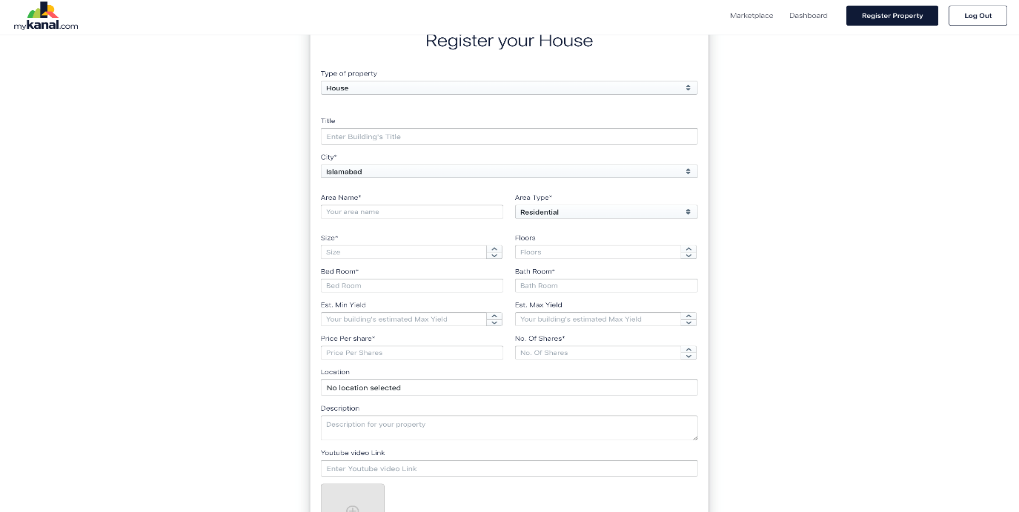
#### Log in



#### User registration



#### **Land registration**



#### Dashboard

## 4+1 View Model of Architecture

In November 1995, Philippe Kruchten published a paper entitled: "Architectural Blueprints The “4+1” View Model of Software Architecture". The intent was to come up with a mechanism to separate the different aspects of a software system into different views of the system. Why? Because different stakeholders always have different interests in a software system. Some aspects of a system are relevant to the Developers; others are relevant to System administrators. Developers want to know about things like classes; System administrators want to know about deployment, hardware, and network configurations and don't care about classes. Similar points can be made for Testers, Project Managers and Customers. It was thought that it made sense to decompose architecture into distinct views so stakeholders could get what they wanted. In total there were 5 views in his approach but he decided to call it 4 + 1 view [8]. We will discuss each view of our system with its diagrams.

### Logical View

This contains information about the various parts of the system. In UML the logical view is modeled using Class, Object, State machine and Interaction diagrams (e.g Sequence diagrams). Its relevance is really to developers.

#### Class Diagram

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram  
that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The class diagram of our system is shown below.

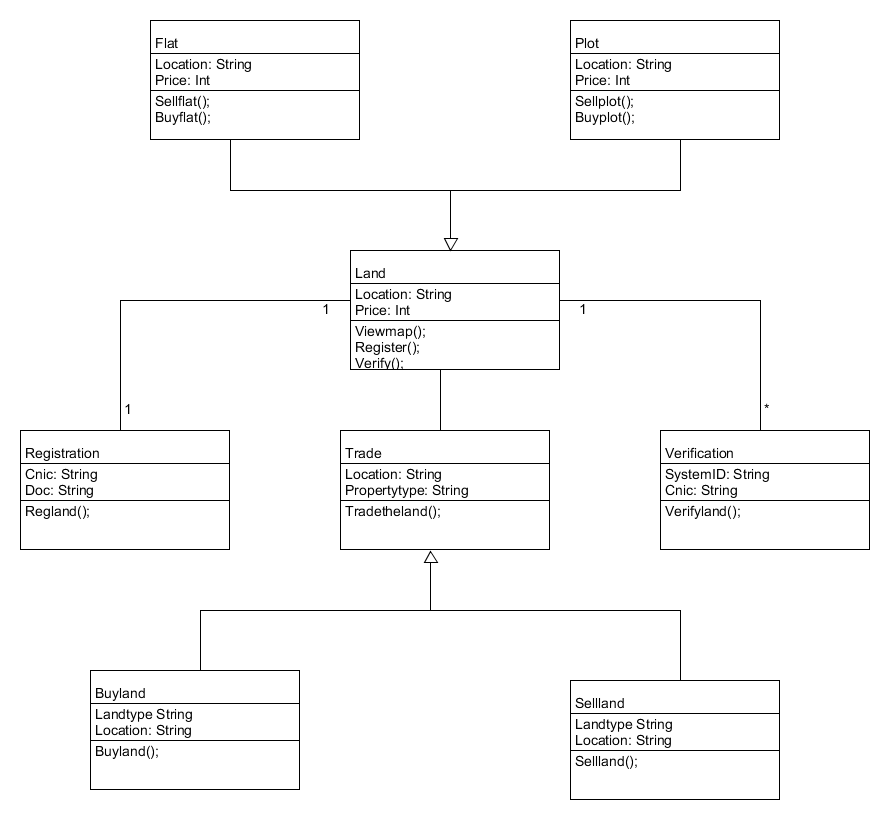


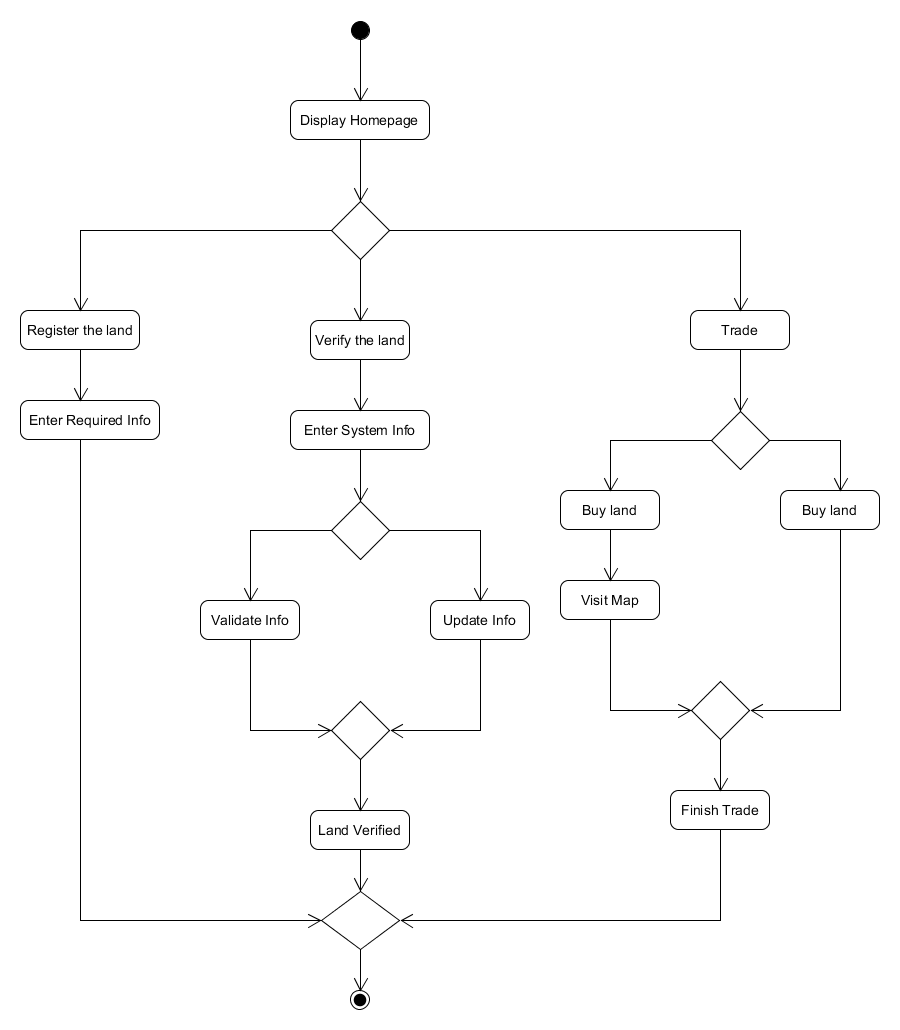
Figure 4. 2 Class diagram of a system

## Process View

This describes the concurrent processes within the system. It encompasses some non-functional requirements such as performance and availability.

### Activity Diagram

The activity diagram is an important behavioral diagram in the UML diagram to describe the dynamic aspects of the system. The activity diagram is essentially an advanced version of the flow chart that modeling the flow from one activity to another activity. The overall flow of the system is described thoroughly.

Figure 4. 3 Activity diagram of a system

### Sequence Diagram

A diagram that shows the steps of interaction between the user, as well as the system to accomplish the objective, is called a sequence diagram. A sequence diagram is also known as an interaction diagram. It describes the interaction between and among processes and user as well in sequential order. A sequence diagram shows the interaction of an object arranged in a time sequence. The most important part of any project is the process between the user and the system. The sequence diagram helps with this process by showing the interaction between user and system in a well-structured manner. An actor represents a type of role where it interacts with the system and its objects. In a sequence diagram, the interaction’s arrangement is organized in the sequence of time. The objects collaborate in a time sequence. There are many scenarios for the user in our system "Distributed Land management system over Blockchain", where the user interacts with the system. The sequence diagram is an important part of the system modeling chapter.

#### Register the Land

In our System, one of the activities for the user is to register the land. The first user clicks one the registration page. Then systems respond by acquiring the information about the land. A user enters the information and the system responds that land is registered. Figure 3.10 shows the sequence diagram.

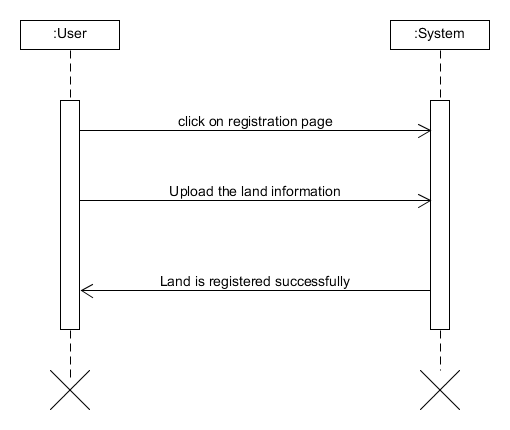


Figure 4. 4 Sequence diagram of register the land

#### Verify the Land

After the Registration of the land user can verify his registered land. This activity is dependent on the previous activity which “registers the land”. In this activity, the user enters systems id which was generated at the time of registration. The system responds by asking the user to verify the given information. Then the user enters “confirm” if the information is correct, he/she enters “Update” if some information is missing.

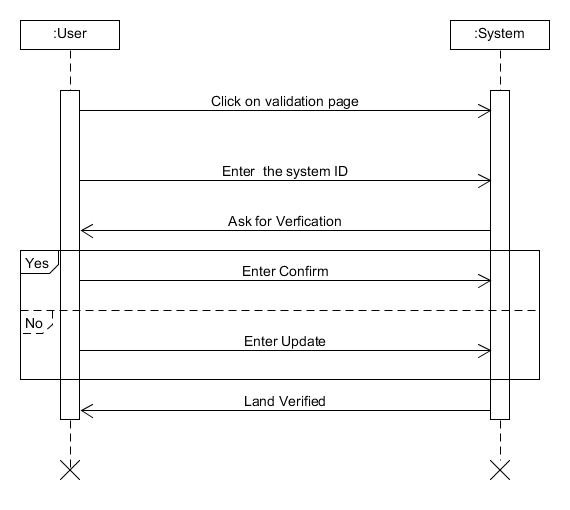


Figure 4. 5 Sequence diagram of verifying the land

#### Buy the Land

The next sequence diagram of our application is to buy the Land. First, the user will enter the type of property he wants to buy. Once the type of property type is selected, the user will enter the location of the property. Then the system will respond with the desired output.

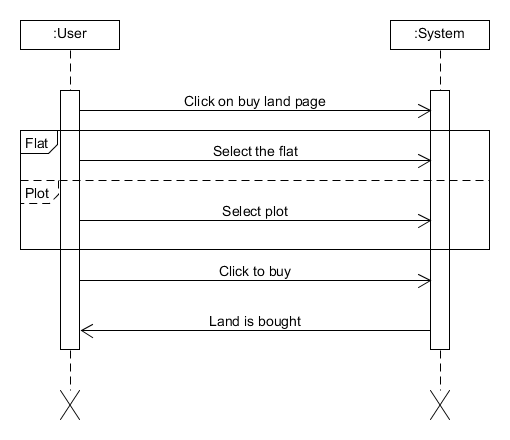


Figure 4. 6 Sequence diagram of buy the land

#### Sell the Land

The next sequence diagram of our application is to buy the Sell. First, the user will enter the type of property he wants to buy. Once the type of property type is selected, the user will enter the location of the property. Then the system will respond with Message “Land is uploaded successfully for the sale”. Shown in figure 3.13

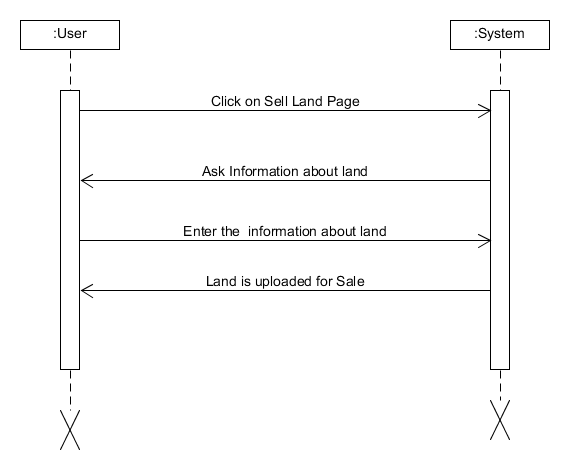


Figure 4. 7 Sequence diagram of sell the land

#### Use the Digital Map

The user simply enters the location of his registered land. Then the system responds by showing the land on the map. This is shown in the figure given below.

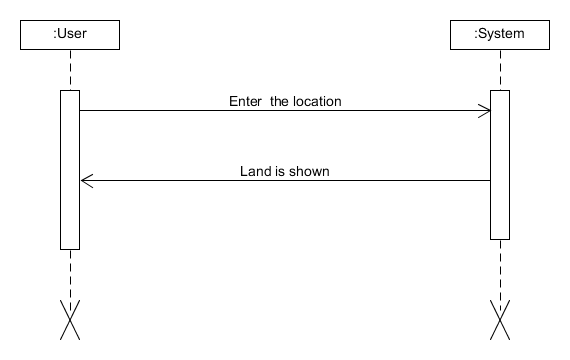


Figure 4. 8 Sequence diagram of digital map

## Development View

The development view focusses on software modules and subsystems. In UML, Package and Component diagrams are used to model the development view.

### Component Diagrams

Component diagrams are utilized in modeling the physical parts of Object-Oriented systems that are utilized for picturing, indicating, and documenting component-based Systems and furthermore for developing executable systems through forward and reverse engineering. Four main components are linked together. The first one is the Sign-Up, it linked to the Registration interface and Trade interface. The registration interface is linked to the verification.

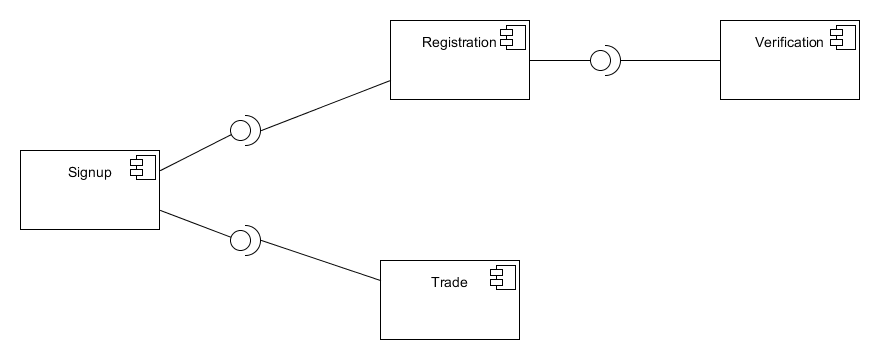


Figure 4. 10 Component diagram of the system

## Physical View

The physical view describes the physical deployment of the system. For example, how many nodes are used and what is deployed on what node. Thus, the physical view concerns some non-functional requirements such as scalability and availability. In UML, Deployment diagrams are used to model the physical view.

### Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. The deployment diagram of our system is given below in the figure.

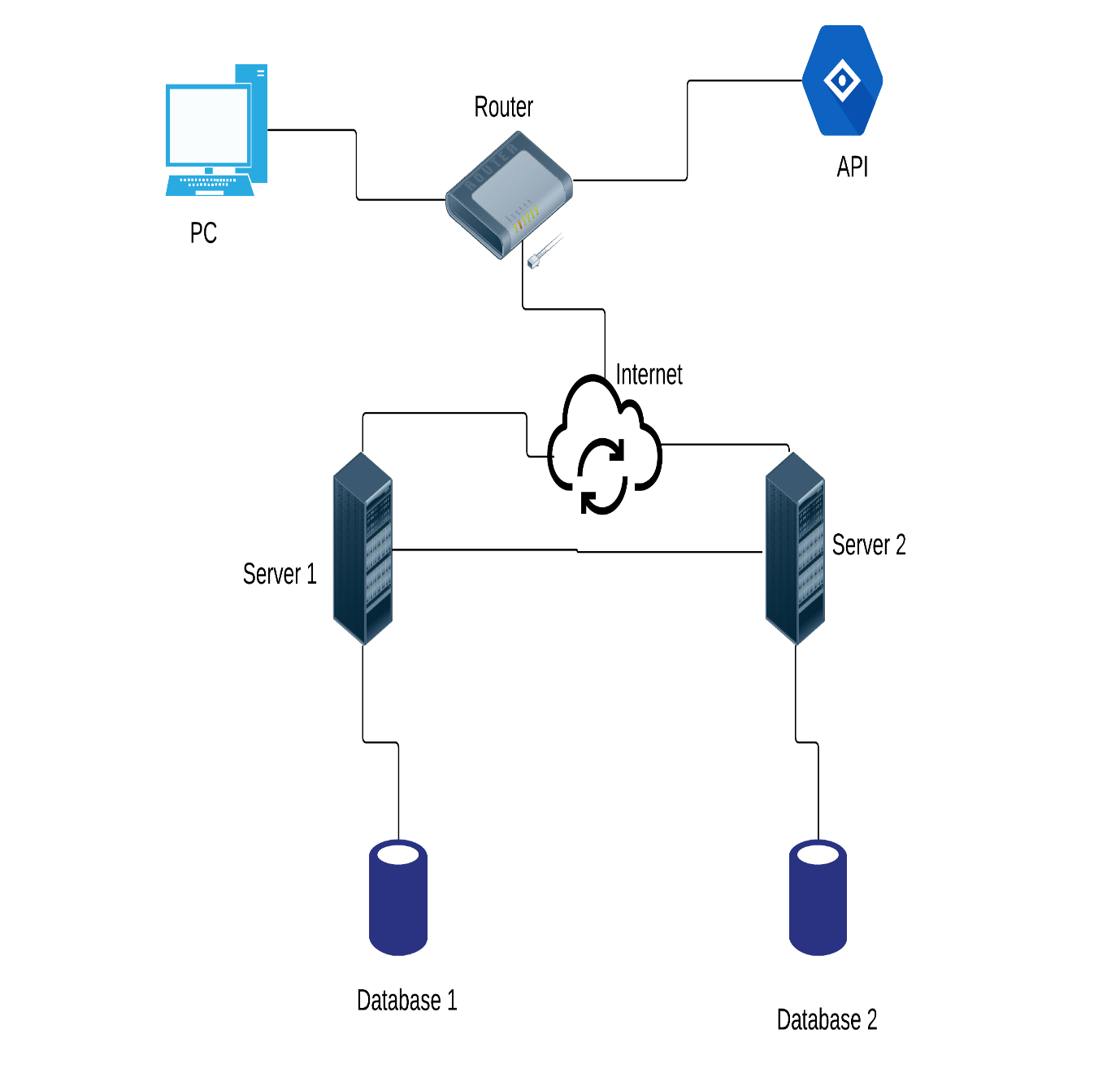
****

Figure 4. 9 Deployment diagram of a system

## The use case view

The use case view describes the functionality of the system from the perspective of the outside world. It contains diagrams describing what the system is supposed to do from a black-box perspective. This view typically contains Use Case diagrams. All other views use this view to guide them.

All use cases with their diagrams have been discussed in chapter 3.

## Summary

In this chapter, we have discussed, the modeling of our complete System that is the complete flow of our System from start to end. Here in this chapter, we have defined use case diagrams with their use cases and we have also discussed the sequence diagram and some explanation. As use cases are used to identify the actors and different use cases like in our System we have “Register The Land”, “Verify The Land”, “Buy the Land” so on and in the primary side of the use case we define the actor that is “User” and secondary side of the use case we define the “Systems” involved which in our Web Application. The Sequence diagram shows object interaction arranged in time sequence and the processes between the user and the system which is one of the most important parts of any project.

**CHAPTER 5**

**IMPLEMENTATION**

# Introduction

This chapter discusses the implementation of the system. Module by module implementation of the system is explained in this chapter. How each and every module of the system was developed and which library, framework, API or service was used. Detail of implementation of every module of the system is discussed. The working of the particular library, framework, API or service that was used in the module is also explained in detail.

## **Modules Our System**

Modules of our system are given below with their implementation details.

### Land Registration

The module land registration will allow the user of the system to register his land on the system. The user will enter his valid CNIC number and proof of ownership of the land to register it on the system.

The user can register his land on our system by following the given steps of the registration process:

1. Documents of land are verified, and its record is verified from the patwari office.
2. After a successful verification record of land is put on our system against a unique ID (signature) of the owner of Land.
3. Further records of that land can only be changed through that unique ID (signature) that’ll be only with the Owner of land.
4. The record of any user can be verified by a government signature that will be held by some single government authority.

### Land Verification

The module land verification of the system will allow users to verify his registered land. The user can verify by entering the specific id of land which was generated by the system at the time of registration and the id of the owner of the land only if the owner has allowed a user to view his land. The user can only read the information on land.

### Land Trade

One of the most important modules of our system island trade. This module provides a platform to trade land. So users can use this platform to have authentic lands and fast land trade instead of the local and conventional physical trade system.

### Digital Map

This Module of the system will allow users to check the location of the land. The user will enter the information about the land and the map will locate the desired land. Our map will replace the register and map patwaris have, with specific terms Khasra number, Kheevat, khatooni, etc

## Tools and Technology

The following are tools, libraries, framework, API and services that were used in developing the modules of the system.

### React.js

React is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications, as it is optimal for fetching rapidly changing data that needs to be recorded[reference]. We used to react to developing the front ends of modules of our system.

### **Blueprint.js**

Blueprint is a React UI toolkit for the web. It is optimized for building complex, data-dense web interfaces for desktop and web-based applications. A blueprint is a tool in the javascript UI libraries category of a tech stack. Blueprint is a newly open-sourced design system implemented as a collection of composable react components and optimized for desktop and web-based applications.

### Node.js

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser. Node.js lets developers use JavaScript to write command-line tools and for server-side scripting running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js is unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

### Hyperledger Fabric

Hyperledger Fabric is a modular blockchain framework that acts as a foundation for developing blockchain-based products, solutions with modular architecture and applications using plug-and-play components that are aimed for use within private enterprises. It offers a unique approach to the consensus that enables performance at scale while preserving privacy.

### **IPFS**

For storing data and image files in a decentralized manner. IPFS is being supported by the blockchain community, ethereum, Mozilla and Linux developers.

### GO

Go was founded by Google as a somewhat replacement for C/C++. It’s an easy low-level language as compared to C/C++, Go is used in native blockchain implementation. And has an SDK for hyperledger fabric.

### AWS/EC2

EC2 is an AWS service, a Linux machine to deploy a variety of applications. We’ll be using Nginx to deploy our Web App client and server both.

### Micro services

Instead of conventional monolithic architecture. We’ll be using micro services architecture dividing the whole app in separate services. Each service will communicate only where it is required. And the crash of one service will prevent the crash of the whole app.

### Docker

Docker is a computer program that performs operating-system-level virtualization. Docker is used to running software packages called containers. We’ll use Docker for deploying our app where the whole environment is not configured.

### Google geolocation API

The Geolocation API allows the user to provide their location to web applications if they so desire. For privacy reasons, the user is asked for permission to report location information. We used geolocation API in our digital map’s module so that the user can use the google map to view his land on the map.

## Summary

In this chapter, we have discussed the module implementation of the system. Tools that were used in developing were discussed briefly. We have used react.js and blueprint.js for the developing of the front ends of modules of our system. Node.js was used in the backend development of the module of the system. A geolocation API was also used to API allow the user to provide their location to the system and use google map. Hyperledger Fabric acts as a foundation for developing blockchain-based products, solutions with modular architecture and applications using plug-and-play components that are aimed for use within private enterprises. Details discussion about the implementation of the system is done in this chapter.

# CHAPTER 6

# TESTING, ANALYSIS, AND VALIDATION

# Introduction

Testing is done in this chapter, which is an activity or process of finding errors and then removing those errors so that the system contains no defects and performs its functions correctly. It means that by making use of test cases the functionalities of the system are checked which are according to the requirement of the user. Testing is the process or activity of executing a software system or application in an environment of real-world and detects and removes bugs and errors in the application. In order to complete this part of the report, many techniques of testing have been performed to achieve the goal which is defects free software system. This testing technique includes system testing, integration testing, white box testing, and black-box testing. Unit screening or unit testing is to be performed to examine or to test the particular part of the software system according to their capability. Integration testing or assimilation testing is to be performed when all parts or some of the parts in a software system are incorporated or integrated. To examine or to treat the overall functionality of the system black box testing is to be done. to evaluate the compliance of the software system or application with the corresponding requirements system testing is to be done. Black box testing nothing to do with the checking of specific pieces of code. It is just dealing with the input and output of the software system that has no idea of internal parts or functions. By considering all these it is very much clear that testing is a very important part or step to determine that the software system or application executes it’s all functions and working accurately and accordingly. By making use of testing and with the help of testing, the testing team could make sure that whether the requirements or demands of the user are being addressed or not. These test cases or test situations are examined by giving input to every component of the software system. All tests or examinations have been executed on an android emulator as well as on mobile phones with the Android operating system. Testing is an essential, the most required or primary element to make sure that all the functionalities and the performances of the system are functioning accurately and to determine that whether the needs of the user are met or not. Errors and bugs of the software system are determined and are also removed with the test cases. At first bugs or failures of the software systems are discovered, after discovering the failure and bugs, these bugs and failures are removed by making changes in the code. All the bugs are fixed by the development.

## Test Bed

A test bed or examination bed is a platform for conducting the testing of the systems. In the test bed or examination, the bed includes hardware and software application is required for testing. All functionalities and performances are tested by providing different kinds of inputs. The test bed used for “Land Management and Information System over Blockchain” is a mobile phone with running Android operating system 4.4 or the latest version with 512 MB Ram as well as 2GB memory.

## System Test Case

The system test instance is executed to make sure the system is performing its functionalities appropriately, accurately and according to the demand of the user. Testing is very important and essential to make sure that all the functionalities of the system are working according to the needs of the user. Throughout the process of the testing of the system many failures and errors also occur. These bugs and errors are removed by making changes in the code yet some of the errors were difficult to settle.

## Test Cases

A test caseis the combination of conditions or variables which is used by the tester to determine whether a software system under testing fulfills the requirements or works accurately. The process of test case developmentcan also help to find problems in the requirements or design of a software system. The efficiency of the structure is validated by the test cases. It is executed according to standard arrangements. After getting the outcomes of the testing, the tester is the one who decides that a specific test is pass or fail. All the test cases consist of the following components which are test case ID, test case reference, QA test engineer, name of personnel, test date, revision history, objective, testing environment, assumptions, preconditions, steps, and status of a test case. Test cases of all features of the system are given below in detail.

### Registration of the Land

The registration of the land is a very important feature of the system. The user of the system registers his land on the system by using this feature. This use case starts when the user enters the specific data of his land.

Table 6. 1 Registration of the land test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Use Case Reference** | **QA Test Engineer** | | **Name of Personnel** |
| TC-1 | UC-1 | Tester | | Inayat Ullah |
| **Test Date** | 30-11-2019 | | | |
| **Revision History** | None | | | |
| **Objective** | To register the data of the land. | | | |
| **Environment** | User Mode | | | |
| **Assumptions** | User is present at registration page | | | |
| **Pre-Requisite** | System is in running state. | | | |
| **Steps #** | **Execution Description** | | **Procedure Result** | |
| 1.  2. | Display registration page  Input the data of land to register it on system | | Page displayed  The user has input the data. | |
| **Comments** | | | | |
| **Status** | Pass  Fail Not Executed | | | |

### Verification of the land

A user here can verify his land which is registered on the system. The user enters the specific ID of his/her registered land which was generated at the time of registration of the land. The system verifies either the specific land belongs to the user or not. Verification of the land test case is shown in table 5.2

Table 6. 2 Verification of the land

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Use Case Reference** | **QA Test Engineer** | | **Name of Personnel** |
| TC-2 | UC-2 | Developer | | Tauqeer Hassan |
| **Test Date** | 30-11-2019 | | | |
| **Revision History** | None | | | |
| **Objective** | To verify the registered land | | | |
| **Environment** | User Mode | | | |
| **Assumptions** | User is currently present at verification page | | | |
| **Pre-Requisite** | The land is previously registered on the system | | | |
| **Steps #** | **Execution Description** | | **Procedure Result** | |
| 1.  2. | Display Verification of the land page  Input the specific ID of the registered land | | Displayed has been page  User input the ID. | |
| **Comments** | | | | |
| **Status** | Pass  Fail Not Executed | | | |

### Buy the land

In this test case, it is described that the user can buy the land. The user goes on-page of land trade which shows him the lands which are posted for the trade. User can select his required land in order to purchase that land. Buy the land test case is shown in table 5.3

Table 6. 3 Buy the land test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Use Case Reference** | **QA Test Engineer** | | **Name of Personnel** |
| TC-4 | UC-3 | Developer | | Arslan Hayat |
| **Test Date** | 30-11-2019 | | | |
| **Revision History** | None | | | |
| **Objective** | To buy the land | | | |
| **Environment** | User Mode | | | |
| **Assumptions** | The user is present on the page of land trade. | | | |
| **Pre-Requisite** | Users must be registered on the system. | | | |
| **Steps #** | **Execution Description** | | **Procedure Result** | |
| 1.  2.  3. | Display land trade page  Click to buy the specific land.  The user then has to set the way of payment to finish the trade | | Displayed page  The user clicked the specific land  User has set way of payment | |
| **Comments** | | | | |
| **Status** | Pass  Fail  Not Executed | | | |

### Sell the land

In this test case, it is discussed that the user can sell the land. A user goes on-page of land trade and selects sell the land and post his land thereby providing information about the land. Sell the land test case is shown in table 3.10

Table 6. 4 Sell the land test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Use Case Reference** | **QA Test Engineer** | | **Name of Personnel** |
| TC-4 | UC-4 | Developer | | Arslan Hayat |
| **Test Date** | 30-11-2019 | | | |
| **Revision History** | None | | | |
| **Objective** | To sell the land | | | |
| **Environment** | User Mode | | | |
| **Assumptions** | User is present at land trade page | | | |
| **Pre-Requisite** | Users must be registered on the system. | | | |
| **Steps #** | **Execution Description** | | **Procedure Result** | |
| 1.  2. | User clicks on the sell land page  A user enters the data of the land which he wants to sell | | Page displayed  Data has been entered and the land is posted on the system | |
| **Comments** | | | | |
| **Status** | Pass  Fail  Not Executed | | | |

## 

### Digital Map

In this test case, it is discussed that the user can view his land by using a digital map. The user goes on-page of a digital map and enters the location of his land and he is able to see his land on the digital map. Digital map test case is shown in the table

Table 6. 5 Digital Map Test Case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Use Case Reference** | **QA Test Engineer** | | **Name of Personnel** |
| TC-4 | UC-4 | Tester | | Inayat Ullah |
| **Test Date** | 30-11-2019 | | | |
| **Revision History** | None | | | |
| **Objective** | To see the land on a digital map | | | |
| **Environment** | User Mode | | | |
| **Assumptions** | User is present at digital map page | | | |
| **Pre-Requisite** | User must be registered on the system and his land too | | | |
| **Steps #** | **Execution Description** | | **Procedure Result** | |
| 1. | The user enters the location of his registered land. | | Land is displayed | |
| **Comments** | | | | |
| **Status** | Pass  Fail  Not Executed | | | |

## Summary

In this chapter, we did the testing, analysis, and validation of our software system. First of all, we have done testing; it is an activity or process of finding errors and then removing those errors so that the system contains no defects and performs its functions correctly. In order to have this part done different kinds of testing or screening have been done for achieving this goal that includes system screening or system testing, integration testing, black-box testing, white box testing, and system testing. To evaluate or to test the specific parts of the application according to their capability then unit screening or testing is to be done. When all parts or some of the parts in an application are incorporated or integrated then integration or assimilation testing is to be done.

# CHAPTER 7

# CONCLUSION AND FUTURE WORK

# Overview

This chapter is all about the conclusion and future recommendations of our system. That how much functions and operations it is performing and how many functionalities can be added to it in the future. How many milestones achieved in the developing are discussed briefly? Also, the limitations experienced in this project are discussed. These limitations need to be handled in the future when the system will be upgraded.

## System overview

“Distributed Land Management System Over Blockchain” is a web-based application for the management of land’s records. The Application allows the user to register his land’s record and verify his record through a system generated id. This system also provides the selling and purchasing of the land. The user can get his desired plot (a piece of land) by putting the information about the plot he wants to purchase. Similarly, the user can put information about his plot which he wants to sell. This system also consists of a map where the user can check its land by entering the required information about his land.

## Milestones Achieved

Milestones trace the start and finish point of the project. It identifies the completion of a particular part of the complete work.

### **Land Registration**

The user is able to register his/her land by providing CNIC and ownership proof of land.

### Land Verification

The user can verify by entering the specific ID of land which was generated by the system at the time of registration and the id of the owner of the Land only if the Owner has allowed him/her to view his/her land.

### Land Trade

It has two features which are given below.

#### Buy the Land

This feature of the system will allow users to buy the land. A user will enter the name of a city, location, and the area of land, property type (plot or cultivable land) and his range of price. The system will show all the suitable lands for that search.

#### Sell the Land

This feature of the system will allow users to sell the land. The user needs to register his land first at our platform and then enter required basic information about his lands like the name of a city, location, the area of land, property type (plot or cultivable) and his demand price.

### Digital Map

This feature of the system will allow users to check the location of the land. The user will enter the information about the land and the map will locate the desired land.

## Limitations

Land management system using blockchain is way efficient than existing systems but the following are limitations or challenges that emerged from discussions with public administration officials:

1. The complexity of blockchain technology
2. Power consumption
3. Immutability
4. Unclear liability issue
5. Re-design of process chains within the administration
6. Limited applicability of legal norms

These challenges need to be addressed in the possible design of a blockchain-based land register through appropriate governance measures so that blockchain technology can be promoted and introduced beyond the land registry [10].

## Future Work

In general, blockchain technology could have a possible application in land administration systems. There are some limitations and questions yet to be discussed, but certain processes could surely benefit from the implementation of blockchain technology. Mainly trust in data stored in land administration systems. In case of registering changes in data stored in land administration systems, it could be time-consuming as we need to develop a consensus between the nodes to make a change in registered data.

The possibility of implementing entire land administration systems in the blockchain is still opened for debate. Land administration systems have a large amount of data stored in them, both legal and spatial, and in case of application of permissionless blockchain, main limitations would be that only hash codes of documents could be stored in blockchain, requiring some other way of storing documents, either for example in torrents or interplanetary File System. On the other hand, there is a possibility of using permissioned blockchain, which is for example implemented using Hyperledger technology, which would allow certain documents to be stored in the blockchain.

There is a limited number of peer-reviewed papers on the possible application of blockchain in land administration systems so this field is still in early stages of development and only couple of case studies, so further research in this field is necessary if benefits of blockchain technology could be brought to land administration systems [10].

## Summary

In this chapter, we discussed the conclusion as we are finalizing our project and we discussed our system and its milestones we set at the start of a project and all milestones are achieved which was set. A user is able to register his/her land and then can verify his registered land from our system. Land trading is a very important module of our system, which is working efficiently as a user can buy or sell the land. We discussed those limitations which can be addressed in the future. There will be some future work that needs to be done as time passes with the up-gradation of technology. So, we also discussed the future recommendations of our system. That how much functionality can be added to it in the future?

# References

|  |  |
| --- | --- |
| [1] | I. Bashir, Mastering Blockchain, London: PACKT, 2018. |
| [2] | P. L. R. Authority, "Home Page," Punjab IT Board, 01 March 2011. [Online]. Available: https://www.punjab-zameen.gov.pk. [Accessed 05 January 2019]. |
| [3] | Z. A. Khan, "Home Page," Zameen.com, 01 March 2006. [Online]. Available: www.zameen.com. [Accessed 05 January 2019]. |
| [4] | Z. U. H. Usmani, Introduction To Blockchain With Use Cases, ISLAMABAD: Guftagoo Publishers, 2018. |
| [5] | D. Janssen, "System Requirements," Janalta Interactive Inc, 10 June 2010. [Online]. Available: https://www.techopedia.com/. [Accessed 02 November 2019]. |
| [6] | A. Rosic, "Introduction to Blockchain Technology," Blockgeeks, 15 June 2013. [Online]. Available: https://blockgeeks.com. [Accessed 04 January 2019]. |
| [7] | Tuko, "different-types-feasibility," 01 April 2015. [Online]. Available: https://www.tuko.co.ke/. [Accessed 06 January 2019]. |
| [8] | P. Kruchten, "The 4+1 View Model of Architecture," vol. 1, no. struggled in the representation of blueprint of 4+1 view, p. 16, 1995. |
| [9] | M. M., "software\_design\_strategies," 10 May 2006. [Online]. Available: www.tutorialspoint.com. [Accessed 10 January 2019]. |
| [10] | M. Stefanović, "Blockchain and Land Administration: Possible applications and limitations," vol. 1, p. 9, 2018. |