National University of Computer and Emerging Sciences

Decentralized Record Filing System

FYP Team

Muhammad Ahmad …… 19L-1199
Muhammad Hamza …… 19L-0907
Humza Noor …… 19L-2375

Supervised by

Mr. Saifullah Tanvir

FAST School of Computing

National University of Computer and Emerging Sciences

Lahore, Pakistan

October 2022

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the:

Title: Decentralized Record Filing System

is the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as **Plagiarism**. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: 12/10/2022

Student 1

Name: Muhammad Hamza

Signature: _____

Student 2

Name: Humza Noor

Signature: _____

Student 3

Name: Ahmad Arshad

Signature: _____

Authors' Declaration

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

Abstract

The local record management system still stands incomplete and inefficient as most of it is non-digitized, even though the digitized aspects are in a rather older era of technology. This takes away the credibility and efficiency of the system, which then makes it prone to inadequacy and operational flaws. This project will solve the problem of digitization and bring decentralized technology to the public sector. Through the combination of blockchain and other suitable technologies, this project will define adequate roles for the users as well as the administrators. Educational and commercial sectors shall be the main circles that will be penetrated by this project.

Executive Summary

This project is aligned with the modern needs of decentralization and its related services. There has been a recent deficit of trust in the traditional centralized approach and there is a race toward blockchain tech. The project shall bring about meaningful research and development to the fore, with the latter being the dominant aspect. By developing a record filing system that provides document verification as well as other documentation for land management, this project is aimed to solve problems in this domain.

Storing land records is a hectic task, especially in an agricultural country. The task is done by local land record holders (called 'Patwaris' in Urdu) at the bottom tier and then managed formally by local developmental authorities. The authorities are institutions of local government and regulate the whole process of land records. The system has been in place for decades, which makes it prone to old practices and certain malpractices. There is a need for the security of data and documents, while there is a huge gap between non-digitized data and digitized records.

The project is aimed to solve two problems; speed up the process of digitization and allow better security by using blockchain. The team also believes that the recent controversies around blockchain technology can be slowed down and give a boost to work being done in this new tech. The system shall act as a web app that will contain all the functionalities needed to perform the required actions. The app will provide a safe and secure platform for users and administrators to store, retrieve and verify documents while also making rental agreements.

A major step in this project is the selection of the appropriate blockchain and a thorough analysis of all its possibilities and limitations. Numerous blockchains are available for educational, commercial, and personal use, but the constraints and efficiency vary from one chain to another. The newer generation of blockchain claim to be green and thus, demand lesser computing power, which is a plus point in an environmental context. Blockchain comes with its strengths and weaknesses, some may work faster but have a smaller clientele, and vice versa.

The selection of blockchain is a major subject of the first phase of the project, while the second phase will predominantly be focused on development. The first phase shall comprise all the basic functionalities like the software structure, user interface, and a prototype of the front-end part. The development of the database, backend, and the linkages between all three parts combined with the deployment of the code will make up the second part of the project. The tools to be used shall be the MERN stack and the language of the selected blockchain.

The project is aligned with the public sector and uses a technology that is almost unknown in the sector. This piece of work has the scope to work as a stepping stone towards the adoption of decentralized technology and fuel the research and development in this field.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

The country faces a huge gap between modern technology and its adoption within state institutions. Land records management remains a controversial and hectic endeavor in all aspects. Using modern decentralized techniques, this project is inclined to bridge the gap by digitizing the whole process and ensuring the security of data.

1.1 Purpose of this Document

This document is a detailed overview, analysis, and report of the Decentralized Land Records Management System and will provide a general and technical overview. This piece of writing will analyze various aspects of our project by offering insight into every corner. The document will elicit, elaborate, evaluate and explain the strengths, weaknesses, and hurdles that are bound to be a part of this project. Our team aims to provide a thorough picture of our project by adding details of every aspect.

1.2 Intended Audience

Primarily, this document is aimed at an academic audience composed of mainly educationalists and professors looking to understand and evaluate this project through the appropriate lens. Secondarily, the document will thoroughly put forth the ambitions, motivations, strengths, and constraints of the proposed project. The problem is understood in detail and various points are viewed when the work is still in the early stages. For a general audience, a manual of the project will be released, which shall be a somewhat simplified variant and act as a user's guide for a non-technical audience.

1.3 Definitions, Acronyms, and Abbreviations

Following are the important definitions, acronyms, and abbreviations used in this document.

SDG: Sustainable Development Goal

Encryption: Conversion of a given document or piece of text into a cipher using the public key of the receiver, which can be stored on the decentralized network.

Decryption: Turning the Cipher into the actual text by using the private key of the receiver.

Zero-Knowledge Proof: The process of verification of the sent material by the sender without revealing any of its contents.

IPFS: Acronym for Interplanetary File System. It is a protocol to store files on a decentralized network using the unique address for each file that assists in locating them in a distributed network.

NFT: Acronym for Non-Fungible Token. It is an entity minted on blockchain that can be used to show individual ownership and originality of the content.

Centralization: An architectural mechanism that uses a central or single medium for its operations.

Decentralization: An architecture that allows a distributed approach and involves multiple stakeholders in the procedure

Cryptographic Functions: Functions used to encrypt and decrypt the data using public and private keys

Electronic Signature: A method of verifying a user's impression of a given file or cipher

Hash Function: Functions that generate hash values for the items to be stored on the blockchain

Public/Private Keys: Public Keys allow a user to receive data on a blockchain, while a private key is paired with it to unlock or decrypt the received cipher text

Blockchain Wallet: An online entity for the identification of every user on the decentralized network that is used to keep track of all activities and ownership

Persistency: Persistent storage is a way of storing data on the free space of hard drives of the users across the blockchain

1.4 Conclusion

The system underdevelopment shall be using a blockchain for the backend, combined with a proper front-end interface that will be assigning proper privileges to the users. Defining roles between the entities of the system and then using the system to properly ensure the flow of the data will be a primary concern of the team. The project demands the review of different work done in this field, which the team has done using various research and development projects. Making sure that all previous work is at hand, whether educational or commercial, the project shall be made operable considering all software requirements, functionalities, and constraints. Defining adequate roles and assigning privileges accordingly is a primary goal of this endeavor. Understanding the working of the tech and its components in use, by clearly defining the terms and definitions of all operational terms.

Chapter 2: Project Vision

To understand the possible outcomes of the system and potential successors of this project, it is necessary to understand and clarify the vision behind it. Starting from the domain and identifying the problem, there shall be a complete description of the project's various angles.

2.1 Problem Domain Overview

There is a continuous struggle to digitize huge chunks of data, independent of the data's nature. This takes away the typical risks of mistakes, and manipulation and prevents corruption. Making the whole process efficient and effective, digital entities ensure better practice. On the other hand, digitization is mostly done using centralized storage techniques, but new technology is shifting towards decentralization. With the servers free from huge storage constraints, the security moves one step ahead using blockchain and offers more control to the users over the data.

2.2 Problem Statement

The data of the land records management stored by local authorities is mostly non-digitized, and the central storage is primarily at risk of malpractice, manipulation, human error, and natural disasters.

2.3 Problem Elaboration

The land records are stored by local government institutions and are stored in a non-digitized manner. While the process is medieval in nature, it involves extensive paperwork. The paperwork is inefficient and constantly vulnerable to fire and manipulation by personnel. The system is prone to the mentioned damages on two fronts. Firstly, the data takes up extensive physical space because of the paperwork being manually entered, retrieved, and updated. This adds up to a solid and exponential overhead in accessing, searching, and evaluating documents. Then, the digitized data, though almost negligible, is stored in a centralized system, thus making the database prone to cyber-attacks and other forms of cyber sabotage.

2.4 Goals and Objectives

The objectives and goals of the project are as follows:

- To develop a user-friendly interface for the management of land/property data using the most updated framework
- Digitizing the current analog system of records.
- To make the storage and retrieval of data on the decentralized network efficient by incorporating the MERN stack and blockchain.
- Using the blockchain approach to increase the credibility of data and solidify the process of storage to tackle the single point of failure caused by a centralized storage solution.
- Solving the vulnerability issues of institutional data (especially public sector), by catering to the exposable nature of centralized servers.
- Provide a new mechanism for the storage of institutionalized land data and act as a catalyst towards digitization.
- Digitize the whole process of land record management.

- Ensure verification of documents by involving all related parties.
- Allow users to have their paperwork related to land in a single place.
- Make sure that data is safe with the use of blockchain technology.
- Freeing the system from vulnerabilities by using decentralized storage.

2.5 Project Scope

The project has the potential to be implemented on a local and national level because it solves the problems of maintenance and storage for all the non-digitized records. Secondly, the usage of blockchain technology will pave the way for a smooth transition towards the new decentralized methodologies in other public and private sector domains. Thirdly, digitization will bring an end to the unsafe nature of public records, especially land-related ones, by taking away the burden of physical storage and the constant risk of crimes like vandalism and arson. On a local level, the project can bring the usage of decentralized technology into the light and stand apart as an early work that shall make room for innovations. Within our university, this project will encourage future students to solve problems of the public domain as well as adopt new technologies to find those very solutions.

Our project will mainly focus on the storage and retrieval of digitized land records on decentralized storage backed by blockchain. For this purpose, a front-end web application will be developed using the MERN stack. The web application will provide a user-friendly interface to the intended organization for easy retrieval and filing of records. In the backend, the documents will be uploaded on selected decentralized storage and their hash will be stored in a secure wallet or database, which can be used for querying, verification, and access of data. Our platform will assist the existing land registry systems with automated rental contracts between different parties.

Moreover, a module will be added for buying and selling land using smart contracts. A complete mechanism will be developed to ensure the smooth transference of land titles once terms set by the parties are met. Furthermore, an option shall be added to create rental agreements using smart contracts. To be precise, our project will deal with the decentralized storage of land records, buying and selling real estate using smart contracts, and creating blockchain-based rental agreements.

We believe that this project will significantly and positively impact the industry and academia while solving a big chunk of problems in the realm of public service.

2.6 Sustainable Development Goal (SDG)

With this project, there shall be a long line of future possibilities in the line of the used technology and the area of expertise. The project shall be made available to the general public and will open the gate towards the digitization of public domain records. The records in general, can fall into various categories, possess many different forms, and cater to different user problems. This project shall act as the stepping stone in two manners; firstly, speeding up digitization in the public domain and secondly, improving the quality of the technology adopted for sensitive data storage. The project shall be sustainable when it is available to the public and there is room for further research and development. The project primarily caters to innovation in the field of digitization and allows the public sector to smoothly transition towards the latest technological implementations.

While innovation is at the center of our idea, it secondarily fuels infrastructure development and will become a source of economic growth. As students of Computer science, we believe

that every production step is a step towards improvement no matter how small, and with this project, we can take a huge leap toward tech adoption and transparency in public institutions.

Among the given areas of SDG, our project shall be primarily catering to number sixteen and secondarily playing a role in number nine and eleven.

Figure 1: Sustainable Development Goals
This figure represents all the SDGs that can be the target of our FYP

2.7 Constraints

There is a list of constraints that can be secondary, if not primary hurdles in the development and deployment of this project.

The technical end will endure the following challenges:

- Huge gap in the research of decentralized technology
- The implementation of blockchain technology is not fully functional in many domains.
- Limited tools are available for development.

The social and legal constraints will be:

- Legalization of blockchain technology and its applications
- Government institutes may not be ready for the adoption of digitized assets
- Lack of confidence in decentralized mechanisms due to the ban on cryptocurrency

2.8 Business Opportunity

The project will primarily solve a problem concerning public welfare and government institutes shall be the primary beneficiary of the finished product. There is a huge market in private enterprises that can benefit from this project. Corporate data, client management, and CRM systems can make excellent avenues for this project's implementation. With slight modifications and further research, this project's trajectory can be turned into a meaningful business case.

2.9 Stakeholders Description/ User Characteristics

The people that own certain pieces of land will be the main users along with the administration of the state institutions that regulate the verification process. The users will use the system to have their documents verified by the system administrators, and this process will be done using digital signatures.

2.9.1 Stakeholders Summary

Keeping the hardships and extensive documentation work of records management, the prime stakeholder shall be the common citizen who owns a certain piece of land in the country, while the second will include developers and administrators. The developers shall be the first ones to be impacted by this project, and thus, will require a thorough analysis and evaluation of each step along the path of development. Then, the administrators will be affected by the project

because they shall belong to the regulatory body for land record management and the protocols will be shifted in the work. The users can vary from record to record; there will be records that shall require the involvement of multiple stakeholders (users). While, the validation will be done by the administrators, which shall be none other than the respective land management authority. The authority shall not only approve of any transactions (affecting land or its documentation) but also be the central authority in the verification procedure.

2.9.2 Key High-Level Goals and Problems of Stakeholders

The main idea of the project is to reduce the overhead of manual work and physical/non-digitized storage. Hence, efficiency and pace of work are major high-level goals. Beginning with the developers; the gap between industry and academia shall be a major setback and precision will be required in the research phase to find out the most relevant work. The stakeholders will be faced with the trust deficit of blockchain technology, considering the recent legal actions and misunderstandings of certain applications. There is an education and training gap as well, whereby the users as well as the administrators shall be required to understand the new system. Lastly, the stakeholders shall need to put their faith in the new system as the documents will be completely saved in the new system.

2.10 Conclusion

The main areas that our project shall be targeting and affecting in the short-term are academic and commercial, but there shall be an impact on the mentioned circuit in the long run as well. This project can pave way for more research and development in the field of decentralized storage by acting as a catalyst. The records system is an entity that shall soon find application in the realm of the corporate world as well. By solving the problems of and assisting all stakeholders, the project will be bringing a multitude of solutions to the table.

Chapter 3: Literature Review / Related Work

This section of the report explores details of different existing platforms, tools, and techniques for the solution of land registry verification, maintaining these records, and some tools that assist users and authorities in signing agreements of renting lands between parties utilizing the key features of blockchain. There are numerous conceptual research papers and few platforms that provide these services commercially but have not been implemented. We will examine a few of those platforms, and tools and provide an extensive analysis of the research papers that have explored the concept of land record verification, the rental agreement between users, and the maintenance of these records using blockchain.

3.1 Definitions, Acronyms, and Abbreviations

In this section, brief definitions of some key technologies related to the literature review of research papers and existing platforms. Based on the architecture, there are two types of applications:

- Centralized Applications
- Decentralized Applications or Distributed Applications

3.1.1 Centralized Applications

In this type of application, all the users of the system accept the local databases for the data storage with the centralized ledger that is controlled and maintained by the trusted central/third party. Record keeping has always been centralized, which requires trusting central record keepers, but you still have to rely upon third parties.

Currently, almost all the applications and networking platforms like Facebook, WhatsApp, Google, etc. have created a large number of connections and also made it a big responsibility to manage user's privacy and data that is the most important asset of any user because all the user's data lies on a centralized server.

3.1.2 Decentralized Applications

In this type of application, every user of the system is completely self-directed, and distributed and runs on a peer-to-peer (P2P) blockchain network rather than on a centralized server/computer. Each user or entity is referred to as a peer of the network. For decentralized applications, blockchain technology is widely used because of its secure architecture, automated smart contracts, and decentralized currencies, and can replace centralized applications. Due to confidentiality, integrity, and transparency constraints, it is becoming mandatory to use blockchain-based applications.

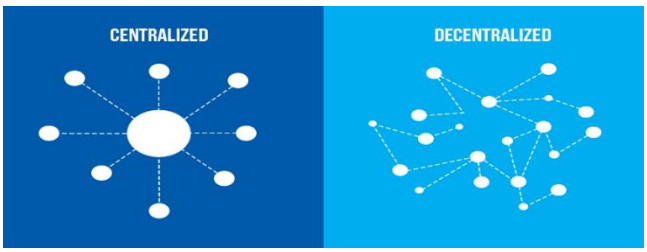

Figure 2: Centralized and Decentralized Networks
This figure represents how centralized and decentralized networks are interconnected.,

3.2 Detailed Literature Review

Land Records are a controversial entity in micro as well as macroeconomics, with the handling, storage, and update of data being unsafe at the core. In our local industry, there have been numerous attempts in digitizing the data, but our project will provide a two-pronged solution. Keeping in mind the vulnerabilities of the land record storage systems, especially the security, we aim to build a system that stores the records using blockchain. The idea is centered on the decentralized nature of blockchain that will ensure that the data is safe from tampering and loss.

Decentralized land record filing system by implementing blockchain technology to store the land data, which helps to attain integrity and security. This system presents ways in which the authority can maintain the records of lands efficiently or the information about their ownership securely. The system will also be able to maintain a contract of rental agreements between parties. Authorities (e.g., Law enforcement agencies, Land Development Authorities, and courts) will be able to add and access land data. We'll briefly describe some of the online platforms, and tools and discuss the research papers that have been written around this problem. The details of such papers are as follows.

3.2.1 Blockchain-Based Land Registration System

A scalable solution that ensures rapid and reliable transfer of funds between the related parties and is designed specifically using smart contracts.

3.2.1.1 Summary of the Research item

A decentralized record-maintaining system based on blockchain technology will remove threats and vulnerabilities to the important records of land because the blockchain uses cryptographic functions for the encryption, integrity, and process of authentication. Because of this reason, blockchain technology can be used for secure processes instead of relying on external or third parties by a smart process of the programmed transaction recording system. This paper provides the conceptual framework based on the proof-of-concept for future use that will benefit Pakistan's land registration agency [1].

3.2.1.2 Critical Analysis of the Research Item

The strength and weaknesses of this paper that we identified are stated below:

- The research paper explores the impact of using blockchain technology in this niche on a large scale.
- It outlines the current system of land registry in Pakistan and states all the stakeholders need to be involved in the process.
- This paper shows the general benefits of blockchain technology but it doesn’t have detailed knowledge about the specific stack or technology to use for the implementation of this system.
- It's only the conceptual framework that’s ideally looking easy to implement, but when developers start implementing any idea there comes a lot of barriers to resolve.

3.2.1.3 Relationship to the proposed research work

This research paper is relevant to the first module of our proposed work because it explores the impact and benefits of implementing such a system utilizing blockchain technology. It also

would be helpful to implement the use cases as it describes the current system and all stakeholders of the system.

3.2.2 Securing Land Registration using Blockchain

The following piece of work analyzes different points that are required for successfully implementing a secure land registration system. The salient features are as follows.

3.2.2.1 Summary of the Research Item

The land registry system using blockchain uses the hash mechanism SHA256 and Proof of Work (POW) that makes the information related to transactions more secure. It uses an Elliptic curve cryptographic algorithm to verify if the signature is signed by the user or not and the Merkle tree is used to link transactions and reduce disk usage. In the blockchain, the blocks are connected through the hash generated by SHA256 which represents the complete set of transactions within blocks.

The system consists of 12 nodes for the calculation of proof-of-work (POW) and is responsible for mining new blocks, adding a new block, and verifying the transaction that also updates the version of the land registry (e.g., ownership, etc.). The system has been tested on 200+ land transactions that offer updated versions of the registry with tamper proof [2].

3.2.2.2 Critical Analysis of the Research Item

The strengths and weaknesses of this paper that we identified are stated below.

- This research paper not only explored the general impact and benefits of a decentralized record filing system but also contains a detailed technical overview of the system.
- This system also has been tested on more than 200 transactions which is a motivation to implement it.
- The weak spot of this paper is, it's been tested on different regions that don't include Pakistan and the existing system here in Pakistan is a bit different and complex which also includes more stakeholders so this is an area that can be explored in our research work and further can be implemented.

3.2.2.3 Relationship to the Proposed Research Work

Blockchain technology is relatively a new technology and not so saturated so there are a lot of areas of it to be explored yet. Without knowing it in depth we can't dive into making any project using it so this research paper is perfect for it as it will help us in this while working on our proposed solution to the problem statement.

3.2.3 LeewayHertz

An effective and efficient system built on blockchain technology that is centered upon property and registry management.

3.2.3.1 Summary of the Research Item

LeewayHertz, a blockchain-based commercial land registry platform, involves three stakeholders: the buyer of the property, the seller of the property, and the land inspector (Admin) of the platform. The workflow for the platform is like the general user registering to the platform for either selling or buying the property. If the user is a seller, the user will upload the specifications of the property. If the user is a buyer, the user will request the existing

listings. In both conditions, the transaction will be recorded on the blockchain. On the buying request, after the seller approves, the admin gets notified, admin verifies the transaction, and initiates the transfer.

The verification of the document is done based on hash; any authorized party can upload the signed document to the registry to check the authenticity. If the hash generated after uploading the document and signing the document is the same, then the document is genuine, and it's built on Hyperledger Sawtooth [3].

3.2.3.2 Critical Analysis of the Research Item

The strength and weaknesses of this platform that we identified are stated below:

- It is a commercial platform that has been used by different housing societies abroad that have provided positive feedback.
- The product is not readily available for public consumption, in other words, it's an administrative service only that only reduces the overhead on the admin side but not on the consumer side.
- The platform is limited to the sale and purchases only while we proposed to make a proper land registration platform that will offer verification, rental agreement, and sale purchase features to the general public as well.
- The platform is built on Hyperledger Sawtooth that is not been publicly adopted by the consumers yet, while we are choosing a layer-2 solution of Bitcoin.

3.2.3.3 Relationship to the Proposed Research Work

LeewayHertz is a commercial platform that has been deployed in many locations and is also trusted by many vendors which increases the usability of this solution. We will use this as assistance for one of the modules and also will implement using reliable blockchain technology.

3.2.4 Gateway Digital

A scalable solution that ensures rapid and reliable transfer of funds between the related parties and is designed specifically using smart contracts.

3.2.4.1 Summary of the Research Item

Gateway Digital developed a land registry platform for property registration and ownership transfer of the land using smart contract technology. It is a scalable decentralized platform that allows buyers and sellers to make agreements directly without any middleman. Other than general website management the smart contract gets triggered when the seller requests the transfer, and the smart contract gives access to the land inspector. Once the process is completed smart contracts automatically transfer the funds and ownership title to the buyer. The seller and buyer both sign the documents using electronic signatures and then these signatures get recorded on Blockchain [4].

3.2.4.2 Relationship to the Proposed Research Work

Gateway is related to the third and final module of our project, the ownership transfer of property. The advanced system would help scale our system by covering its weaknesses of it.

3.2.5 Blockchain-Based Solution to Make Land Registry Reliable

A system that has been implemented across the border used decentralization to create a solution for the land registry hassle. The system ensures the integrity of data using available technology and allows for a real-time experience.

3.2.5.1 Summary of the Research Item

This is a project of building a land registry platform using blockchain technology for Panchkula, in the state of Haryana, India. It is based on 'proof of concept' also known as evidence-based for land registries in the region. The project incorporates key features like the immutable history of transactional records, authenticity, and integrity. This allows anyone to see the records at any time. The project uses Ethereum blockchain technology and allows the administrator to monitor the state of the property and sale deals in real-time and record the history of all transactions permanently [5].

3.2.5.2 Critical Analysis of the Research Item

The strength and weaknesses of this project that we identified are stated below:

- It is an evidence-based project also known as 'proof of concept' that enhances its credibility of it.
- The project had some conflicts with local legislation which is its weak point of it.
- It uses fewer verification and biometric techniques for the verification and transfer unlike the latest available approaches such as electronic signatures through cryptographic functions.

3.2.5.3 Relationship to the Proposed Research Work

The people who did a feasibility study and analysis before starting the project explored the side of tempering land records of poor people. We will use this as assistance for one of the modules and also will implement using reliable blockchain technology to overcome and maintain the integrity of records.

3.2.6 Secure Blockchain-Based Housing Rental Platform

This project worked out a solution for renting out houses using a blockchain-based solution.

3.2.6.1 Summary of the Research Item

In this research paper, the researcher highlights the problem in the housing rental market and proposed a solution based on blockchain technology. In their solution, they replaced the third party that needs to be involved for reliability and trust with smart contracts using blockchain technology to automate the process with a faster processor and ensure the identity of the legal party. They use zero-knowledge proof for this purpose and encryption methods to secure the private information of parties [6].

3.2.6.2 Critical Analysis of the Research Item

The strength and weaknesses of this project that we identified are stated below:

- It is an evidence-based project also known as 'proof of concept' that enhances its credibility of it.

- The paper explores algorithms such as zero-knowledge-proof and other encryption algorithms that solve the drawbacks of traditional housing rental platforms.
- It uses fewer verification and biometric techniques for the verification and transfer unlike the latest available approaches such as electronic signatures through cryptographic functions.

3.2.6.3 Relationship to the Proposed Research Work

The research paper is relevant to the third module of the project which is maintaining rental agreements between parties. As it explores advanced algorithms to ensure the drawbacks of traditional rental systems. We can implement them in our project.

3.2.7 Midasium

The research paper is relevant to the third module of the project which is maintaining rental agreements between parties. As it explores advanced algorithms to ensure the drawbacks of traditional rental systems. We can implement them in our project.

3.2.7.1 Summary of the Research Item

Midasium made software and smart tenancy contracts. It's software that is available online for landlords and property agents to assist them with the management of the cash flows of their property dealings. This application brings the automation of the process replacing traditional mechanisms using smart contracts [7].

3.2.7.2 Critical Analysis of the Research Item

The strength and weaknesses of this project that we identified are stated below:

- It overcomes the drawbacks of the existing system that involves a lot of paperwork and processing by reconciling the cash flow through the contracts.
- It brings transparency that can be verifiable by anyone reducing fraud.
- The project delivers security to bond deposits, tenancy ledgers, and transparent expense management that increases investment opportunities.

3.2.7.3 Relationship to the Proposed Research Work

The research paper is relevant to the second and third modules of our project that is rental agreement management and ownership shipment of the property. In both of these modules, the transfer and management of cash flows between parties is a nail-biting concern so that our clients can rely on it. This project highlights the mechanism of how it will work reliably, securely, and transparently.

3.3 Literature Review Summary Table

This section contains the summary of the literature reviews performed for the proposed solution work in a tabular form.

Table 1: Summary of the references explored

The summary of various related platforms and research papers published in recent years

No.	Name, reference	Owner(s)/Author(s)	Year of Publish/Release	Description
1.	Blockchain-Based Land Registration System: A Conceptual Framework, [1]	Muhammad Irfan Khalid, Jawaid Iqbal, Saddam Hussain, Amerah Alabrah	2022	It is a conceptual framework of a decentralized record-keeping system based on blockchain technology that will remove threats and vulnerabilities to the important records of land because of the blockchain features.
2.	Securing Land Registration using Blockchain, [2]	Krishnapriya, Greeshma Sarath	2020	A conference paper about a land registry system using blockchain that uses hash mechanism SHA256 and Proof of Work (POW) that makes the information related to transactions more secure. It uses an Elliptic curve cryptographic algorithm to verify if the signature is signed by the user or not and the Merkle tree is used to link transactions and reduce disk usage
3.	LeewayHertz, [3]	Akash Takyar	2021	LeewayHertz, a blockchain-based commercial land registry platform, involves three stakeholders: the buyer of property, the seller of the property, and the land inspector (Admin) of the platform aiming to reduce fraud and delays.
4.	Gateway Digital, [4]	Neraj Gemawat, Indrajeet Mitra, Vipin Moharir	2021	It is a group of companies that have developed a commercial application for land registration based on blockchain. It is relatively more advanced than other existing solutions.
5.	Blockchain-based solution to make land registry reliable, [5]	Alexandru Oprunenco, Chami Akmeemana	2018	This is a project of building a land registry platform using blockchain technology for Panchkula, in the state of Haryana, India. It is based on 'proof of concept' also known as evidence-based for land registries in the region.

6.	A secure blockchain-based housing rental platform, [6]	Rong Yu, Zhenqi Wang, Conghui Zhang, Shaopeng Guan	2021	A research paper that explores the traditional rental market drawbacks and proposed blockchain-based solutions to overcome them using advanced algorithms.
7.	Midasium, [7]	Michael Smolenski	2020	It's a platform that provides blockchain solutions for real estate and has already provided a commercial solution, and smart contract tenancy to manage the cash flow of properties.

3.4 Conclusion

We believe that the recent work done is mostly research-based in nature and thus, there is a lot of space in the practical realm. It is an opportunity for growth and translating blockchain ideas into a practical outcome. We are currently viewing several implementations in various capacities and we get the results that the system for Pakistan is a bit complicated and required. Considering the strength of our idea and with continuous efforts along with research and development, this project is not far from success. The project shall reduce manual effort by taking the whole process online and digitally. The use of decentralized technology and methodology adds to the integrity, and confidentiality and frees the work from being tampered with. Our project will automate the rental contracts between all involved parties and provide a means to verify all related documents.

Chapter 4: Software Requirement Specifications

This chapter entails all the software requirements, hardware requirements, quality attributes, and functional/non-functional requirements of the system. Furthermore, the graphical user interface, the use cases that will be implemented by the system, and the design of the database are also discussed.

4.1 List of Features

The following are the salient features of the system:

1. Decentralized Storage
2. Ownership Verification
3. Smart Contract-Based Rental Agreement
4. Smart Contract-Based Ownership Transfer

4.2 Functional Requirements

Following are the functional requirements of our project.

4.2.1 General Functional Requirements

- The system shall let the user log in using his or her credentials.
- The system shall store the files of the organization on the decentralized storage network.
- The system shall retrieve the files from the decentralized storage upon the user's request.
- The system shall manage wallets for usage as necessary.
- The system shall deploy a contract for the rental agreement between landlord and tenant.

4.2.2 Functional Requirements for the Administrator(s)

- The system shall let the administrator upload digitized copies of the property onto the decentralized storage.
- The system shall let the administrator retrieve the digitized copies of a property from the decentralized storage.
- The system shall let the administrator verify the ownership of a property.
- The administrator shall be able to view all of a person's owned properties by querying using his or her CNIC
- The system shall allow the administrator to administer and destroy a rental agreement between two parties (landlords and tenants).
- The system shall let the administrator view his or her recent activities stored in a log file.
- The system shall let the administrator transfer the ownership of a property.
- The system shall let the administrator add new properties to the system's database.

4.2.3 Functional Requirements for the Applicant(s)

- The system shall let the applicant verify his or her property using his CNIC and the address.
- The system shall let the applicant upload his credentials (CNIC, copy of the registry, etc.) when requesting ownership verification.
- The system shall limit the queries that can be generated against an applicant's CNIC to maintain the performance of the system.
- The system shall let the applicant view his properties using his or her CNIC.
- The system shall allow an applicant to request a rental agreement for a property.

4.3 Quality Attributes

Following are the quality attributes of the system.

4.3.1 Reliability

The system shall be reliable. It shall give correct information consistently and in any working environment.

4.3.2 Maintainability

The system shall be easy to maintain i.e., it shall be easy to improve the system by adding more code to it. It shall also be easy to integrate new features and functionalities into the system in the future.

4.3.3 Usability

The system shall have an easy to comprehend user interface which should be easy to navigate.

4.3.4 Correctness

The system shall provide correct results upon queries and provide correct navigation. It shall adhere to the functional requirements.

4.3.5 Integrity

The system shall provide only authorized access to the critical features and encrypt the files to maximize security and privacy when the files are stored on the decentralized storage. Moreover, backups of the data shall be stored on physical media to secure the data of the organization.

4.3.6 Efficiency

The system shall be efficient enough to fetch and deposit files, provide verification results and establish smart-contract-based agreements in the minimum period for feasible real-time usage.

4.3.7 Testability

It shall be easy to test the system for defects by its division into different modules.

4.3.8 Interoperability

The system shall be designed such that it shall be able to exchange data or services with other systems. The system shall work on multiple operating systems and different hardware configurations.

4.4 Non-Functional Requirements

Following are the non-functional requirements of the system.

- The system shall secure the files that are uploaded onto the decentralized storage by encrypting the data.
- The system shall maintain a log of all the activities performed by each administrator for accountability purposes.
- The system shall provide an easy-to-understand and simple user interface that can be learned with minimal effort.
- The system shall store backups of the files uploaded to the decentralized storage on a physical media in case of any data loss.
- The system shall perform all the functionalities in the most efficient way possible.

4.5 Assumptions

Following are the aspects that have been assumed for the specification of the system.

- The organization has the digitized copies of the properties available at hand or has the capabilities to digitize the physical data.
- The organization can restructure itself to operate through the proposed system.
- The organization has continuous and stable high-speed internet connectivity.
- The organization is open to training and hiring staff with a high-level understanding of the working of the system.
- The facility of obtaining cryptocurrency wallets will be provided to the public
- The organization has the clearance to access records of CNIC from the NADRA database.

4.6 Hardware and Software Requirements

Following are the hardware and software requirements for the system.

4.6.1 Hardware Requirements

The hardware requirements for the development and deployment of the system are as follows.

- A computer with 4 gigabytes or more RAM.
- Stable internet connection.

4.6.2 Software Requirements

The software requirements for the development of the system are as follows.

- JavaScript libraries such as React and Express.js for the development of the decentralized web application.
- Node.js for the JavaScript runtime environment.
- MongoDB for the NoSQL database.
- Decentralized storage APIs such as IPFS, and Web3.storage, NFT.storage, or Stacks Gaia.
- Clarity language for writing smart contracts.
- Tailwind CSS framework for the development of the front end of the decentralized web application.
- A JavaScript-enabled web browser.
- GitHub for version control and collaboration.
- A modern operating system (Linux, Mac, or Windows).
- A modern IDE or code editor.

4.7 Use Cases

The use cases for the various functionalities of the system are described as follows.

4.7.1 Login

Name	Login		
Actors	Administrator, Applicant		
Summary	The user shall provide his or her email and password registered with the system for authentication.		
Pre-Conditions	The user shall be registered with the system and his credentials must be present in the database. The user shall not already be logged into the system		
Post-Conditions	The user's session is successfully established and shall be redirected to the home page.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the login page.	2	The login page is displayed asking for a username and password.
3	The user enters a valid username and password.	4	The system verifies the email and password, establishes a session for the user, and redirects the user to the admin's panel if he is an admin or the applicant's panel if he is not an admin.
Alternative Flow			
3	The user enters an invalid username or password.	4-A	The system responds with an error message: *Incorrect username or password*

4.7.2 File Retrieval

Name	File Retrieval		
Actors	Administrator		
Summary	The admin retrieves the document for a specific plot from the decentralized storage.		
Pre-Conditions	The document/file for the plot must be uploaded beforehand on the decentralized storage.		
Post-Conditions	The file is displayed inside the integrated frame and the options to print a copy or save it on the disk are given to the user.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the ‘Retrieve File’ page.	2	The page for file retrieval is displayed to the user which contains the options for searching the document for a plot of land.
3	The user selects the area, block, and plot number from the given options and clicks on the 'Search' button	4	The system looks up the CID of the file from the database and uses it to retrieve the file from the decentralized storage and display it in the integrated frame.
5	The user clicks on the ‘Save’ button shown inside the integrated frame	6	The user clicks on the ‘Save’ button shown inside the integrated frame
7	The user clicks on the ‘Save’ button shown inside the integrated frame	8	The system prints the document for the user.

4.7.3 File Upload

Name	File Upload		
Actors	Administrator		
Summary	The user shall select the land for which he wishes to upload a document on the decentralized storage and attach the document. The system will upload the file on the decentralized storage and show a successful message.		
Pre-Conditions	The plot for which the file is being uploaded should be valid and the CNIC of the documents must match the CNIC of the new owner stored in the system.		
Post-Conditions	The file is uploaded to the decentralized storage and a “Document Uploaded Successfully” alert is displayed by the system.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the “Upload File” page	2	The page designated for uploading documents of the plot on the decentralized storage is displayed by the system
3	The user selects the appropriate plot for which the documents are to be	4	The system prompts the user to confirm the upload of the file

	uploaded, enters the correct owner's CNIC, and clicks on the upload button.		
5	The user clicks on “Confirm”.	6	The system uploads the file for the plot on the decentralized storage and displays an “Upload Successful” message to the user
Alternative Flow			
3	The user selects an incorrect plot for which the document is to be uploaded or inputs an incorrect CNIC of the owner.	4-A	The system responds with an error message: *The owner does not match the provided CNIC*
5	The user clicks on “Cancel”.	6-A	The system closes the prompt for the confirmation of upload without uploading any file onto the decentralized storage.

4.7.4 Ownership Verification

Name	Verify Ownership of a Property		
Actors	Administrator, Applicant		
Summary	The user shall provide the owner's details such as CNIC, name, and date of birth along with a property address he intends to verify the ownership of. The system will compare the details and intimidate the user about the ownership.		
Pre-Conditions	The plot must be present in the database of the system. The owner must have a valid CNIC or B-Form.		
Post-Conditions	The user is shown an alert confirming or denying the ownership of the plot about which query is submitted to the system.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user clicks on the “Verify Ownership” link.	2	The ownership verification page is displayed.
3	The user enters the location of the plot and the CNIC of the genuine person claiming to be the owner of the plot and clicks on the verify button.	4	The system will check the ownership of the plot from the blockchain and show the user that the person whose CNIC has been given is the owner of the property.
Alternative Flow			
3	The user enters the location of the plot and the CNIC of a fake person claiming to be the owner of the plot and clicks on the verify button.	4-A	The system will check the ownership of the plot from the blockchain and show the user that the person whose CNIC has been given is not the owner of the property.
3	The user is not an administrator and has exceeded his query limit for verification of ownership and submits another query to verify the ownership of a property.	4-A	The system shows an error message that the daily query limit has been exceeded.

4.7.5 Rental Agreement Generation

Name	Rental Agreement Generation
Actors	Administrator

Summary	The user will provide the CNIC of the tenant and the landlord, the location of the property to be rented, the other details of the landlord and the tenant, and also the agreed upon monthly rent and total rent period along with the wallet addresses of both tenant and landlord. The system will then deploy a smart contract indicating the rental agreement which will automatically deduct the amount from the tenant's wallet and deposit the rent in the landlord's wallet.
Pre-Conditions	Both the landlord and the tenant should have valid CNIC, and cryptocurrency wallets and the property to be rented must be developed.
Post-Conditions	The system shall deploy a smart rental agreement and show the summary to the administrator which he can use for other clerical purposes.
Special Requirements	None

Basic Flow			
Actor Action		**System Response**	
1	The user clicks on the “Rental Agreement” link.	2	The page for the rental agreement between landlord and tenant is displayed by the system.
3	The user enters the details of the landlord including his CNIC which matches with the CNIC of the owner of the property, along with the details of the tenant, the wallet addresses of both tenant and the landlord, the monthly rent agreed upon by the two parties and the duration of rent, and finally clicks on the “Deploy” button.	4	The system verifies all the details successfully, deploys the smart contract to the blockchain, and displays a message saying "Agreement Generated Successfully"
Alternative Flow			
3	The user enters the details of the landlord including his CNIC which does not match with the CNIC of the owner of the property, along with the details of the tenant, the wallet addresses of both tenant and the landlord, the monthly rent agreed upon by the two parties and the duration of rent, and finally clicks on the “Deploy” button.	4-A	The system will display an error message stating that the details of the landlord do not match the owner of the property.

4.7.6 Rental Agreement Request

Name	Rental Agreement request
Actors	Applicant
Summary	The user will apply the system so that an administrator may arrange a rental agreement between him and a tenant.
Pre-Conditions	The user must be an owner of the property for which he is requesting a rental agreement.
Post-Conditions	The system shall intimidate an administrator about the request so that he may arrange a smart-contract-based rental agreement between the landlord and tenant

Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user clicks on the “Apply for Rental Agreement” link	2	The system displays the page for the rental agreement to the user.
3	The user enters his CNIC along with the address of the property he owns and clicks on the “Request for Rental Agreement” button.	4	The system verifies that the applicant is the owner of the property, notifies an administrator about the request, and shows a message to the user saying "Request Submitted Successfully".
Alternative Flow			
3	The user enters a CNIC which does not match the CNIC of the owner of the property for which he is requesting a rental agreement.	4-A	The user will display an error message stating that the details of the landlord do not match the owner of the property.

4.8 Graphical User Interface

The GUI dumps of screens that will be implemented in the system have been illustrated as follows.

4.8.1 Login:

This page will allow the authorized administrator and applicant to login into the system through the credentials provided by the organization.

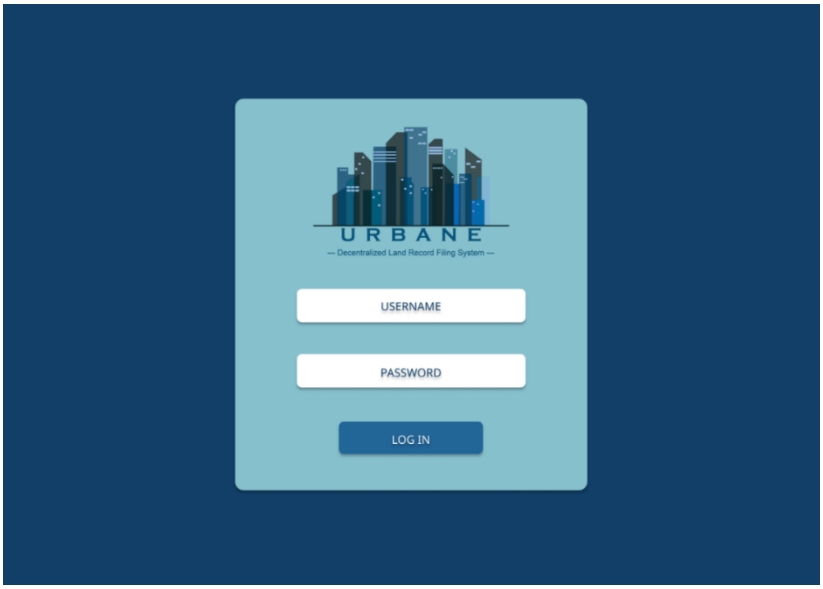

Figure 3: Login Page

The above figure shows the login page which will be used by both the administrator and the applicant.

4.8.2 File Retrieval

This page allows the administrator to upload original documents of the property into the decentralized storage system to use later for further processing i.e., Verification or ownership shipment of the property.

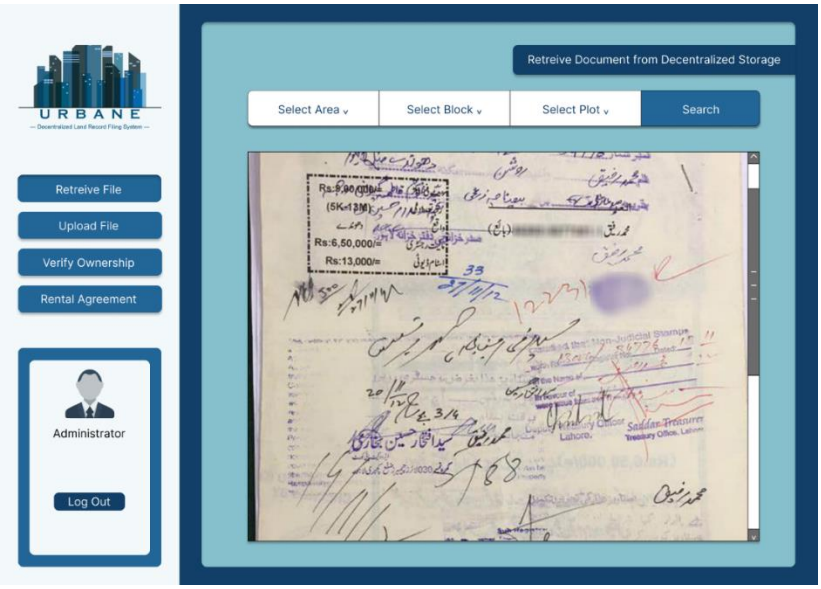

Figure 4: Original Documents Upload panel by Admin

The above figure shows the page used by the admin to download and view files from the decentralized network.

4.8.3 File Upload

This page allows the administrator to retrieve original documents of the property from the decentralized storage system to use later for further processing i.e., Verification or ownership shipment of the property.

Figure 5: Document Retrieval Panel by Admin

The above figure shows the page which will be used by the administrator to upload files to the decentralized network.

4.8.4 Ownership Verification

This page allows the administrator to verify the documents using the applicant's provided credentials with the help of a smart contract.

Figure 6: Document Verification Panel by Admin

The above figure shows the page which will be used by the administrator to verify the ownership of a claimant.

4.8.5 Rental Agreement Generation

This page allows the administrator to maintain the rental agreement between the parties using smart contracts to automate the process.

Figure 7: Rental Agreement Panel by Admin

The above figure shows the page used by the administrator to create and deploy the smart-contract based rental agreement between two parties.

4.8.6 Ownership Verification (Applicant's View)

This page allows the applicant to request document verification from the existing organization's data record.

Figure 8: Document Verification Request Panel by Applicant

The above figure shows the page used by the applicant to verify the ownership of property of a claimant.

4.8.7 Application for Rental Agreement

This page allows the applicant to request the rental agreement between parties.

Figure 9: Rental Agreement Request Panel by Applicant

The above figure shows the page used by the applicant to request a rental agreement between him and a tenant.

4.9 Database Design

The design of the database used in the proposed system is explained using the below sections.

4.9.1 ER Diagram

The ER diagram of the proposed system's database is illustrated below:

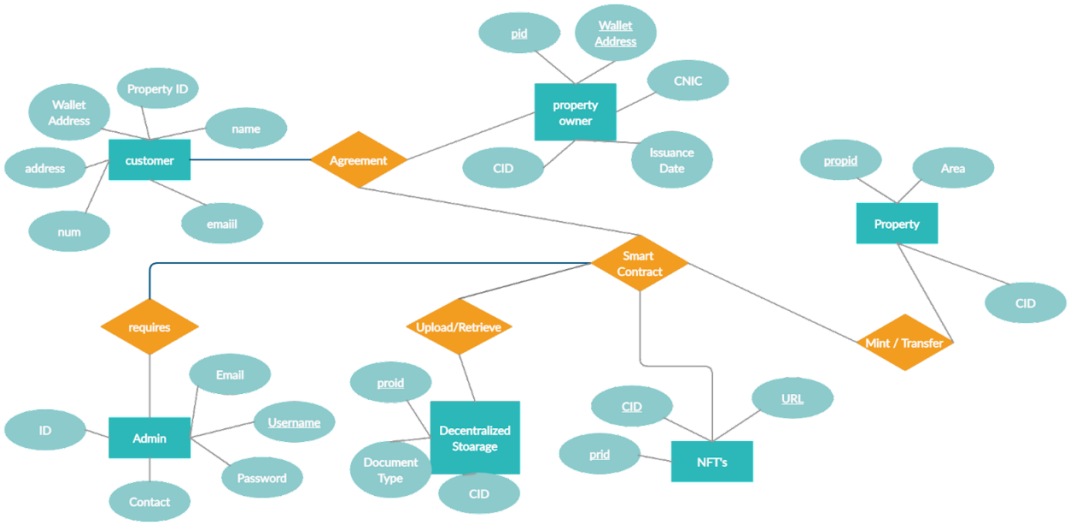

Figure 10: ER Diagram

The above figure shows the entity relationship of the database which will be used by our system.

4.9.2 Data Dictionary

The data dictionary of the database which will be used by the system is shown below.

Table 2: Data Dictionary

The table contains the dictionary of all the data which will be stored in the system's database.

Entity	Attribute	Data Type	Nullable	Relation to	Relation type	Description
Customer	Name	Varchar	No	Property Owner	1 to 1	Name of Customer
	Email	Varchar	No			Email of Customer
	Property ID	Int	No			Foreign Key of Property
	Wallet Address	Int	No			Wallet Add. of Customer
	Address	Varchar	Yes			Address of Customer
	CusID	Int	Yes			Primary key of Customer

Admin	ID	Int	No	Property	1 to Many	PK of Admin
	Email	Varchar	No			Email of Admin
	Username	Varchar	No			Username of Admin
	Password	Varchar	No			Pass of Admin
	Contact	Int	Yes			Contact of Admin
Property Owner	PID	Int	No	Property, Storage	1 to Many	FK of Property
	Wallet Address	Int	No			Wallet Add. of Owner
	CNIC	Int	No			PK of Owner
	Issuance Date	Date	Yes			Date since he is the owner
	CID	Int	No			FK of Storage
Decentralized Storage	PID	Int	No	Property	One to One	FK of Property
	Document Type	Varchar	No			Documents type i,e Registry, etc
	CID	Int	No			PK of Storage
Property	PID	Int	No	Storage, NFT’s	One to One	PK of property
	Area	Varchar	No			Area of Property
	CID	Int	No			FK of Storage
	PID	Int	No	Property,		FK of Property

NFTs	CID	Int	No	Storage	One to One	FK of Storage
	URL	Varchar	No			URL to NFT
	ID	Int	No			PK of NFT

4.10 Risk Analysis

The technical and business risks that may be encountered during the development of the project are as follows.

Technical Risks: A framework or API being used in the development may become obsolete or lose support from its developers. Moreover, a technical issue may arise that causes compatibility or interoperability issues between the multiple libraries and APIs requiring the reassessment of the technological stack.

Legal Risks: A certain legal complexity may arise that may require the redesigning of the fundamental working of the project or make it impossible to apply for general use.

Business Risks: The organization for which the system is being developed may deem it financially infeasible to use it for their everyday operations.

4.11 Conclusion

The structure of our project shall comprise the basic skeleton of any software project. The task will be divided into different classes as per the design guidelines defined in this report. The team will seek continuous assistance from this document and continue to make any changes in the structure if felt necessary. The changes, however, will be completely minor in nature and aligned to not affect the functionalities proposed previously. An initial design of the user interface is provided in this document and shall act as the backbone of the front-end development, which is among the foremost and fundamental parts of the project. A thorough insight is given into the risk analysis so that every aspect is secure and there are no loose ends in the system.

Chapter 5: High-Level and Low-Level Design

5.1 System Overview

The system will be developed in such a way that it allows the user to verify documents, request and make rental agreements in an environment that stores data on decentralized storage, The system shall connect the decentralized mechanism to the front-end design.

A user will be able to upload and retrieve data to and from the system as required. It is necessary to note that the system is aimed to reduce the overhead caused by human interactions and digitize the process.

The user interface will provide all the necessary steps and requirements for the users, whether new or old, to login and sign up on it. The login and signup will not only comprise of username and password, but the wallet ID for the blockchain as well. This will enable the user to use the potential of the storage to the fullest.

The business layer will contain all the operations of the main classes, including the user data, record and data sharing operations. The operations that follow behind the UI will be catered to at this layer. Data is saved, input from the UI is taken and sent to be stored through this layer. Once the user logs in to the system and uploads a document for verification or rental agreement, the business layer requires the approval of the admin to proceed forward. After the admin approves, the original document is stored on the storage layer, and an NFT for it is generated. The business layer keep record of the NFTs and documents by storing the URL of each NFT in the property class.

The third layer of the system will be the decentralized storage system. The business layer will take data from the user and the admin, wait for appropriate actions from both i.e. requests and approvals, and passes the data to this layer. The decentralized storage will not only store the data, but also keep track of it to ensure a smooth retrieval process. The NFTs against each property and document will be generated by this layer, and the track will be kept on the business layer.

The users shall be able to login using their wallet ID, to ensure uniqueness and then allowed to request verification and rental agreements. The approval from the admin shall signal the system to proceed. NFTs will be used to ensure storage is unique and safe.

5.2 Design Considerations

5.2.1 Assumptions and Dependencies

Primarily the system will operate on the following assumptions:

- The data provided in the documents has been stored on multiple backups apart from our system.
- There has been a thorough digitization of the documents beforehand.
- The blockchain technology used in the project cannot be hacked (Although there is sufficient evidence to prove this, but this assumption is being included keeping the few recent accidents with the technology in sight).

Following dependencies will govern the working of this system:

- The blockchain under use will not go out of date or face technical issues that shall affect the efficiency, availability or accuracy of our system.

- There shall be no change in the core algorithms of Proof-of-transfer (PoX) or Proof-of-stake (PoS) that dictate the working of the decentralized mechanism underuse.
- There shall be no hurdle in the implementation of the system on the practical scale and all the users/admins involved will allow us to use the system in an actual setting without any trust issues.

5.2.2 General Constraints

The system lies in the common area between novelty and extension of previous work, but the general constraints of the system include:

- The compatibility issues (subject to full development of the project) between the technologies and tools underuse.
- The lack of research around the area of work (blockchain).
- The difference between the levels of achievements in the academic and the industrial world, in this line of work and its derivatives.
- Complete absence of working models that are scalable and resolve the issues under consideration, on the local or even industrial level.

5.2.3 Goals and Guidelines

Using the techniques and methodologies of the system will be aimed at catering the underlying goals:

- The main issue that shall be resolved by the development and implementation of the system is to streamline the current process of document storage and verification
- Solidify the trust element in the process of rental agreements regarding property
- Bring the managers/administrators of the land record filing and the general users closer by taking out any chances of compromise

Guidelines of the system are:

- The system is developed for storage of land records, the latter's ownership(s) and rental agreement(s).
- This system is an attempt to bring decentralized storage in the realm of public service, hence, all users are advised to read this documentation for general understanding and the documentations dedicated to each of the tools used for specific and in-depth understanding.
- In order to use the system effectively, all users and administrators will require a wallet ID on the blockchain that is underuse. The details of all sorts regarding technology are included in this document.

5.2.4 Developmental Models

A step-by-step and methodical technique was used for the development of this system. Each of the steps included background research, related studies (if any), related practical work (if any) and a systematic approach towards development and implementation.

Information gathering and market research constituted the first step of the process, followed by a thorough requirements engineering and technological analysis. Weighing in the possibilities

and closely studying the current models on the global market made up most of the next step. It must be noted that reviews of practical and academic works were a crucial part of the analysis process. Analyzing the compatibility and availability of the tools and then selecting the right ones among them was the next stage, this required detailed technical study and research. Designing the user interface came next, with proper tools to begin with, the team ensured that the system looks professional ad aesthetically pleasing.

For a system like this, it is important to weigh-in all tools individually and collectively, as the factors of functionality and compatibility are top priority. The first technical phase includes development of the prototype, which in our case is an evolutionary prototype. The prototype is developed using the same tools that will be used in the actual project, as it will add to the efficiency of the whole procedure.

The main development model underuse throughout the project is incremental in nature, as every new step is a derivative and successor of the previous one. The documentation of the whole work has run parallel and will act as the document of guidance during and after the complete development.

5.3 System Architecture

The software's core responsibilities are storing and retrieving files from an IPFS based decentralized storage system, verifying the ownership of a property and creating smart-contract based rental agreements for a landlord and his tenant(s).

Keeping these responsibilities/functionalities in mind, the architecture of the system was divided into three main modules which reside in the middleware of the application, as can be seen in the high-level architecture diagram below.

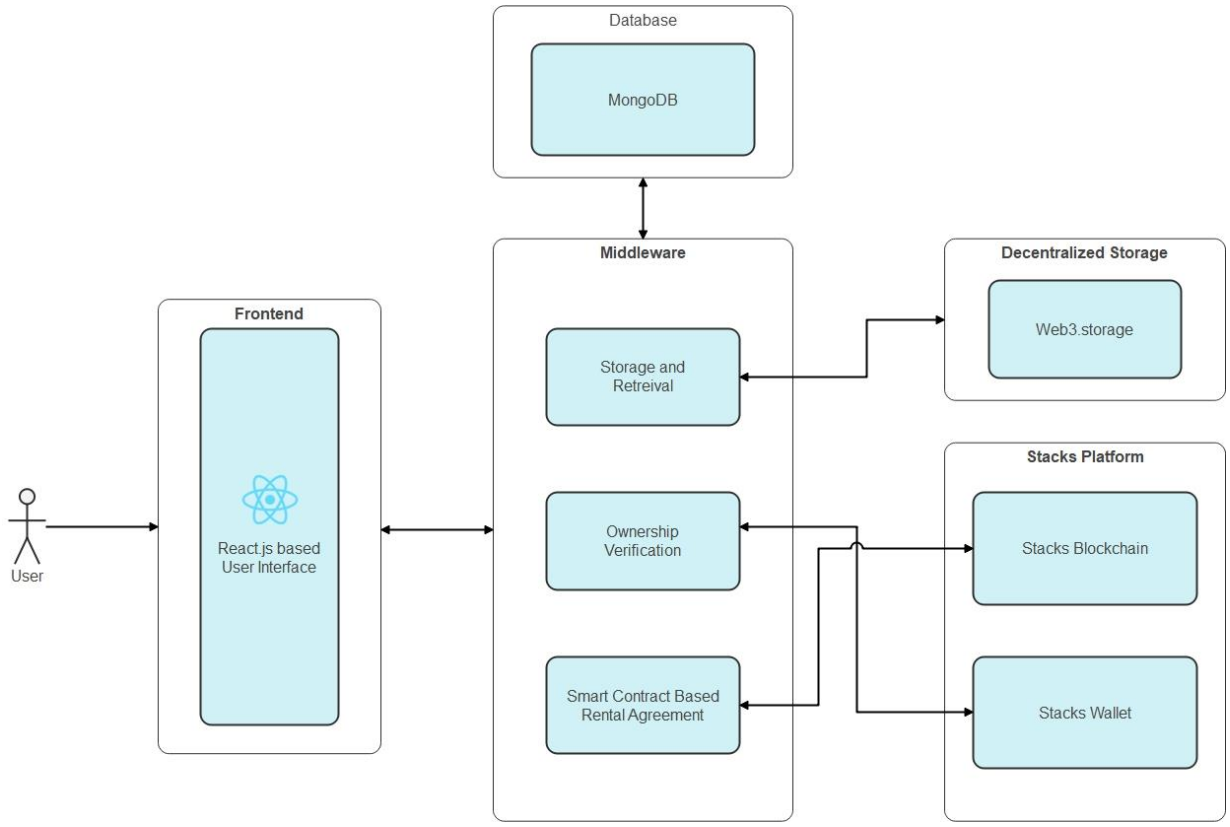

Figure 11: High-Level Architecture Diagram of the System

The figure above shows the high-level architecture diagram of the system, showing its modules and components.

5.3.1 Subsystem Architecture

The detailed discussions of the modules residing in the middleware is presented as follows.

5.3.1.1 Ownership Verification

This module will handle verifying the ownership of a property. The module will invoke a smart contract residing in the blockchain which will determine whether a person claiming to be the owner of a property is really the legitimate owner or note.

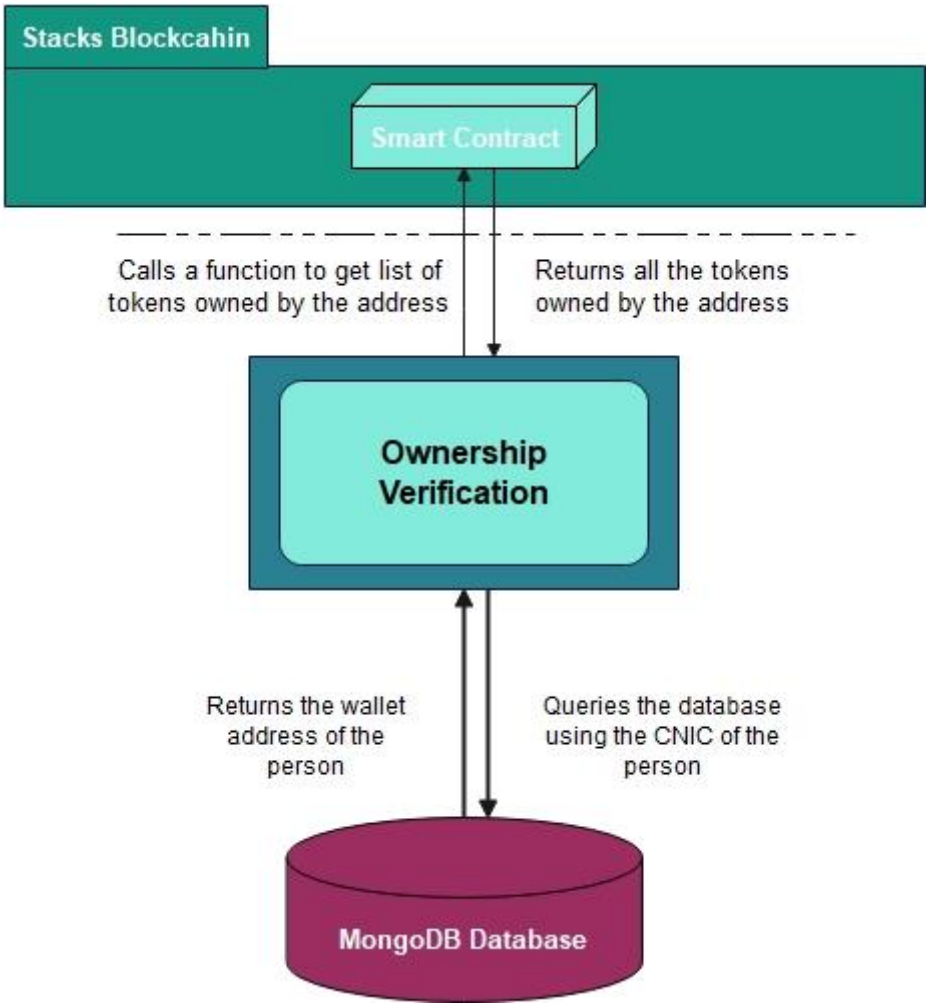

Figure 12: Low-level diagram of the Ownership Verification module
The figure shows the working of the Ownership Verification module at a low level.

5.3.1.2 Storage & Retrieval

This module is responsible for both storing the documents of a property on the decentralized storage network and also its retrieval. The module will also maintain an index of the CID (content identifier) of each document in the NoSQL database. The low-level diagram of the module in discussion is as follows.

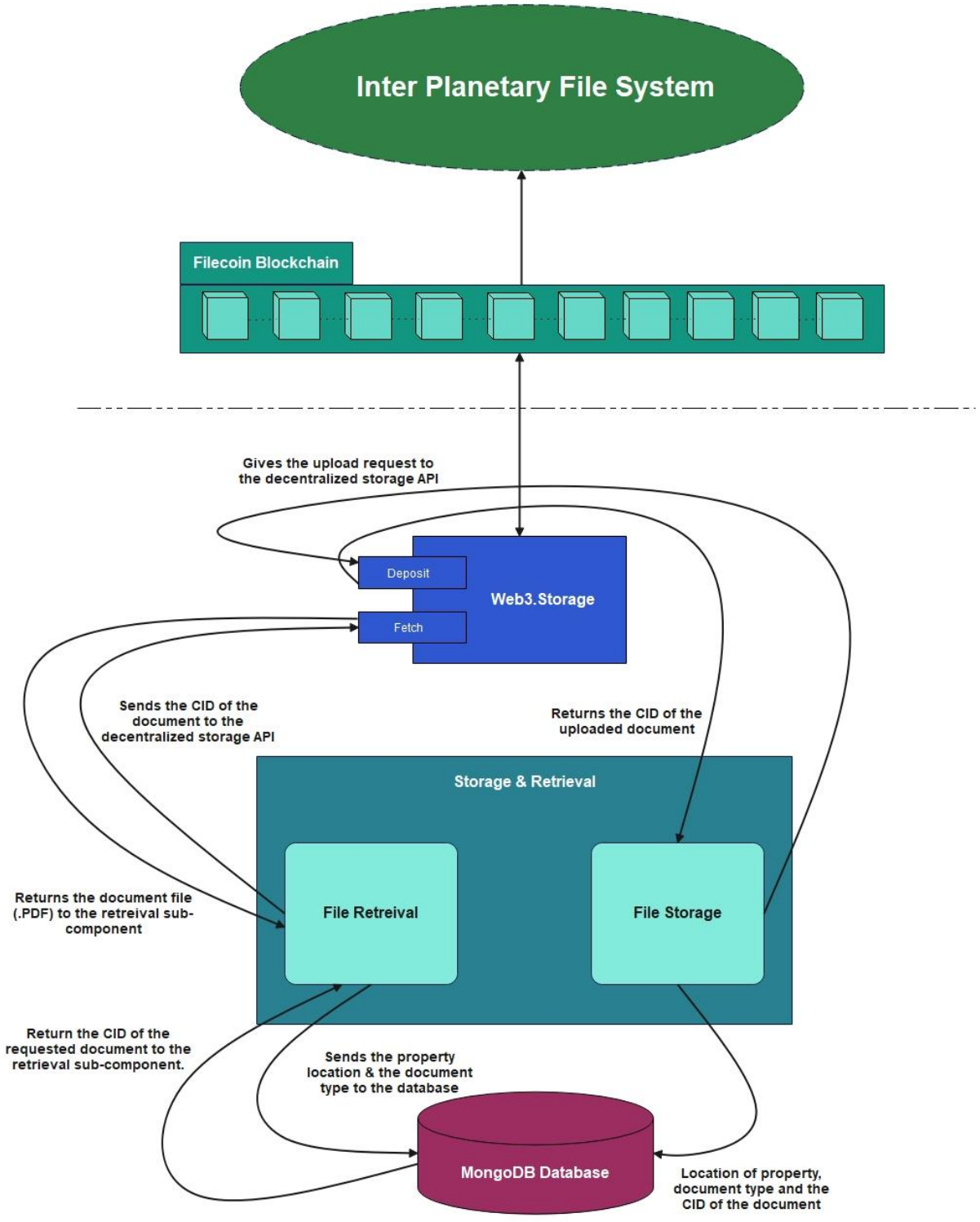

Figure 13: Low-level diagram of the Storage and Retrieval module
The figure shows how the Storage and Retrieval module stores and fetches data from an IPFS-based decentralized storage network.

5.3.1.3 Smart Contract Based Rental Agreement

This module is responsible for generating a rental agreement between a landlord and his/her tenant(s) using a smart contract. The smart contract takes input the cryptocurrency wallet addresses of all the parties involved, as shown in the figure below.

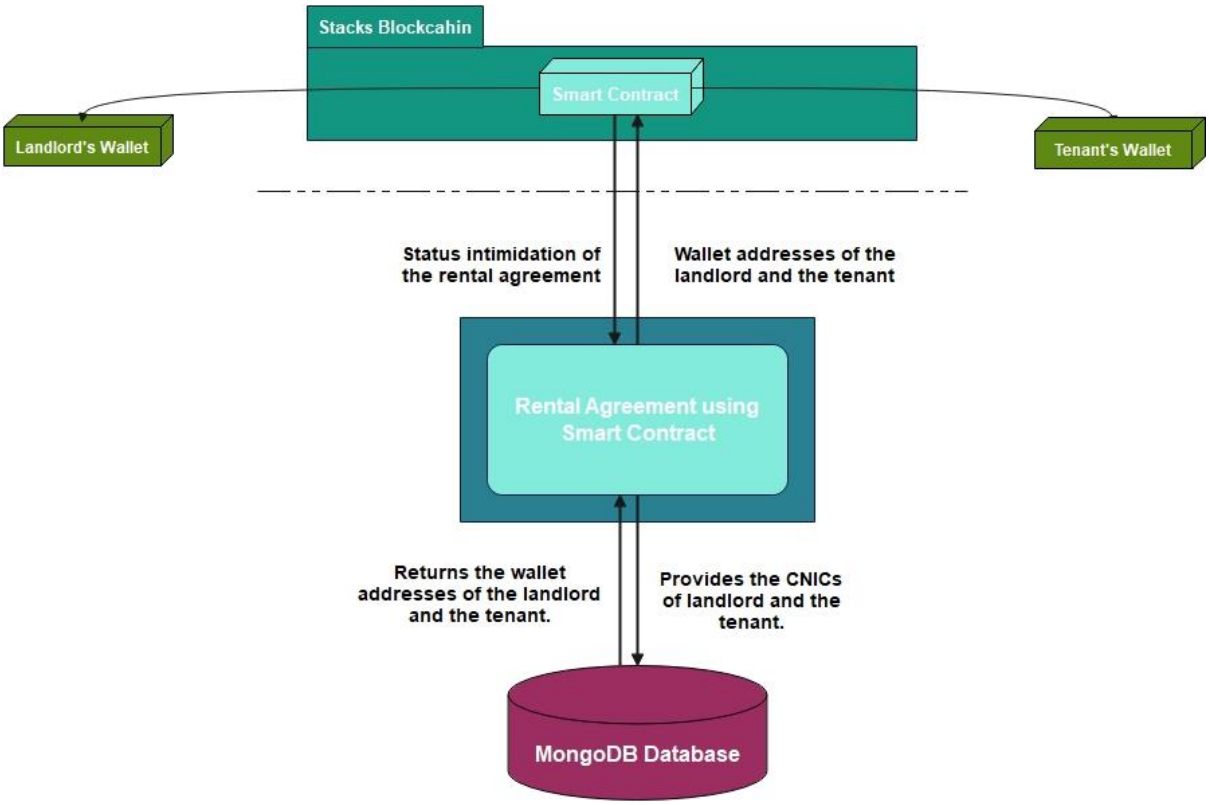

Figure 14: Low-level diagram of the Smart Contract Based Rental Agreement module
The figure demonstrates how the module will interact with a smart contract residing in the blockchain to generate a rental agreement between landlord and his/her tenant(s).

5.4 Architectural Strategies

The strategies used for the design of the architecture of the system are as follows.

5.4.1 Use of the MERN stack

The MERN stack comprises of the following technologies:

- MongoDB
- Express.js
- React.js
- Node.js

The MERN stack is a web development stack which uses JavaScript for every component of the application. The stack is based on the MVC model. The model is created using MongoDB, which is a NoSQL database which uses JavaScript and JSON to store data. Similarly, the view is developed using React.js which is a component-based JavaScript front-end development library which makes use of a virtual DOM for rendering elements on the browser. The

controller part is implement using Express.js, which is a middleware framework used to create the business logic of an application. This framework acts as a gateway between the database (model) and the frontend (view).

The decision to implement the system using this stack was mainly because of two reasons. One is that the complete stack can be implemented using a single language making the development and maintenance of the system less complex. The other one is that for integrating blockchain technologies, the developers of the APIs of these technologies provide libraries which are written in JavaScript. Therefore, by using an all-JavaScript stack, the compatibility concern of different blockchain technologies with each other becomes a non-issue.

From a design perspective, the design of a MERN application becomes very simple as a three-tier architecture can be used to organize the components of the system.

One other alternative JavaScript stack was the MEAN (MongoDB, Express.JS, Angular.JS, Node.JS) stack. However, the MERN stack was more preferrable as React.js is far simpler and easier to use and maintain when compared to Angular which is an extensive framework developed by Google to develop frontend user interfaces.

5.4.2 Web3.Storage

The Web3.Storage platform provides an API to store the data on the IPFS-based decentralized network. Although this functionality can be achieved by using the official IPFS libraries, there are some limitations (file persistency) due to which a platform which was backed by a blockchain was required.

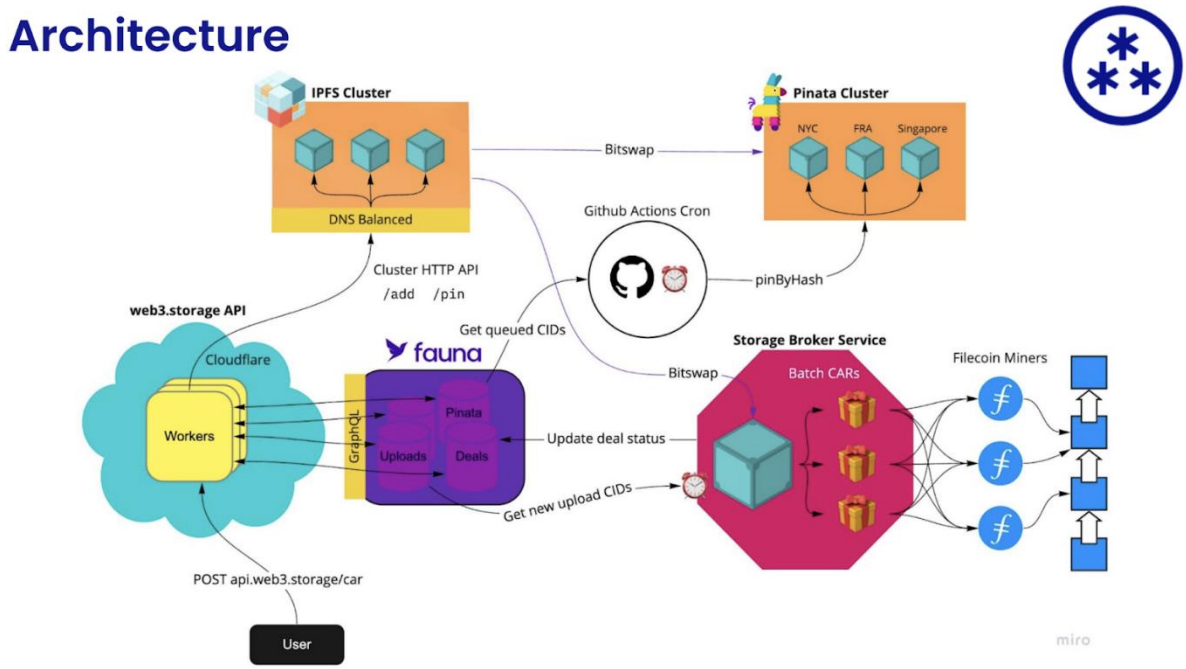

Figure 15: Architecture of the Web3.Storage platform
The figure shows how the Web3.Storage combines different technologies to provide a truly decentralized persistent storage system

Web3.Storage uses the Filecoin blockchain to add persistency to the IPFS platform. Filecoin is an incentivizing layer on top of the IPFS which incentivizes the storage of files on different nodes of an IPFS cluster.

5.4.3 Stacks Blockchain

The Stacks blockchain is developed on the existing Bitcoin blockchain and uses STX tokens. The blockchain comprises of a “testnet” which is uses as a testing blockchain and a “mainnet” which is the actual blockchain.

Although there were other choices for blockchain, Ethereum being the best alternative, Stacks was decided for development as it provides great toolkits for developing decentralized applications, including wallets and support for NFTs. Moreover, it also has a growing community in Pakistan.

5.4.4 External Databases & Persistent Data Storage

As discussed earlier, IPFS does not provide a persistent decentralized storage by default. Therefore, in order to ensure persistency, a third-party platform (Web3.Storage) is used which provides three gigabytes of persistent data storage on the free plan. However, extra cost can be paid in order to increase the storage capacity.

Moreover, the IPFS-based decentralized data storage platforms provide CIDs to identify the files on the network. These files need to be indexed with a key which can be used by an-end user to fetch the file from the network. For this purpose, MongoDB is used which can maintain this record.

5.4.5 User Interface

As the system is to be used by an organization whose sole motive is to gain an efficient way to file digitized records safely on the internet, therefore a simple and easy to understand user interface has been designed which can be learned through minimal training, saving costs and increasing efficiency. The interface is developed using the React.js library and a CSS framework like tailwindCSS or bootstrap.

5.5 Domain Model/Class Diagram

The class diagram of the system is as follows.

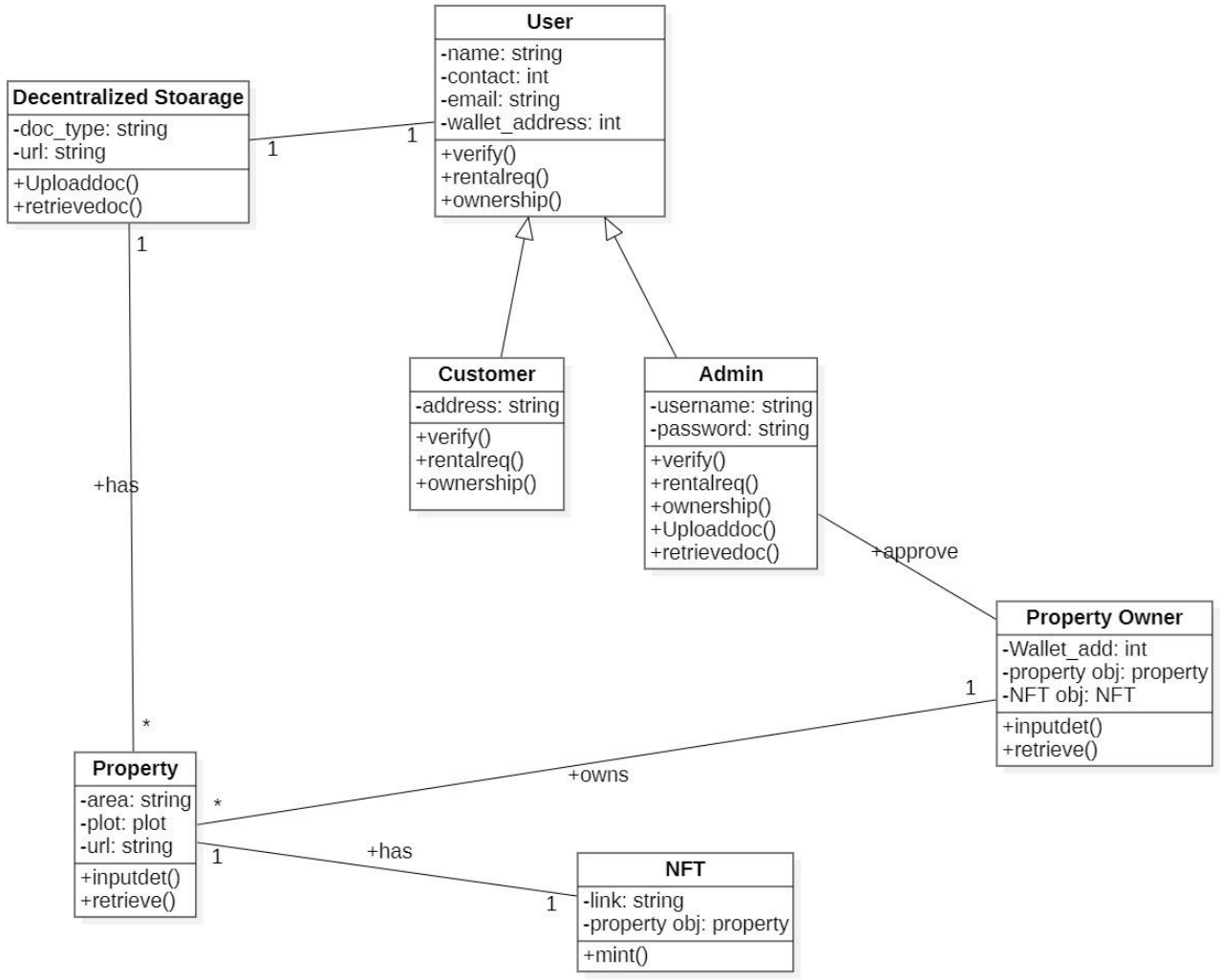

Figure 16: Class diagram of the system

The figure shows the class diagram of the system, describing each module and class with its functionality in detail

5.6 Sequence Diagrams

The sequence diagrams of each module of the system are as follows.

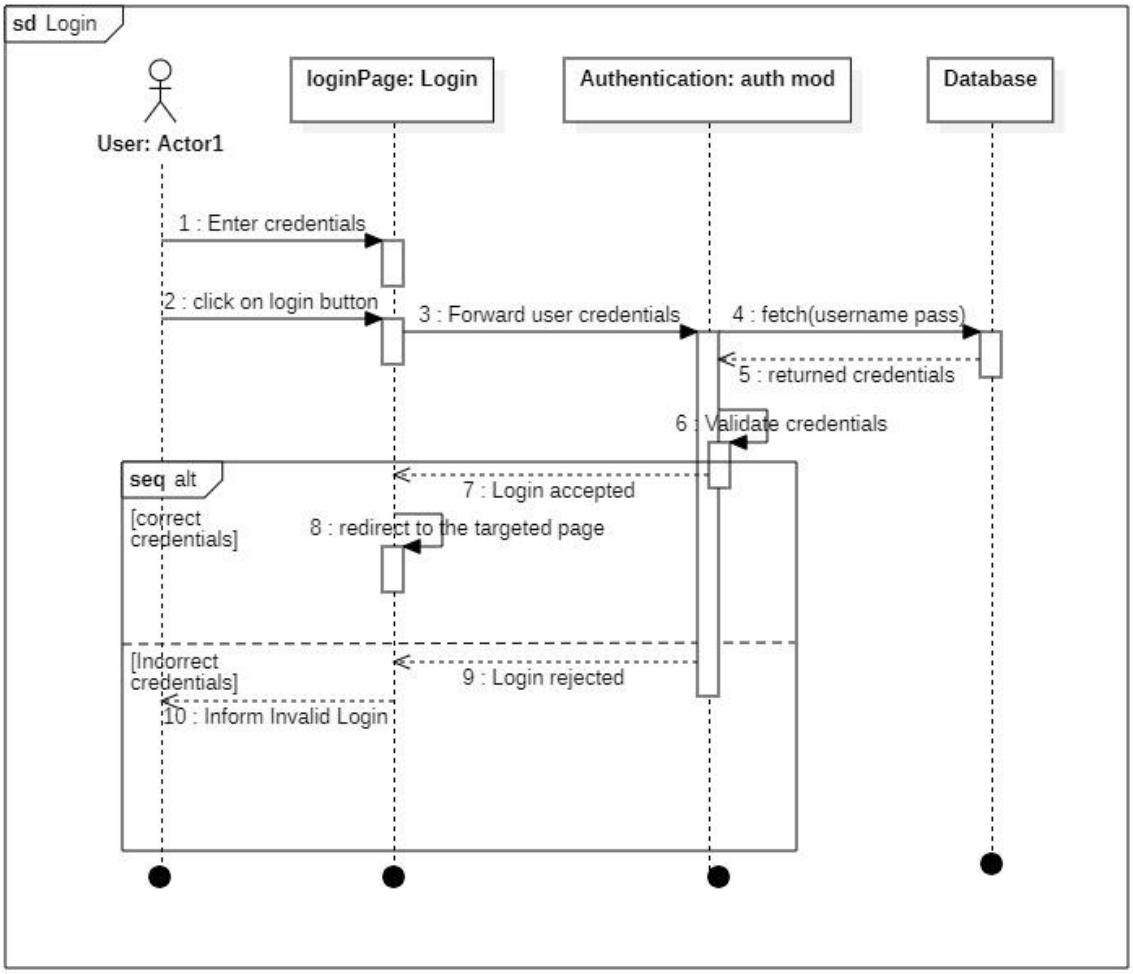

Figure 17: Sequence diagram of log-in functionality

The figure shows the sequence diagram of the log-in functionality of the system which is used to let an authorized user log in to the system.

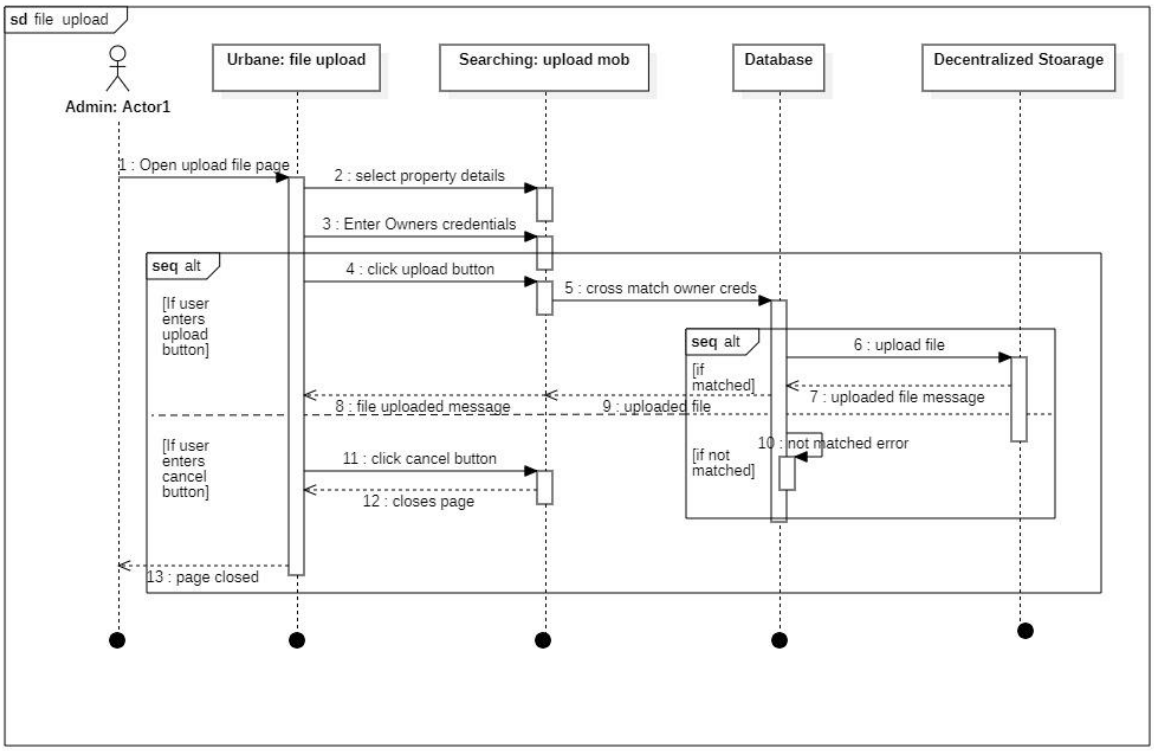

Figure 18: Sequence diagram of file uploading

The figure shows the sequence diagram for uploading a file to the decentralized storage

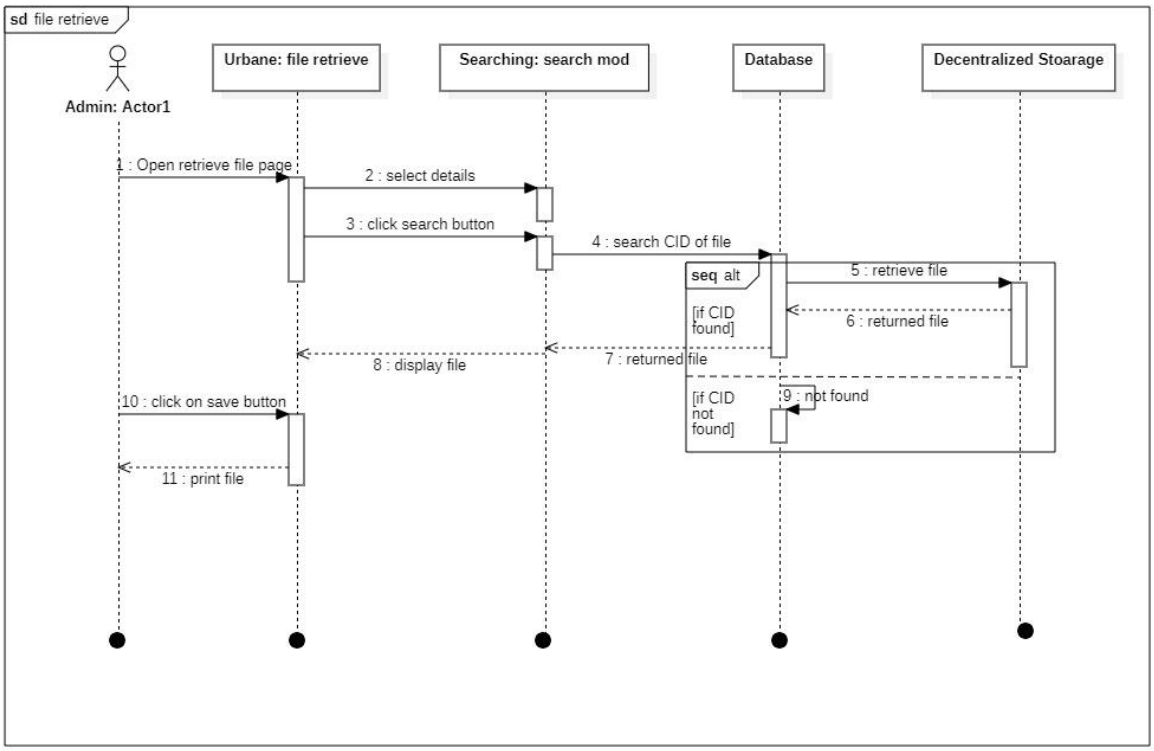

Figure 19: Sequence diagram of file retrieval

The figure shows the sequence diagram for retrieving files from the decentralized storage

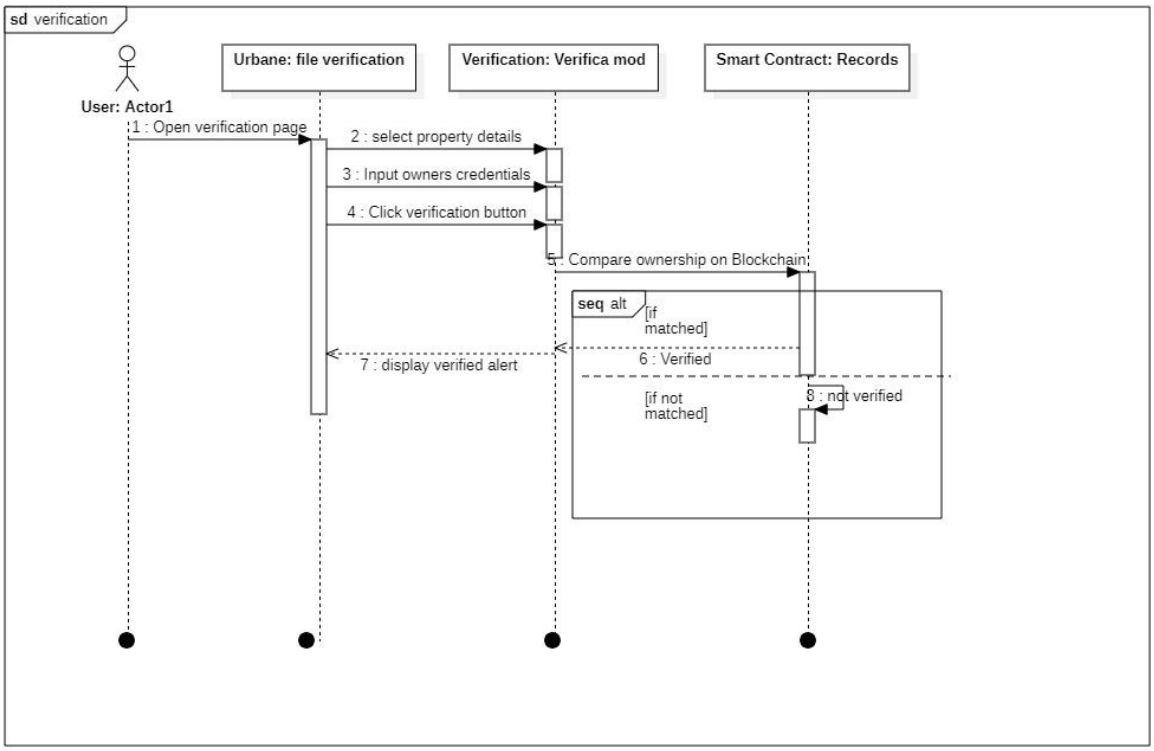

Figure 20: Sequence diagram for ownership verification

The figure shows the sequence diagram for verifying the ownership of a property using the system

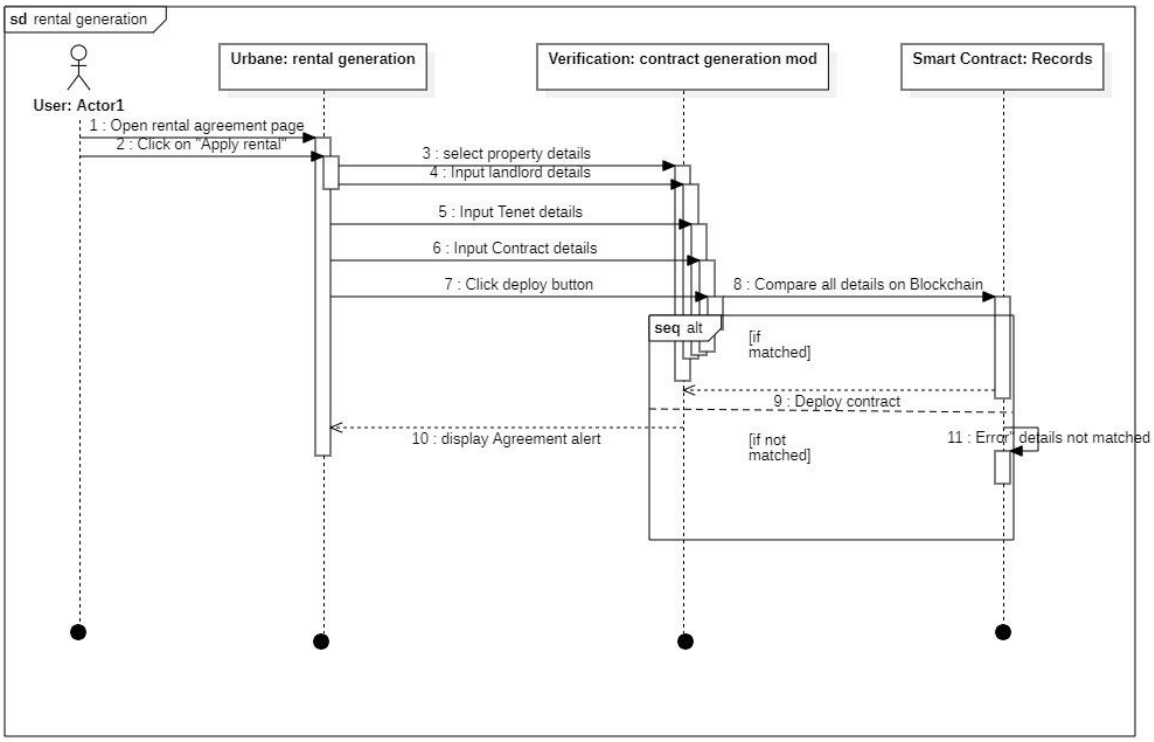

Figure 21: Sequence diagram for generating rental agreement

The figure shows the sequence diagram for generating a rental agreement between a landlord and tenant using smart contract

5.7 Policies and Tactics

The developed system will be centered upon the following policies:

- The users and admins will have to formally join and register themselves upon the used blockchain by making a wallet ID.
- Wallet IDs is a unique aspect of the web 3 framework, thus, the users and admins themselves are advised to guard their respective IDs.
- Only the admins have the access to upload registry documents, to remove the possibility of tampered documents from the user's end.
- The system will be using digital signatures and hash functions to identify users and generate hash numbers respectively.
- The system, in the initial phases, will be catering primarily to local land registry data. Although, the team aims to expand this to further levels on a technical and industrial level.

To ensure that the policies run smoothly, the tactics will be listed as:

5.7.1 Documentation

It is advised for all users to read and understand the documentation of this system for a primary understanding of the topic. This documentation is developed by the team and has been run parallel to the development process. This makes it extremely crucial to the work and usage of the system.

5.7.2 Tools Used

For technical audience or any personnel eager to learn more, it is advised to go through the formal documentation of each and every tool that is being employed in the development and implementation of this system.

5.7.3 Algorithms in use

It must be noted that blockchains are dependent upon Proof-of-stake (PoS) and Proof-of-Transfer (PoX) algorithms at the core, which give power to users in accordance with the storage and financial resources they allocate respectively.

Chapter 6: Implementation and Test Cases

The implementation of the project will cover a number of aspects in different dimensions. It is inclined towards the actual implementation, but establishes the framework of functionalities that is needed in the implementation of the whole project.

6.1 Implementation

Following are the components of the system which have been implemented in the prototype.

6.1.1 Decentralized Storage

Dealing with the storage and retrieval of the files from the decentralized storage medium, this functionality is responsible for the control of all the documents linked to users and properties. The admin shall have the power to upload the original files and verify the ownership of the user that has requested it. The system shall treat each property and owner as separate entities to keep the functionalities clearly aligned and not allow any overlapping. Each property object will hold the link to its NFT (Non-Fungible Token). The NFT shall be stored on the decentralized storage and minted onto the blockchain. This procedure will ensure uniqueness and stop any duplication, as NFTs are unique in their existence.

6.1.2 Filing

Dealing with the storage and retrieval of the files from the decentralized storage medium, this functionality is responsible for the control of all the documents linked to users and properties. The admin shall have the power to upload the original files and verify the ownership of the user that has requested it. The system shall treat each property and owner as separate entities to keep the functionalities clearly aligned and not allow any overlapping. Each property object will hold the link to its NFT (Non-Fungible Token).

The NFT shall be stored on the decentralized storage and minted onto the blockchain. This procedure will ensure uniqueness and stop any duplication, as NFTs are unique in their existence.

6.1.3 Non-Fungible Token Generation and Minting

A non-fungible token (NFT) is the fundamental building block of blockchain-based decentralized record filing. It is an asset that only one person owns, and unless transferred by the owner, it cannot be owned by another person. By minting a hash of the record, a link to the actual record, the address of the owner, and the public key on the blockchain as metadata, the project successfully enables an authorized administrator to generate Non-Fungible Tokens of records.

Using a blockchain wallet, an administrator can mint these NFTs with ownership given to an owner. To publish the token to the blockchain, a smart contract had to be written in stacks clarity language. The smart contract transaction must be called by approved addresses or principals of admins.

6.1.4 Record Owner Verification Through Cryptography

Secure communication is necessary for any Internet-based verification, and cryptography makes this possible. MongoDB, a cloud-based database, was used to put this feature into action. The application depends on MongoDB Atlas to store the proprietor's wallet address and metadata of property records like plot number, record type, and so forth. The metadata

associated with each record created by an administrator on the blockchain is also saved on MongoDB and the smart contract.

The application automatically searches the token ID (to search the NFT on the blockchain) of the respective record from the database that formed a digital signature by encrypting the record hash with their private key when a user wants to verify the owner of the property. All the user has to do to verify the owner of the property is select the record from the drop-down menu and add the data that the user has, such as the owner's CNIC, the owner's wallet address, etc.

The token ID is then used by the verifier module to retrieve the blockchain metadata and decrypt the hash to compare it to the blockchain hash. If it matches verification prompt pop-ups otherwise not verified error message rises. MongoDB is used to store metadata during this process, with the data being automatically added by the minting module and only read by the verifier module. Through cryptography, it is guaranteed that the minter is the administrator and approved to do this.

6.1.5 Record Ownership Transfer Through Blockchain

If an existing owner of the record wants to transfer the ownership of the record to another person. The new owner must have the blockchain wallet on the same blockchain. The owner initiates the transfer of the property by selecting the property in its ownership. This happens through the logged-in user session details.

When a user logged in, we store the wallet address of the user and search it against the properties minted on the blockchain to see how many and which property is owned by this wallet address. Once the owner of the property is authenticated and the property to be transferred is selected, the owner adds the details of the new user such as the wallet address of the new user, associated CNIC, etc.

Our application matches logged in user’s public key with the public key of the record owner to be transferred (stored on the blockchain) to check if the user that initiates the transfer request and the owner of the property are the same and also checks if the new owner’s address entered by the user is valid and existed on the blockchain. If all the checks return true it allows the user to make an NFT ownership transfer transaction to transfer the record ownership to a new owner.

6.2 Test Case Design and Description

Following are the test cases that were designed and conducted in order to check the system for errors and irregularities.

6.2.1 User Login Test Case

Urbane User			
Main Page			
Test Case ID:	*1*	**QA Test Engineer:**	*Ahmad*
Test case Version:	*1.0*	**Reviewed By:**	*Hamza*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Login*
Revision History:	-		
Objective	*Test the login functionality*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser, Internet Connection*		
Assumptions:	-		
Pre-Requisite:	*The user is on the login page*		

Step No.	**Execution description**	**Procedure result**
1	*Username and password are entered.*	*Dashboard is opened for each user.*

Comments: -

☒ *Passed* ☐ *Failed* ☐ *Not Executed*

6.2.2 User Login Test Case Alternate Scenario

Urbane User			
Main Page			
Test Case ID:	*2*	**QA Test Engineer:**	*Ahmad*
Test case Version:	*1.0*	**Reviewed By:**	*Hamza*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Login*
Revision History:	-		
Objective	*Test the login functionality when incorrect credentials are provided.*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web browser and a working internet connection*		
Assumptions:	-		
Pre-Requisite:	*User is on the login page.*		

Step No.	**Execution description**	**Procedure result**
1	*Username and password are entered.*	*Wrong credentials error message is shown*

Comments: -

☒ *Passed* ☐ *Failed* ☐ *Not Executed*

6.2.3 User Log Out Test Case

Urbane User			
Main Page			
Test Case ID:	*3*	**QA Test Engineer:**	*Ahmad*
Test case Version:	*1.0*	**Reviewed By:**	*Hamza*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Logout*
Revision History:	-		
Objective	*Test the Logout Functionality*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	-		
Pre-Requisite:	*User is already logged in.*		

Step No.	**Execution description**	**Procedure result**
1	*Press Sign Out Button*	*User is signed out and redirected to main landing page*
Comments: -		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.2.4 User Sign Up Test Case

Urbane User			
Main Page			
Test Case ID:	*4*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Sign Up*
Revision History:	-		
Objective	*Test the addition of a new Stacks wallet*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	*The user has a valid email address to create a new Stacks wallet account.*		
Pre-Requisite:	*User is on Sign Up Page*		

Step No.	**Execution description**	**Procedure result**
1	*Enter: Email, Password*	*Stacks API will request email verification*
2	*Open Email Account* *Verify the provided email address*	*User is successfully registered and given a wallet principal with key pair (Public/Private)*
Comments: -		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.2.5 Uploading Record to Decentralized Storage Test Case

Urbane User			
Dashboard			
Test Case ID:	*5*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Record Uploading on Decentralized Storage*
Revision History:	-		
Objective	*File Uploading on Decentralized Storage*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	-		
Pre-Requisite:	*User is logged into his account*		

Step No.	**Execution description**	**Procedure result**
1	*Press the "File Upload" tab.*	*The system will redirect the user to file uploading page.*
2	*User attaches the file, input file details and press "Upload" button.*	*User attaches the file, input file details and press "Upload" button.*
Comments: -		
☒*Passed* ☐*Failed* ☐*Not Executed*		

6.2.6 Reveal Wallet Principal Test Case

Urbane User			
Dashboard			
Test Case ID:	*6*	**QA Test Engineer:**	*Hamza*
Test case Version:	*Version number*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Fetch Wallet Info*
Revision History:	-		
Objective	*Fetch wallet principal*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	-		
Pre-Requisite:	*User is logged into his account*		

Step No.	**Execution description**	**Procedure result**
1	*Press the Reveal Wallet Principal Button*	*The stx (stacks wallet) principal is revealed*
Comments: -		
☒*Passed* ☐*Failed* ☐*Not Executed*		

6.2.7 Minting a Record Test Case

Urbane User			
Dashboard			
Test Case ID:	*7*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Minting Record as NFT*
Revision History:	-		
Objective	*Minting a record using an authorized principle or admin.*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser, Working Internet Connection*		
Assumptions:	-		
Pre-Requisite:	*The user must be logged into the application.*		

Step No.	**Execution description**	**Procedure result**
1	*Provide record and the principal (wallet address) of the new owner and press the "Mint" button*	*The system will open the transaction confirmation window*
2	*Accept the transaction*	*The system will call the smart contract and make the transaction which will succeed*
Comments: -		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.2.8 Minting a Record Test Case Alternate Scenario 1

Urbane User			
Dashboard			
Test Case ID:	*8*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Minting Record as NFT*
Revision History:	-		
Objective	*Minting Record as NFT using an unauthorized principle not an admin*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	-.		
Pre-Requisite:	*The minter must be logged into the application*		

Step No.	**Execution description**	**Procedure result**
1	*Provide record and the principle (wallet address) of the new owner and press the "Mint" button*	*The system will open the transaction confirmation window*
2	*Confirm the transaction*	*The transaction will fail and return the "Unauthorized Principal Error"*
Comments: -		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.2.9 Minting a Record Test Case Alternate Scenario 2

Urbane User			
Dashboard			
Test Case ID:	*9*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Minting Record as NFT*
Revision History:	-		
Objective	*Minting a record with insufficient STX tokens (Transaction fee)*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	*The minting authority is a recognized admin.*		
Pre-Requisite:	*The minter must be logged into the application*		

Step No.	**Execution description**	**Procedure result**
1	*Provide record and the principal (wallet address) of the new owner and press the "Mint" button*	*The system will check for available STX tokens and return error stating "Insufficient Balance"*
Comments: -		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.2.10 Cancelling Minting Record Test Case

Urbane User			
Dashboard			
Test Case ID:	*10*	**QA Test Engineer:**	*Hamza*
Test case Version:	*Version number*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Minting Record as NFT*
Revision History:	-		
Objective	*Canceling Minting a Record*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	-		
Pre-Requisite:	*The admin is logged in and has pressed "Mint Record" button.*		

Step No.	**Execution description**	**Procedure result**
1	*Press the "Cancel" button*	*The system will cancel the minting of the record*
Comments: -		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.2.11 View List of Owned Records Test Case

Urbane User			
Dashboard			
Test Case ID:	*11*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Fetch Wallet Information*
Revision History:	-		
Objective	*View all the records that an owner own.*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	*The user owns at least one record*		
Pre-Requisite:	*The user is logged in the application*		

Step No.	**Execution description**	**Procedure result**
1	-	*Upon logging in to the application, the user will be presented with a list of the records that they own*

Comments:

☒ *Passed* ☐ *Failed* ☐ *Not Executed*

6.2.12 Viewing a Record in PDF Format Test Case

Urbane User			
Dashboard			
Test Case ID:	*12*	**QA Test Engineer:**	*Ahmad*
Test case Version:	*1.0*	**Reviewed By:**	*Hamza*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Retrieve record from decentralized storage*
Revision History:	-		
Objective	*View the record in PDF format*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	*The record has been uploaded to decentralized storage*		
Pre-Requisite:	*The user is logged in the application*		

Step No.	**Execution description**	**Procedure result**
1	*Press the "Retrieve" button on a record from the list of owned records*	*The system will open an overlay with the details of the record in pdf format*

Comments: -

☒ *Passed* ☐ *Failed* ☐ *Not Executed*

6.2.13 Record Owner Verification Case

Urbane User			
Dashboard			
Test Case ID:	*13*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Verify Record Owner*
Revision History:	-		
Objective	*Verifying whether a presented owner is the valid owner of the record*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser, Working Internet Connection*		
Assumptions:	*The user requests for ownership verification*		
Pre-Requisite:	*The user is logged in the application*		

Step No.	**Execution description**	**Procedure result**
1	*Click on the "Ownership Verification" button*	*The system will redirect the user to verification page*
2	*Choose record to verify, enter the credentials and click on the "Verify" button*	*The system will fetch metadata of NFT presents on the blockchain. If they match with the one user entered, verification succeeds and Valid prompt is shown, else Invalid is shown.*
Comments: -		
☒*Passed* ☐*Failed* ☐*Not Executed*		

6.2.14 Record Owner Verification Case Alternative Scenario

Urbane User			
Dashboard			
Test Case ID:	*14*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Verify Record Owner*
Revision History:	-		
Objective	*Verifying whether a presented owner is the valid owner of the record*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser, Working Internet Connection*		
Assumptions:	*The user requests for ownership verification*		
Pre-Requisite:	*The user is logged in the application*		

Step No.	**Execution description**	**Procedure result**
1	*Ownership Verification button is clicked.*	*System redirects user to verification page.*
2	*Choose record to verify, enter the credentials and click on the "Verify" button*	*If the fetched metadata matches with the one user entered, verification succeeds and Valid prompt is shown, else Invalid is shown.*
Comments:		
☒*Passed* ☐*Failed* ☐*Not Executed*		

6.2.15 Encrypting Hash of Record Test Case

Urbane User			
Dashboard			
Test Case ID:	*15*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Verifiable Proof for Verification*
Revision History:	-		
Objective	*Admin while minting the record as NFT must also send the hash of the record data encrypted with the owner's key for authentication*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser, Working Internet Connection*		
Assumptions:	*The admin is authorized.*		
Pre-Requisite:	*The user is logged into the application*		

Step No.	**Execution description**	**Procedure result**
1	*Press the "Mint" button on the record to be shared.*	*The system will show multiple text fields to add the address and details of the owner.*
2	*Visually verify the record details and click the "Confirm" button.*	*The system will pop up a transaction box to make transaction*
3	*Input a valid address and details and click mint.*	*The front-end will compute hash of the file and give successfully minted message.*
Comments:		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.2.16 Record Ownership Transfer Test Case

Urbane User			
Dashboard			
Test Case ID:	*16*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Testing Team lead*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Relation to use cases*
Revision History:	-		
Objective	*Transferring Ownership of a record*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser, Working Internet Connection*		
Assumptions:	-		
Pre-Requisite:	*The user must be logged into the application*		

Step No.	**Execution description**	**Procedure result**
1	*Provide record and the principle (wallet address) of the new owner and press the "Transfer" button*	*The system will verify the owner principal of the record and check if the new owner principal is valid and will open the transaction confirmation window*
2	*Accept the transaction*	*The system will call the smart contract and make the transaction which will succeed*
Comments: -		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.2.17 Usability Test Case

Urbane User			
Dashboard			
Test Case ID:	*17*	**QA Test Engineer:**	*Hamza*
Test case Version:	*1.0*	**Reviewed By:**	*Ahmad*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Return Valid Results to Query*
Revision History:	-		
Objective	*To check if the system is easily usable or not*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	*The user knows about the functionalities of the application*		
Pre-Requisite:	*The user is logged into the application*		

Step No.	**Execution description**	**Procedure result**
1	*The user navigated through the application using intuition to guess what each button does*	*The system behaves accordingly with the expectations of the user*
Comments: -		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.2.18 Performance Test Case

Urbane User			
Dashboard			
Test Case ID:	*18*	**QA Test Engineer:**	*Ahmad*
Test case Version:	*1.0*	**Reviewed By:**	*Hamza*
Test Date:	*04/21/2023*	**Use Case Reference(s):**	*Performance*
Revision History:	-		
Objective	*Checking if the system is performing according to expectations and in a user-friendly way*		
Product/Ver/Module:	*Urbane – 1.0*		
Environment:	*Web Browser* *Working Internet Connection*		
Assumptions:	*The user knows about the functionalities of the application*		
Pre-Requisite:	*The user is logged into the application*		

Step No.	**Execution description**	**Procedure result**
1	*The user clicks on multiple buttons*	*All the buttons react in a quick and consistent way.*
Comments: -		
☒ *Passed* ☐ *Failed* ☐ *Not Executed*		

6.3 Test Metrics

The test metrics are as follows.

6.3.1 Functionality

The “functionality” test metric has been described in the table below.

Table 3: Functionality Test Metric

The following table provides the details of the Functionality test metric used for testing the system.

Functionality	**Check if all the core functionalities of the system are working as intended.**
Number of Test Cases	16
Number of Test Cases Passed	16
Number of Test Cases Failed	0
Test Case Defect Density	0
Test Case Effectiveness	18.75

Traceability Matrix

Test ID	**Requirement**	**Implemented Feature**
1	User Authentication	User Login
2	User Authentication	User Login
3	User Authentication	User Logout
4	User Authorization	User Sign Up
5	Uploading files to decentralized storage.	Decentralized File Upload
6	Integration of crypto wallet	Hiro Wallet Integration
7	Trade property using blockchain	NFT Minting of Property
8	Trade property using blockchain	NFT Minting of Property
9	Trade property using blockchain	NFT Minting of Property
10	Trade property using blockchain	NFT Minting of Property
11	Admin’s access to ownership information.	List of owned records.
12	Retrieving files from decentralized storage	Decentralized File Retrieval
13	Verify the claim of ownership of a property.	Smart Contract Based Ownership Verification
14	Verify the claim of ownership of a property.	Smart Contract Based Ownership Verification
15	Trade property using blockchain	NFT Minting of Property
16	Trade property using blockchain	Smart Ownership Transfer

6.3.2 Usability

The “functionality” test metric has been described in the table below.

Table 4: Usability Test Metric

The following table provides the details of the Usability test metric used for testing the system.

Usability	**Testing the system for usability and user experience.**		
Number of Test Cases	1		
Number of Test Cases Passed	1		
Number of Test Cases Failed	0		
Test Case Defect Density	0		
Test Case Effectiveness	50		
Traceability Matrix	**Test ID**	**Requirement**	**Feature**
	17	The system shall be easy to use and understand	GUI based web application.

6.3.3 Performance

The performance test metric has been described in the table below.

Table 5: Performance Test Metric

The following table provides the details of the Usability test metric used for testing the system.

Performance	**Check the performance of the system.**		
Number of Test Cases	1		
Number of Test Cases Passed	1		
Number of Test Cases Failed	0		
Test Case Defect Density	0		
Test Case Effectiveness	100		
Traceability Matrix	**Test ID**	**Requirement**	**Feature**
	18	The system shall be efficient and work correctly under stress.	Three tier architecture based decentralized application.

Chapter 7: User Manual

This section provides guidelines and instructions for the proper use of the system that has been developed and shall serve the purpose of a user manual. The manual contains instructions ranging from the installation and setup of local development environment to the usage of the decentralized storage and smart contracts depending on the role of the user.

7.1 Installation & Setup

In order to launch the system locally for development purposes, following prerequisites & dependencies must be installed on the computer.

- Node JS
- Clarinet
- Mozilla Firefox/Google Chrome
- Hiro Wallet extension for Firefox/Chrome
- Docker
- MongoDB

After the dependencies mentioned above have been installed, extract the archive containing the source files in a folder and follow the procedure mentioned below for setting up the two components of the system i.e., the web application and the local blockchain simulation (referred to from here on as “devnet”).

7.1.1 Web Application

The web application consists of two subcomponents: the frontend and the backend. To install the packages of the frontend application, open a Command Prompt as administrator and change the directory to the folder in which the archive containing the source files has been extracted. Now change the directory to the “frontend” folder by entering the command “cd frontend”.

The details of all the packages required for the frontend application are stored in the “package.json” file. To install these packages, simply enter the command “npm install” and wait for the process to finish.

After the process has finished, simply enter the command “npm start” to launch the frontend of the web application. Although the frontend of the application is ready to use, it still needs to be connected to the backend of the web application in order to perform its functionalities.

To setup the backend of the web application, launch another instance of command prompt as admin and move back the working directory to the root folder and change the directory to the folder named “backend”. The procedure to install the dependencies is the same as the frontend. Enter the command “npm install” and wait for the process to finish.

Before running the backend application, you need to install a package called “nodemon” globally. To do so enter the command “npm install -g nodemon” and wait for the process to finish. After completion of the process, enter the command “nodemon server.js” to launch the backend application.

7.1.2 Blockchain Simulation/Devnet

To host the smart contracts, a local simulation of the Stacks blockchain is required to provide a runtime environment. To install this local simulation called the “devnet”, launch docker and

open a command prompt instance as administrator and change the working directory to “frontend\smart\”. Now enter the command “clarinet integrate”. This will initiate a process to create docker images of all the services required to run the local devnet. The progress of the process will be shown on the command prompt and the devnet will have started successfully after the red or yellow blocks indicating the progress of the installation of the services have turned green.

7.1.3 Hiro Wallet

The Hiro wallet extension is essential for deploying the smart contracts on the devnet and also for calling the smart contract functions which enable the NFT minting and ownership transfer functionalities of the application. Before configuring the Hiro Wallet, make sure that the docker instances are running by using the “clarinet integrate” command in a command prompt opened in the frontend\smart\ directory. Now open the browser and click on the Hiro wallet extension from the extensions menu of the browser.

This will open the Hiro wallet window, select the dropdown menu indicated by three dots in the upper right corner of the window. Click on “Change Network” and select “Devnet”. This will change the Hiro wallet’s network to the local simulation of the Stacks blockchain.

As this is a simulation, a dummy account is required which is authorized to deploy the smart contracts and invoke the functions stored in it. The details of this account are provided in the “Devnet.toml” file in the directory “frontend\smart\settings\” after the line having written “[accounts.deployer]”. These details can be entered into the Hiro wallet to use the local deployer account which is required by the admin user to mint properties as NFTs as well as transfer ownership of properties using smart contracts.

7.2 User Accounts

There are two types of accounts that the application has, specifically the applicants and administrators. Applicants have the rights to search for documents related to real estate, verify ownership of a property and submit requests for ownership transfers, whereas the administrators can also initiate ownership transfers upon the request of applicants as well as mint properties as NFTs.

When the web application is launched, the user is provided the login screen. If the user is already registered, he can enter his credentials to log into the system. If there is no account, one can be created by using the sign-up page by clicking “Sign Up” link below the “Login” button.

Admin accounts can only be created (converted from “applicant” to “admin”) manually by accessing the MongoDB database and changing the user role field by a database administrator.

7.3 Wallet Connection

When a user logs into the application, he is redirected to the homepage. On the homepage, on the upper right corner there is a button named “Connect Wallet”. Upon clicking the button, the Hiro Wallet extension asks the user to connect the wallet to the application for use with the Stacks blockchain. This connection is necessary to make in order to communicate with the blockchain. After the wallet has successfully been connected, the user can move to the specific service he intends to use by selecting from the “Services” dropdown menu of the navbar.

7.4 Retrieving Files from Decentralized Storage

There are five types of documents that the application stores for a certain real estate plot. These documents can be retrieved from the document retrieval page, which can be accessed from the sidebar menu or the navbar if you are on the homepage.

To retrieve a document of a plot of land, select its area, block and plot number of the plot from the selection menu as well as the type of document and click on the “Search” button.

The document will be retrieved from the decentralized storage and displayed below the selection menu.

7.5 Uploading Documents to the Decentralized Storage

To upload documents to the decentralized storage, the user must be an admin. This functionality is provided by the “Upload Document” page which can be accessed through the sidebar navigation or the navbar of the homepage under the “Services” dropdown menu.

The “Upload Document” page requires the information about the property which can be provided used the selection menu on the page, similar to the one on the document retrieval page. After choosing the area, block, plot number and the document type, attach the document file by clicking the “Browse” button and click on the “Upload” button.

7.6 Ownership Verification

The ownership verification can be performed from the “Verify Ownership” page which can be accessed through the sidebar navigation or the navbar from the homepage. To verify ownership of a property, select its area, block and plot from the selection menu.

After its location has been selected, enter the information of the person claiming to be the owner of a property such as his first and last name along with his CNIC and click on the “Verify” button. If the ownership details match, a prompt will be displayed containing the message “Ownership Verified Successfully”, otherwise a message will be displayed stating “Ownership Verification Failed”.

7.7 Ownership Transfer Request

This request can only be generated by the owner of a property which needs to be registered as an applicant. The ownership transfer request can be generated through the “Generate Transfer Request” page requiring the location of the property as well as the CNIC of the owner and the buyer/receiver of the property and clicking on the “Generate Request” button.

If the user’s profile matches that of the owner of the property and his wallet address is the same as found in the decentralized ledger, a message alert stating “Request Generated Successfully” will be displayed, otherwise an error message stating “Owner Does Not Match” is displayed to the user.

7.8 Ownership Transfer

This functionality is only available to the admin. This functionality is accessible from the “Ownership Transfer” page from the sidebar navigation or the homepage navbar. All the ownership transfer request are displayed to the admin and he can approve the transfer by clicking “approve” on each transfer request.

Chapter 8: Conclusion and Future Work

8.1 Summary & Conclusion

This project is inclined towards solving the record filing problem in general and finds specific application in land record storage and verification. The work done so far allows the user to sign up on the system with user and/or admin roles, store and retrieve files on the decentralized storage, request verification of their desired documents, transfer ownership of property and mint NFTs of documents.

Considering the filing as an integral part of the project, it is connected to all components. Connecting the user to the system, recording the credentials and keeping track of the storage and retrieval of files has been done using decentralized storage, smart contracts have been developed and connected backend and front-end tools.

The functionality of the rental agreement was kept tentative from the very start of the project. The team intends to fully develop and test the app as a whole before starting the work on the rental agreement contract. Currently, the app is a working single entity, with all functionalities connected and working in coherence. Deployment on the Dev net has been done and the deployed app is being tested

8.2 Future Work

The following work can be done in order to further enhance and improve the project.

8.2.1 Improved Graphical User Interface

The graphical user interface can be made more user friendly in order to enhance the user experience.

8.2.2 Rental Agreement

A smart contract for rental agreements can be created in order to automate paying and receiving rent and managing the lease of the property according to the agreed upon duration.

8.2.3 Deployment of the Application on Public Network

The application can be deployed on a cloud service and its blockchain module can hosted on the Stacks public blockchain to make the application ready for public use.

8.2.4 Digitization Module

A module can be created for digitizing the hard copies of the documents through scanner and uploading them into the decentralized storage. Currently only the storage and retrieval has been implemented and it is assumed that the targeted organization has already digitized its data.

References

[1] Muhammad Irfan Khalid, Jawaid Iqbal, Ahmad Alturki, Saddam Hussain, Amerah Alabrah, Syed Sajid Ullah, "Blockchain-Based Land Registration System: A Conceptual Framework", Applied Bionics and Biomechanics, vol. 2022, Article ID 3859629, 21 pages, Feb 2022.

[2] Krishnapriya S, Greeshma Sarath, “Securing Land Registration using Blockchain”, Procedia Computer Science, Volume 171, Pages 1708-1715, June 2020.

[3] LeewayHertz, “Blockchain Land Registry Platform – Reducing Frauds and Delays”.

[4] Gateway Digital, “Blockchain based Land Registry Platform”.

[5] Alexandru Oprunenco, Chami Akmeemana, “Using blockchain to make land registry more reliable in India”, https://www.undp.org/blog/using-blockchain-make-land-registry-more-reliable-india, May 2018.

[6] R. Yu, Z. Wang, C. Zhang, and S. Guan, "A secure blockchain-based housing rental platform," 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2021, pp. 2049-2053, July 2021.

[7] Midasium, “SMART TENANCY CONTRACTS- The Blockchain of Real Estates”.

[8] Rajit Nair, S.N. Zafrullah, P. Vinayasree, Prabhdeep Singh, M.M.A. Zahra, T. Sharma and Fardin Ahmadi, Blockchain-Based Decentralized Cloud Solutions for Data Transfer, 2022.

[9] Kumar Shrestha, Julita Vassileva and Ralph Deters, A Blockchain Platform for User Data Sharing Ensuring User Control and Incentives, 2020.