ASHESI UNIVERSITY

LAND MANAGEMENT SYSTEM USING BLOCKCHAIN - ASAASE ABAN

APPLIED CAPSTONE

B.Sc. COMPUTER SCIENCE

AGYEI EMMANUEL

2025

ASHESI UNIVERSITY

LAND MANAGEMENT SYSTEM USING BLOCKCHAIN - ASAASE ABAN

APPLIED CAPSTONE

APPLIED CAPSTONE submitted to the Department of Computer Science & Information Systems, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Science.

AGYEI EMMANUEL

2025

DECLARATION

I hereby declare that this APPLIED CAPSTONE is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature: .... Candidate's Signature: ....

Candidate's Name: .... Candidate's Name: ....

Date: .... Date: ....

I hereby declare that preparation and presentation of this APPLIED CAPSTONE was supervised in accordance with the guidelines on supervision of APPLIED CAPSTONE laid down by Ashesi University.

Supervisor's Signature:
....

Supervisor's Name:
....

Date:
....

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Mr. Charles Adjectey, for his invaluable guidance, encouragement, and support throughout the development of this project. His insights and mentorship were instrumental in shaping the success of this work.

I am deeply grateful to my father, Mr.Jerry Agyei, whose unwavering support and belief in me have been a constant source of motivation.

A special acknowledgement goes to my grandfather, Nana Addo Kweku II, whose knowledge and experience in land matters greatly enriched my understanding of the topic and provided an important foundation for this project.

I would also like to acknowledge my friends Victor Quagraine, Oheneba Dadee, Pius Dumfour-Toku, Barbara Anku, Chelsea Owusu, Maame Adjoa Abankwa, Emmanuel Osei Kesse and Shadrack for their encouragement, advice, and friendship throughout this journey. Their support played a crucial role in helping me stay focused and committed to completing this project.

Thank you all for being a part of this achievement.

Abstract

The Asaase Aban project presents a blockchain-based land administration system designed to enhance transparency, security, and efficiency within Ghana's dual land governance framework. By integrating Ethereum smart contracts, a PostgreSQL/PostGIS database, decentralized file storage through IPFS, and an intuitive web interface, the system provides a robust platform for land registration, ownership verification, and property transfers. Key features include tamper-proof transaction records, dynamic geospatial visualization, and role-based access control. Comprehensive testing across smart contracts, APIs, database services, and user interfaces validated the system's functionality, security, and usability. Despite challenges like blockchain transaction delays and GeoJSON parsing issues, iterative development and user feedback informed continuous improvements. Asaase Aban demonstrates the transformative potential of blockchain technology in land governance and establishes a scalable foundation for future innovations, including mobile applications, AI-driven insights, and integration with national land registries.

Table of Contents

DECLARATION .... **III**

ACKNOWLEDGEMENTS .... **IV**

ABSTRACT .... **V**

LIST OF TABLES .... **VIII**

LIST OF FIGURES .... **IX**

CHAPTER 1: INTRODUCTION **1**

1.1 AIMS AND OBJECTIVES OF THE PROJECT 4
1.2 LITERATURE CONTRIBUTIONS AND IDENTIFIED GAPS 5
1.2.1 Related Work and Contributions *5*
1.2.2 Remaining Gaps and Research Directions *9*
1.2.3 Proposed Contribution *10*

CHAPTER 2: REQUIREMENTS **11**

2.1 REQUIREMENTS GATHERING AND ANALYSIS PROCEDURE 11
2.2 USER IDENTIFICATION 13
2.3 USE CASES ANALYSIS 13
2.4 DETAILED REQUIREMENTS 17
2.4.1. Functional Requirements *17*
2.4.2. Non-Functional Requirements *17*
2.4.3. Socio-Political Requirements *18*
2.5 ADDRESSED REQUIREMENTS 18
2.6 REQUIREMENTS ANALYSIS 19

CHAPTER 3: ARCHITECTURE AND DESIGN **22**

3.1 SYSTEM OVERVIEW 22
3.2 SYSTEM ARCHITECTURE 22
3.3 MAPPING DESIGN TO REQUIREMENTS 24
3.4 KEY MODULES AND DESIGN 26
3.5 SYSTEM SCHEMATICS 27
3.6 KEY MODULES AND DESIGN 28
3.6.1 Decentralized Ledger Module *28*
3.6.2 Smart Contract Module *29*
3.6.3 API Gateway Module *29*
3.6.4 User Interface Module *30*
3.7 SUMMARY 30

CHAPTER 4: IMPLEMENTATION **31**

4.1 INTRODUCTION ... 31
4.2 SYSTEM IMPLEMENTATION OVERVIEW ... 31
4.3 BLOCKCHAIN NETWORK SETUP ... 32
4.4 SMART CONTRACT DEVELOPMENT ... 34
4.5 API GATEWAY DEVELOPMENT ... 36
4.6 USER INTERFACE DEVELOPMENT ... 39
4.7 POSTGRESQL DATABASE INTEGRATION ... 41
4.8 DECENTRALIZED FILE STORAGE INTEGRATION ... 43
4.9 SYSTEM DIAGRAM ... 45
4.10 INTEGRATION WITH EXTERNAL SYSTEMS ... 48

CHAPTER 5: TESTING AND VALIDATION ... 49

5.1 TESTING AND VALIDATION OVERVIEW ... 49
5.1.1 Smart Contract Testing ... *49*
5.1.2 Backend API Testing ... *51*
5.2 FRONTEND-BACKEND INTEGRATION TESTING ... 52

CHAPTER 6: CONCLUSION ... 67

6.1 PROJECT SUMMARY ... 67
6.2 RECOMMENDATIONS ... 56
6.3 LIMITATIONS AND CHALLENGES ... 68
6.4 FUTURE WORK ... 68
6.5 CONCLUSION ... 68

REFERENCES ... 70

APPENDICES ... 61

List of Tables

Table 1. Browser and Device Compatibility 63
Table 3. Test Results Summary 64

List of Figures

Figure 1. blockchain-enabled framework for stool and skin lands in Ghana 6
Figure 2. A conceptual framework from Ameyaw and de Vries' paper. 8
Figure 3. Fit-for-purpose land administration framework 8
Figure 4. Use Cases 14
Figure 5. Mapping Design to Requirements 25
Figure 6. Key Modules and Design 27
Figure 7. System Design 28
Figure 8. A code snippet showing the Frontend interaction with the blockchain. 34
Figure 9. A code snippet showing the backend interaction with the smart contract 34
Figure 10. A code snippet showing the smart contract code for land title registration 36
Figure 11. Code Snippet showing how to install all dependencies for the backend. 36
Figure 12. A code snippet for the .env file for the whole setup 36
Figure 13. A code Snippet showing the API endpoint. 37
Figure 14. A code snippet of a function for getting all lands. 37
Figure 15. A code snippet of a function for getting all lands details. 38
Figure 16. A code snippet showing how a land is transferred. 38
Figure 17. A code snippet showing blockchain interaction using ethers.js 39
Figure 18. A code snippet of a database operation using the PostgreSQL client 39
Figure 19. A picture of the User interface 40
Figure 20. A picture of the Admin interface 41
Figure 21. A pciture of Postgres query and the results. 43
Figure 22. A code snippet that handles file uploads to IPFS using Web3.Storage 45
Figure 23. A code snippet showing how IPFS hash for each land document is stored. 45
Figure 24. System Interaction Diagram 47
Figure 25. A code snippet of the Smart Contract. 51
Figure 26. A code snippet showing a function that registers lands on the blockchain. 52
Figure 27. A code snippet showing the checkUserRole function 53
Figure 28. A code snippet showing a design of a registered land 54
Figure 29. A picture of registered lands 55
Figure 30. A picture of the landing page of AsaaseAban 57
Figure 31. A picture of showing which different metamask account wallet to connect to. 57
Figure 32. A picture showing the admin interface after the Admin connected the metamask wallet. 58
Figure 33. A picture showing Admin successfully verified a new land to complete the registration. 58
Figure 34. A picture showing a land of ID of 7 that is going to be transferred to the given address below it 59

Figure 35. A picture of a land that has been successfully transferred. 59
Figure 36. A picture showing the transaction of land on the blockchain using the sepolia Testnet. 60
Figure 37. A code snippet showing a challenge encountered when Geojson data was trying to retrieve data form the Postgis database. 66

Chapter 1: Introduction

National development rests on land administration as a cornerstone since land ownership, rights and transactions are structured in managed portfolios in an ordered manner. Land in Ghana is seen as a critical player in maintaining the country's economic stability and social cohesion, yet inefficiencies, inaccuracies, and corruption plague the systems surrounding the assets [1][2]. The dual land tenure system of Ghana, which comprises the customary and statutory systems of ownership, makes the administration process more complex. Ghana's land use rights are divided between approximately 80 per cent statutory land and 20 per cent customarily governed land [3]. This structure, however, also leads to governance challenges such as overlapping claims, fraudulent sales and disputes due to the discretionary power of traditional leaders in land allocation. Statutory institutions like the Lands Commission, on the other hand, have inefficiencies such as long processing times and in verifying ownership [4].

Traditional land administration systems in Ghana rely heavily on manual and centralized processes, prone to human error and lacking transparency. The absence of a reliable and immutable system for recording and verifying land transactions has fostered an environment where fraudulent practices, such as the multiple sales of the same parcel of land, thrive [3][5]. This situation undermines public trust and perpetuates tenure insecurity. The challenges are further compounded by weak coordination between customary and statutory systems, with neither adequately integrated into the broader framework of national land governance [6].

Besides operational problems, the inefficiency of Ghana's land administration system is of significant socio-economic impact. Land tenure insecurity deters investments in key sectors like agriculture and real estate, which are critical to the country's economy [3][7]. Without reliable land records, financial institutions are unsure if they can use the land as collateral and limit access to credit for farmers and small businesses. The lack of access to credit makes economic growth and poverty much more difficult to escape, especially in rural areas where land is the principal source of economic activity [8]. Like any other form of violence, the social consequences are no less severe; there are often land disputes that end up as violent conflicts. These disputes have led land guards to appear in urban areas where land values are high, and land guards are armed groups hired to enforce property claims. Not only does it erode social cohesion, but it also creates problems for national security [3].

Blockchain technology offers a transformative solution to land administration challenges by providing a decentralized, tamper-proof platform for recording transactions. Unlike traditional systems, blockchain ensures data integrity and transparency, reducing the opportunities for fraud and unauthorized alterations [4][9]. Blockchain's distributed ledger technology allows for real-time verification of land ownership and transaction history, addressing the opacity and inefficiencies of manual processes. By leveraging smart contracts, blockchain can automate complex processes such as title transfers and property subdivisions, thereby reducing transaction times and minimizing the risk of human error [10].

In Sweden, Georgia, and the United Arab Emirates, throughout the world, the potential of blockchain for land administration has been proven. However, having provided these implementations, blockchain has proved to do well in increasing the transparency, reducing fraud and ensuring public trust in government [11][12]. For example, Georgia's blockchain-based land registry has led to an improved registration process with much shorter transaction times for its stakeholders [11]. Like Sweden's pilot project has demonstrated [13], blockchain can coexist easily with today's systems to improve efficiency and reliability. These examples present an opportunity for Ghana to learn from how the country tailored its approach in balanced ways to address the country's own unique set of challenges resulting from its dual land tenure system.

Ghana's land governance landscape presents distinct challenges that necessitate a customised solution. The coexistence of customary and statutory systems creates a fragmented framework where stakeholders operate under different rules and standards [3]. Customary land management is deeply rooted in local traditions and norms, making it resistant to standardisation. Chiefs and traditional authorities often exercise significant discretion in land allocation, which can lead to disputes and a lack of accountability [5]. On the other hand, the statutory system, despite its potential for modernization, is hampered by limited technological adoption and interoperability [6]. Efforts to digitize land records, such as the Land Administration Project (LAP), have failed to address these systemic issues, leaving the sector vulnerable to inefficiencies and corruption [4].

Despite these challenges, blockchain technology provides a pathway to unify Ghana's land administration systems. By creating a decentralized platform accessible to both customary and statutory stakeholders, blockchain can bridge the gap between these parallel systems. This integration would enhance transparency, reduce conflicts, and foster trust among stakeholders [10][14]. For example, blockchain's ability to provide a tamper-proof record of transactions can help resolve disputes over overlapping claims, while its real-time verification capabilities can prevent fraudulent practices such as multiple sales. Additionally, the automation of processes through smart contracts can streamline operations, making land transactions faster and more cost-effective [9].

The inefficiencies of Ghana's current land administration system highlight the urgent need for reform. Blockchain technology offers a promising solution to address these challenges, providing a secure, transparent, and efficient framework for managing land transactions. By integrating blockchain into the land administration process, Ghana can create a system that fosters trust, reduces conflicts, and supports economic development. The proposed AssaseAban system represents a critical step towards realizing this vision, offering a modernized approach that aligns with the unique socio-political and cultural dynamics of Ghana's land governance system [1][7].

1.1 Aims and Objectives of the Project

Aim

The primary aim of this project is to develop a blockchain-based land administration system, Assase Aban, specifically designed to address the challenges of Ghana's dual land governance system. The system seeks to enhance transparency, security, and efficiency in land transactions while bridging the gap between customary and statutory land administration practices

1. **Objectives**

   Develop a Blockchain-Based Framework: Create a permissioned blockchain framework tailored to Ghana's dual land governance system, ensuring immutable, tamper-proof records for all land transactions to prevent fraud and enhance data integrity.

2. **Automate Key Land Administration Processes:** Implement smart contracts to automate critical operations such as title registration, ownership transfers, and dispute resolution, reducing delays and reliance on intermediaries.

3. **Integrate Customary and Statutory Land Systems:** Design a system that bridges the socio-political gap between customary and statutory land governance practices, fostering collaboration among stakeholders such as chiefs, landowners, and regulatory authorities.

4. **Enhance Interoperability and Scalability:** Build RESTful APIs for seamless integration with existing property systems and financial databases, ensuring real-time data access while designing for scalability to accommodate future growth.
5. **Improve Transparency and Stakeholder Trust:** Provide stakeholders, including landowners, financial institutions, and government authorities, with access to secure, transparent, and verifiable land records, thereby reducing conflicts and promoting trust in land transactions.

1.2 Literature Contributions and Identified Gaps

Blockchain technology has garnered significant attention for its transformative potential in land administration. Research has been conducted in this domain, and frameworks have been put forward to solve problems like fraud and inefficiency of information. Although these studies have facilitated the progression of blockchain applications with new understandings on certain fronts, such as the localized governance system, scalability, and socio-political integration, many gaps still need to be addressed. This section examines some significant contributions, frameworks and limitations from the literature with attention placed on the gaps that this research seeks to address.

1.2.1 Related Work and Contributions

Ali et al. [3] proposed a permissioned blockchain framework designed to create tamper-proof and immutable land records. The framework incorporated smart contracts to automate property transactions and reduce reliance on intermediaries. A RESTful API allowed real-time integration with existing property dealing systems, enhancing transparency and trust among stakeholders. The end product was a blockchain-enabled property registry system aimed at improving efficiency and reliability. However, the study primarily targeted centralized governance systems, leaving the complexities of dual governance models, such as those in Ghana, unaddressed.

Mintah et al. [2] developed a blockchain-enabled framework for stool and skin lands in Ghana, focusing on resolving issues like multiple sales and unreliable records. By integrating blockchain with customary practices, the framework enhanced transparency and real-time public access to land transaction data. While this study contributed significantly to the socio-cultural aspects of land governance, it focused narrowly on acquisition and title registration, leaving gaps in other critical areas such as land valuation and use planning.

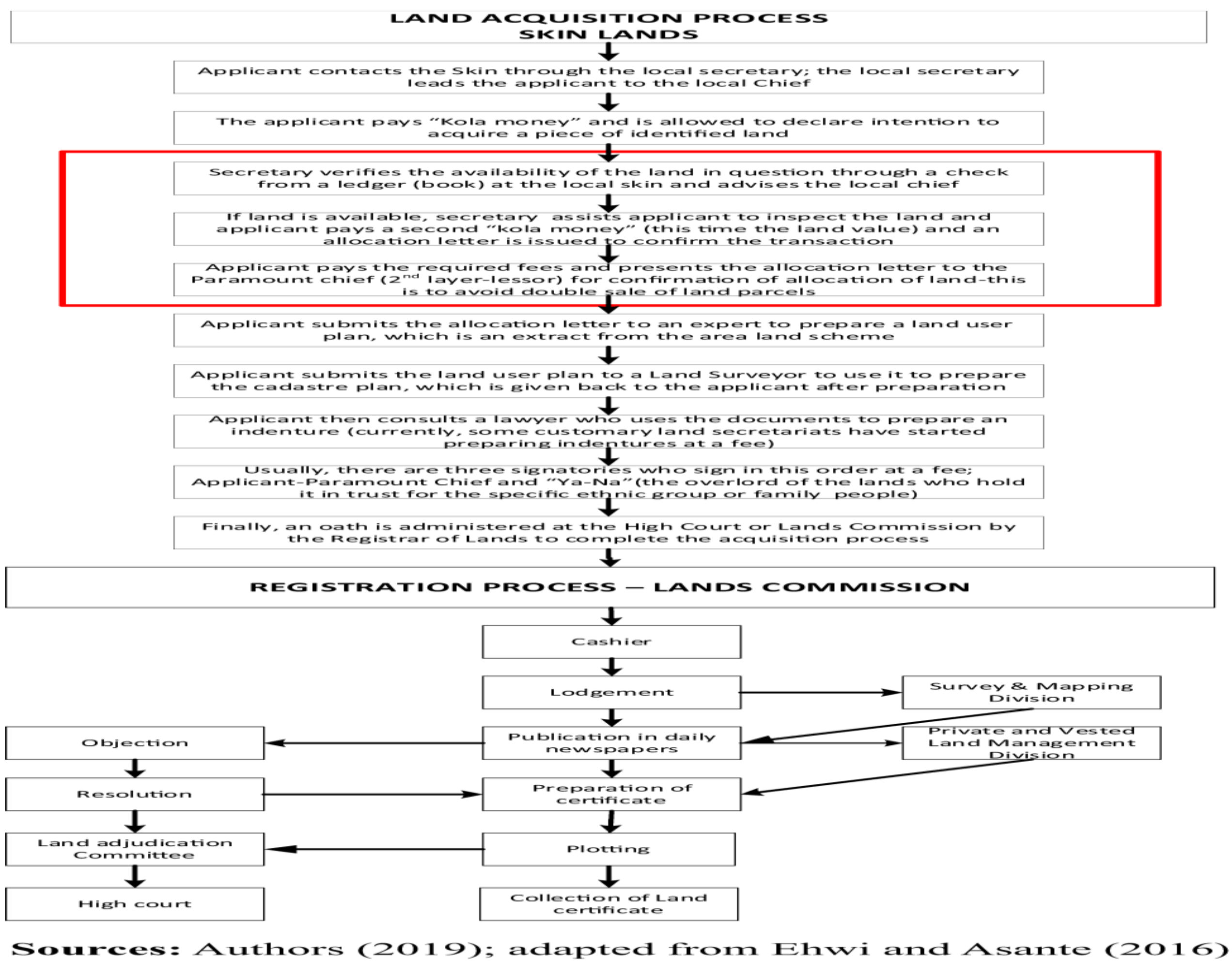

Figure 1. blockchain-enabled framework for stool and skin lands in Ghana

Kaushik [7] introduced a high-level blockchain-based framework for digitizing land records, emphasizing fraud prevention and process automation through smart contracts. The framework demonstrated scalability and efficiency in managing title transfers and property verifications. However, the study did not address the unique challenges posed by fragmented governance systems, such as those involving both customary and statutory authorities. Future research recommended exploring blockchain's adaptability to socio-cultural contexts like Ghana.

Permissioned blockchain was demonstrated by Rouhani et al. [4] to be secure, modular systems. However, their framework was not domain specific to land administration, yet they demonstrated robust, useful features such as a modular design, data integrity, resource efficiency. Their framework is flexible for decentralized implementations, and due to this, it can be a valuable reference. However, due to its lack of focus on land governance, it cannot be applied to this domain.

Ameyaw and de Vries [1] explored blockchain's potential to enhance transparency and efficiency in Ghana's land administration. Their integrative framework addressed valuation, tenure security, and land use planning, making it one of the more comprehensive studies in the field. However, the framework remained conceptual, with no detailed implementation strategy or scalability testing. They identified pilot programs needing to refine their approach and assess their practical feasibility.

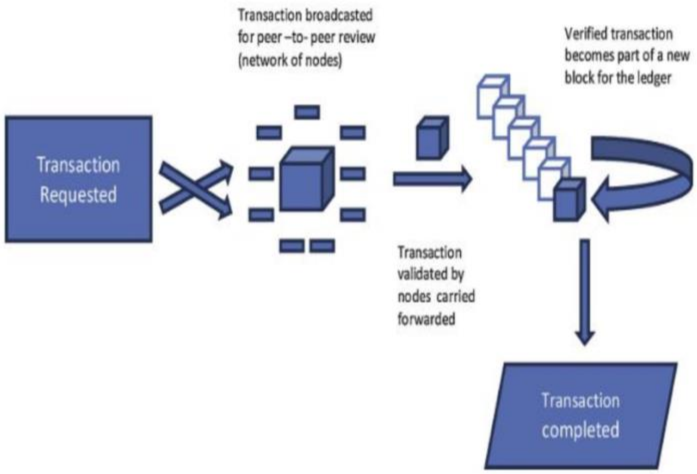

Figure 2. A conceptual framework from Ameyaw and de Vries' paper.

Ansah et al. [8] integrated blockchain technology with ISO 19152:2012 standards to develop a fit-for-purpose land administration framework. The framework aligned blockchain with international standards, emphasising transparency, automation, and process standardisation. While this approach offered insights into standardisation and global adaptability, it lacked focus on localized governance models like Ghana's, where customary systems coexist with statutory frameworks.

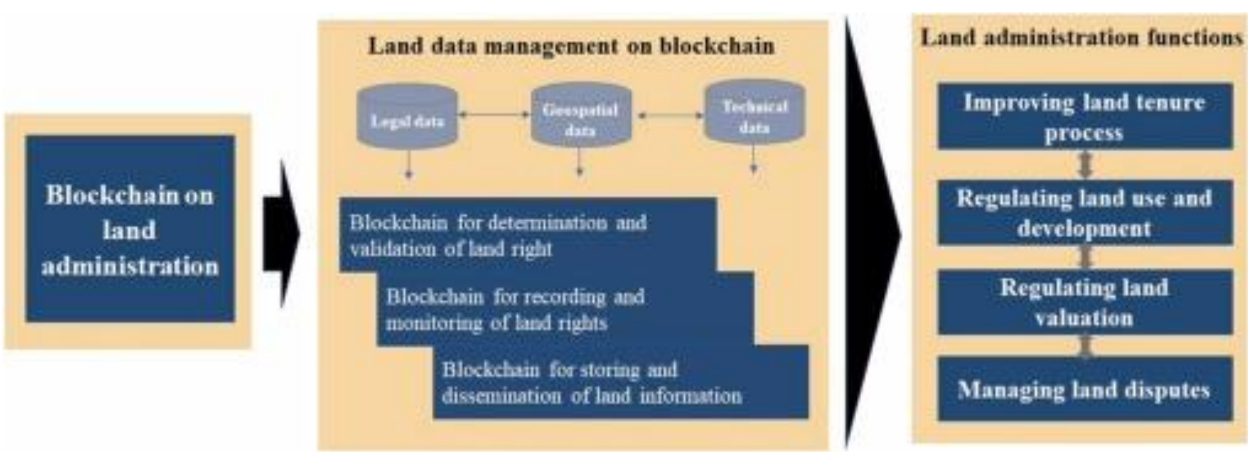

Figure 3. Fit-for-purpose land administration framework

Summary of Contributions and End Products

The literature demonstrates significant advancements in blockchain-based land administration, with notable contributions including:

Decentralized Record-Keeping: Frameworks by Ali et al. [3] and Verma et al. [5] introduced permissioned blockchains for tamper-proof land records, addressing fraud issues and enhancing stakeholder trust.

Automation through Smart Contracts: Studies by Kaushik [7] and Mintah et al. [2] emphasized using smart contracts to streamline processes such as title transfers and ownership verifications, reducing transaction times and costs.

Integration of Customary Systems: Mintah et al. [2] provided a socio-culturally sensitive framework tailored to Ghana's customary land governance, addressing challenges unique to stool and skin lands.

Global Standardization: Ansah et al. [8] highlighted the importance of aligning blockchain implementations with international standards, enhancing scalability and interoperability.

1.2.2 Remaining Gaps and Research Directions

Despite these contributions, the literature reveals critical gaps that this research seeks to address:

Integration of Dual Governance Systems: Current frameworks inadequately address the integration of customary and statutory systems, a critical challenge in Ghana's land administration.

Scalability and Implementation: Conceptual frameworks, such as those by Ameyaw and de Vries [1], require detailed implementation strategies and pilot testing to evaluate scalability and practical effectiveness.

Broadening Scope: Most studies focus narrowly on title registration and acquisition, leaving other areas like land valuation, use planning, and dispute resolution underexplored.

Socio-Political Adaptation: Few frameworks account for the socio-political complexities inherent in land governance systems, particularly in countries with pluralistic or fragmented models.

1.2.3 Proposed Contribution

Building on the reviewed literature, this research proposes a blockchain-based framework tailored to Ghana's dual governance system. It integrates blockchain technology with customary and statutory practices to create a unified, transparent, and scalable solution. The framework includes:

A Decentralized Ledger: For tamper-proof and immutable land transaction records.

Smart Contract Automation: To streamline processes such as ownership transfers and property subdivisions.

Interoperability Mechanisms: Using APIs to integrate blockchain with existing property management systems.

Scalability Testing: A pilot implementation to evaluate the framework's adaptability to Ghana's socio-political and economic contexts.

This research aims to advance blockchain-based land administration and provide a comprehensive solution tailored to Ghana's unique challenges by addressing gaps in integration, scope, and implementation.

Chapter 2: Requirements

2.1 Requirements Gathering and Analysis Procedure

The requirements gathering and analysis for this research adopted a comprehensive, multi-dimensional approach to ensure that the proposed blockchain-based land administration framework adequately addresses the unique challenges of Ghana's dual land governance system. This process involved the following key steps:

Stakeholder Analysis: The first step involved the identification and categorization of critical stakeholders, including chiefs, landowners, statutory authorities such as the Lands Commission, property developers, and financial institutions. Each stakeholder group's needs, challenges, and expectations were systematically documented to inform the requirements process. Prior research [1][2], as well as informal consultations, significantly contributed to understanding stakeholder perspectives, particularly regarding inefficiencies in land transaction processes and trust deficits within the existing system.

Literature Review: The review involved an exhaustive study of the work that has been achieved in the realm of land administration concerning current challenges and technological interventions. Mintah et al. [3] and Ameyaw and de Vries [1] showed that the persistent problems of land multiple sales, unreliable record keeping and disjointed operation of customary and statutory land governance frameworks persist. This was further in line with critiquing the technical remedies proposed in prior literature, such as adopting decentralised ledgers and smart contracts. Additionally, several scholars have expanded the conversation regarding blockchain's capacity to augment tenure security and agricultural financing in rural African settings [15] or to develop blockchain solutions with a potential for large-scale use in rural land registry administration in developing countries [16]. With these recent insights, the growing applicability of the blockchain was also emphasised beyond mere record keeping, for the blockchain to be empowered to more people, and govern more.

Focus Groups and Informal Interviews: Customary authorities, real estate developers, land commission officials, and financial sector representatives were interviewed and engaged in focused group discussions. It provided operational inefficiencies under the current practices, socio-cultural aspects of land transactions and perceived feasibility and acceptability of blockchain-based systems by end users. Using participatory techniques in this discussion allowed us to produce contextually rich data that supplements those findings from the literature review.

Use Case Analysis: Representative scenarios were developed to model typical land administration processes such as title registration, land transfers, dispute resolution, and property valuation. The analysis of these scenarios was informed by frameworks proposed in prior studies, including those by Verma et al. [4] and Kaushik [5], which emphasized the critical role of smart contracts in automating land administration workflows. Additional consideration was given to ensuring that these use cases reflect the complexities of dual governance systems, as advocated by more recent blockchain applications in similar developing country contexts [15][16].

Feasibility Study: A detailed feasibility study was undertaken to assess both the technical and socio-political viability of implementing a blockchain-based system within Ghana's land governance environment. Drawing on insights from earlier works [6][7] and recent evidence from applications in Nigeria, Benin, and other developing economies [15][16], the study evaluated factors such as scalability, interoperability with existing land management systems, stakeholder adoption likelihood, and legal compliance with customary and statutory land laws. Emphasis was placed on designing a system flexible enough to adapt to socio-political realities while maintaining the core principles of transparency, security, and efficiency inherent in blockchain technology.

2.2 User Identification

The proposed system involves diverse users with varying needs, including:

- **Customary Authorities (Chiefs and Elders):**

  Responsible for managing stool and skin lands, these users require tools for secure record-keeping, real-time transaction verification, and conflict resolution.

- **Statutory Authorities (Lands Commission):**

  Tasked with regulating land transactions, this group requires a unified system to integrate customary records with statutory databases.

- **Property Developers and Real Estate Agents:**

  These users need access to verified land records to facilitate secure transactions and reduce fraud.

- **Financial Institutions:**

  Require verifiable land ownership data to assess creditworthiness and provide loans against collateralized property.

- **General Landowners and Buyers:**

  Seek transparency and security in land transactions to prevent disputes and ensure rightful ownership.

2.3 Use Cases Analysis

Use-case diagrams model the behavior of a system and help to understand and capture the requirements of the system. They describe the high-level functions and scope of a system and illustrate how the actors interact with the system, rather than detailing the internal operations of the system. The system's components are modeled with a single use-case diagram to provide an overview of how actors use and interact with the system.

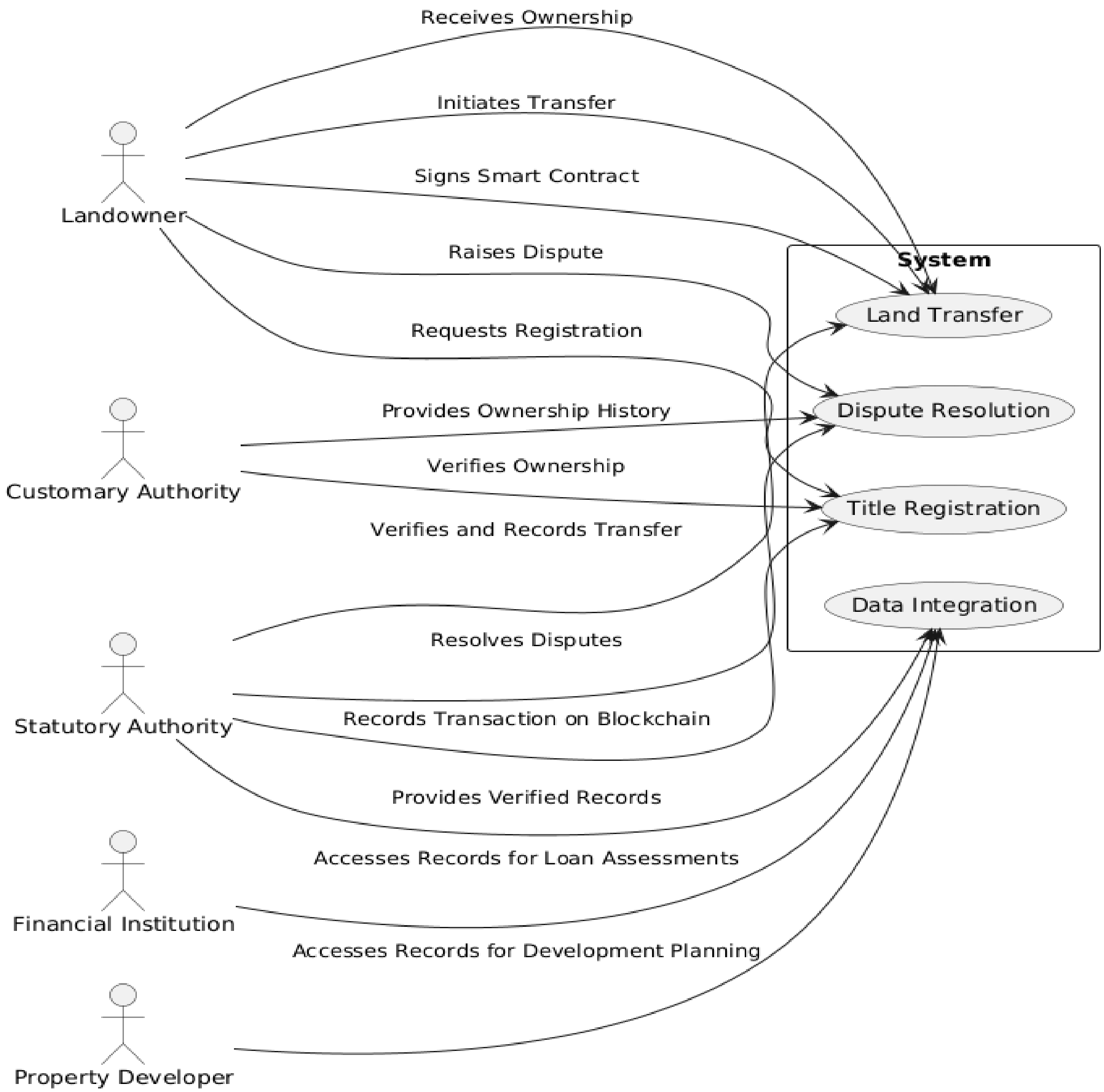

Figure 4. Use Cases

The following use cases outline key interactions with the proposed system:

Title Registration

Actors: Landowners, Customary Authorities, Statutory Authorities.

Process:

- A landowner submits a request to register a land title through the system's frontend interface.
- The land details, including GPS coordinates and ownership documents, are uploaded and hashed (e.g., using IPFS for document storage).
- Customary authorities verify the ownership details and approve the registration.
- The verified land details are registered on the blockchain using the registerLand function in the smart contract.
- The system stores the land details in the PostGIS database for geospatial queries and administrative purposes.
- A transaction hash is returned to the landowner as proof of registration, and the land is displayed on the user dashboard.

Land Transfer

Actors: Current Landowner, Prospective Landowner, Statutory Authorities.

Process:

- The current landowner initiates a transfer request by providing the land ID and the wallet address of the prospective landowner.
- The system verifies that the land is owned by the current landowner and is eligible for transfer (e.g., verified status).

- Both parties agree to the transfer terms, and a smart contract is executed using the transferLand function in the blockchain.
- The blockchain updates the ownership record in real time, and the new owner is recorded in the system.
- The database is updated to reflect the new ownership details, and the transfer is displayed on the admin and user dashboards

Dispute Resolution

Actors: Customary Authorities, Statutory Authorities, Disputing Parties.

Process:

- A dispute is raised regarding the ownership of a specific land parcel.
- The system retrieves the ownership history and transaction records from the blockchain using the getLandDetails function.
- Customary and statutory authorities review the records to establish the ownership history.
- The dispute is resolved transparently based on the immutable records stored on the blockchain.
- If necessary, the resolution is recorded on the blockchain for future reference.

Data Integration

Actors: Statutory Authorities, Financial Institutions, Property Developers.

Process:

- Blockchain APIs are used to provide seamless access to verified land records.

- Financial institutions use the verified records for loan assessments and collateral verification.
- Property developers access land data for project planning and due diligence.
- Statutory authorities integrate blockchain data with their systems for efficient land management and policy enforcement.

2.4 Detailed Requirements

2.4.1. Functional Requirements

Decentralized Ledger: The system uses a blockchain-based smart contract to record land transactions immutably.

Smart Contract Automation: Smart contracts automate processes such as title registration, ownership transfers, and land verification.

API Interoperability: RESTful APIs in the backend enable seamless integration with external systems like financial institutions and property databases. APIs provide endpoints for registering land, fetching land details, and transferring ownership.

Real-Time Verification: The system retrieves real-time ownership and transaction history from the blockchain using functions. Verification processes are supported by smart contracts and backend services.

2.4.2. Non-Functional Requirements

Scalability: The system is designed to handle high transaction volumes by leveraging Ethereum's blockchain network and scalable backend infrastructure. Using PostgGis for geospatial data ensures efficient querying and storage of land records.

Security: Blockchain ensures data integrity and prevents unauthorised modifications. Private keys and secure wallet integrations protect user accounts and transactions.

Usability: The frontend provides an intuitive interface for both technical and non-technical users.

Adaptability: The system supports Ghana's dual governance model by involving both customary and statutory authorities in land verification and dispute resolution.

2.4.3. Socio-Political Requirements

Customary Authority Involvement: The system includes mechanisms for customary authorities to verify land ownership and participate in blockchain governance.

Stakeholder Training: Training programs can be implemented to educate stakeholders (e.g., landowners, authorities) on using the system and understanding blockchain technology.

Legal and Customary Compliance: The system complies with Ghanaian laws and customary practices by integrating statutory and customary governance models. Blockchain records are designed to align with legal requirements for land transactions and ownership.

2.5 Addressed Requirements

This research focuses on the following requirements within the scope of the proposed system:

1. Implementation of a decentralized ledger for land transactions.

2. Development of smart contracts for title registration and ownership transfers.
3. Integration with existing property systems via APIs.
4. Real-time verification of ownership and transaction history.

Requirements such as scalability testing, broader dispute resolution mechanisms, and extensive training programs for stakeholders are beyond the current scope but are recommended for future work.

2.6 Requirements Analysis

The requirements analysis reaffirms that a blockchain-based land administration system offers a robust and transformative pathway to addressing the persistent challenges of transparency, security, and operational inefficiency within Ghana's dual governance framework. By comprehensively targeting functional, non-functional, and socio-political requirements, the proposed framework aspires to bridge the entrenched gap between customary and statutory land administration systems, fostering trust among stakeholders, streamlining transactions, and significantly mitigating the prevalence of land disputes.

Central to the system's architecture is its modular design, which leverages decentralized ledger technology to ensure data immutability, security, and verifiability. This design choice not only guarantees scalability to accommodate future land registration demands but also enables adaptability for integrating with external databases, financial systems, and property management tools. The deployment of smart contracts facilitates the automation of core processes such as title registration, ownership transfer, and dispute resolution, reducing transaction times and minimizing human error. RESTful APIs further enhance the system's interoperability, enabling seamless interaction between blockchain components and legacy statutory and customary systems. These features collectively align with earlier frameworks proposed by Ali et al. [3], Kaushik [5], and

Ansah et al. [7], which emphasized the critical need for decentralized, tamper-proof solutions capable of operating within fragmented governance ecosystems.

The relevance and necessity of these architectural choices are further corroborated by contemporary studies beyond Ghana. Research conducted in rural Nigeria and Benin highlights blockchain's broader applicability, demonstrating its potential to secure land tenure, enhance agricultural financing, and stimulate rural economic development through transparent property rights [15]. These findings echo the socio-economic imperatives underpinning land reform initiatives in Ghana. Moreover, Mansoor et al. [16] stress the importance of developing modular, context-sensitive blockchain systems for land registry management in developing countries, capable of overcoming infrastructural weaknesses and institutional inefficiencies. Their recommendations reinforce the need for blockchain implementations that are not only technically sound but socio-politically adaptable — a principle embedded within the design philosophy of the proposed Assase Aban framework.

In sum, the requirements analysis validates that a decentralized, modular, and socio-culturally integrated blockchain solution, such as Assase Aban, holds significant promise for reforming Ghana's land administration system. It addresses critical gaps identified in prior research while aligning with global best practices and emerging innovations in blockchain-based land governance.

2.7 Ethics

Ethical standards were followed by our institutional IRB board for our stakeholder engagement and requirements collection process to ensure participant privacy and well-being. No personal identifiers were retained, and data were collected only from those who agreed to participate in the study. All collected information was anonymized. Furthermore, at the end of this research project, all data will also be permanently deleted.

Chapter 3: Architecture and Design

3.1 System Overview

The challenges in Ghana’s land governance that Assase Aban addresses are using a blockchain-based land administration system, which combines customary and statutory practices. The main aim of the system is to make land transactions free from corruption, ensure transparency, and promote efficiency. It is a modular architecture that is scalable and adaptable and includes components for decentralised record keeping, smart contract management, API integration and interface for many stakeholders.

3.2 System Architecture

The system architecture provides a blueprint that manages the complexity of the proposed land administration system and establishes clear communication and coordination among its various components. The architecture adopted in this project is a layered architecture, which organises the system into distinct layers, each responsible for a specific set of functionalities. This design choice enhances modularity, scalability, and maintainability, essential for building a secure and efficient blockchain-based land administration system.

In developing the Assase Aban land management platform, the layered architecture decomposes the system into four major layers: the User Interface Layer, Application Layer, Blockchain Layer, and Integration Layer. Each layer operates independently but communicates through well-defined interfaces, ensuring separation of concerns and ease of future system enhancements.

The User Interface Layer provides a graphical interface for stakeholders such as chiefs, Lands Commission officials, and landowners to interact with the system. It abstracts the complexity of backend processes, allowing users to perform actions like land registration, ownership verification, and transaction initiation through intuitive dashboards and forms.

The Application Layer contains the business logic to process user requests, manage workflows, and interface with blockchain components. It acts as the intermediary between the user-facing interface and the blockchain network, ensuring that all processes, such as transaction creation and validation, are handled securely and efficiently.

The Blockchain Layer serves as the system's core, responsible for decentralized data storage, transaction validation, and the execution of smart contracts. By leveraging blockchain technology, this layer ensures that all land records are immutable, transparent, and securely maintained across a distributed network.

The Integration Layer facilitates seamless communication between the blockchain system and external property management databases or financial institutions via RESTful APIs. This layer enhances interoperability, enabling real-time access to land ownership data by authorized external systems while maintaining security and data integrity.

This layered approach supports modularity by isolating different functionalities into manageable units, allowing future extensions, such as integrating advanced GIS features or additional financial services, with minimal disruption to existing operations. It also promotes reusability and maintainability, which are critical for ensuring the long-term sustainability of the Assase Aban system.

3.3 Mapping Design to Requirements

The system architecture of the Assase Aban platform is deliberately structured to align with the functional, non-functional, and socio-political requirements identified during the requirements gathering and analysis phase outlined in Chapter 2. Each major architectural layer is mapped directly to a specific set of requirements, ensuring that the design not only addresses technical needs but also facilitates operational efficiency, stakeholder trust, and system scalability.

The Blockchain Layer is responsible for fulfilling the requirement of Decentralized Record-Keeping. By leveraging a permissioned blockchain network, the system ensures that all land transaction records are immutable, tamper-proof, and transparently verifiable. This decentralization minimizes fraud risks and unauthorized alterations, addressing critical transparency and security challenges inherent in Ghana’s current land governance frameworks.

Additionally, the Blockchain Layer manages Smart Contract Automation, providing a secure and efficient mechanism for executing processes such as ownership transfers, title registration, and property subdivisions. Smart contracts enforce pre-defined business logic without intermediaries, reducing transaction times, minimizing errors, and ensuring that all actions are recorded on the blockchain in real time.

The Integration Layer satisfies the requirement of Integration with Existing Systems. This layer acts as a bridge between the blockchain network and external property management databases, financial institutions, and regulatory systems. Through RESTful APIs, the integration layer enables real-time data access and interoperability, facilitating seamless verification of land records by authorized third parties while maintaining system security and data integrity.

The User Interface Layer addresses the requirement for User-Friendly Interfaces, providing stakeholders such as chiefs, Lands Commission officials, property developers, and landowners with intuitive dashboards and forms. This layer abstracts the complexity of backend processes, offering a streamlined and accessible interaction experience for users of varying technical expertise.

By structurally aligning each architectural layer with corresponding system requirements, the Assase Aban platform ensures that the final implementation remains faithful to the original design objectives. This alignment not only enhances system reliability and usability but also lays the foundation for future scalability and adaptability, supporting the long-term vision of transforming Ghana's land administration landscape.

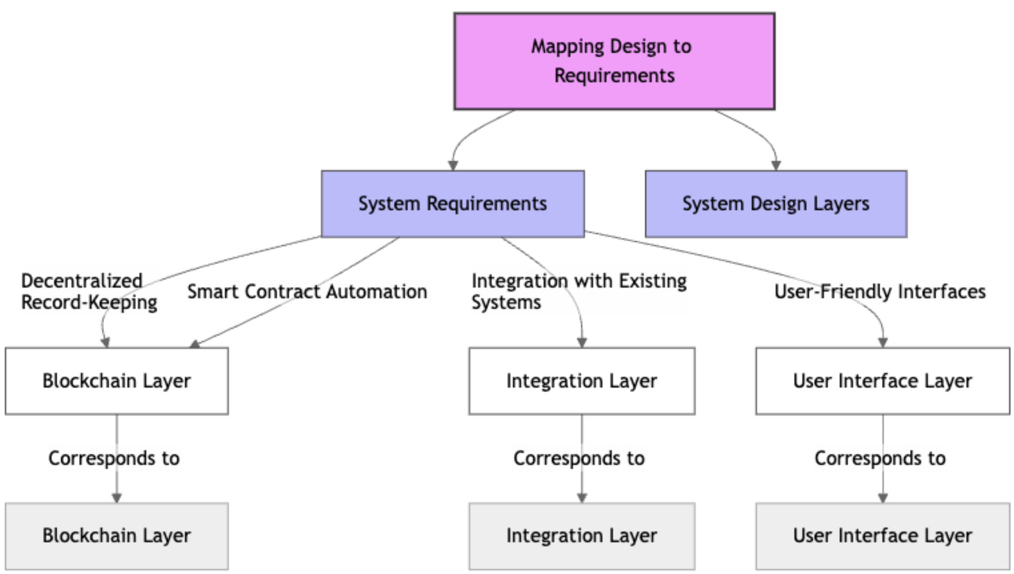

Figure 5. Mapping Design to Requirements

3.4 Key Modules and Design

1. **Decentralized Ledger Module:**
    - **Functionality:** Records and validates transactions immutably on a permissioned blockchain.
    - **Design:** Uses Hyperledger Fabric with consensus algorithms to validate transactions.
2. **Smart Contract Module:**
    - **Functionality:** Automates processes such as title registration, land transfers, and dispute resolution.
    - **Design:** Built using Solidity, with contracts deployed on the blockchain layer.
3. **API Gateway Module:**
    - **Functionality:** Enables integration with existing systems like property management tools and financial databases.
    - **Design:** RESTful API with authentication mechanisms for secure data exchange.
4. **User Interface Module:**
    - **Functionality:** Provides dashboards for stakeholders to view and manage land records.
    - **Design:** Developed using React.js for flexibility and responsiveness.

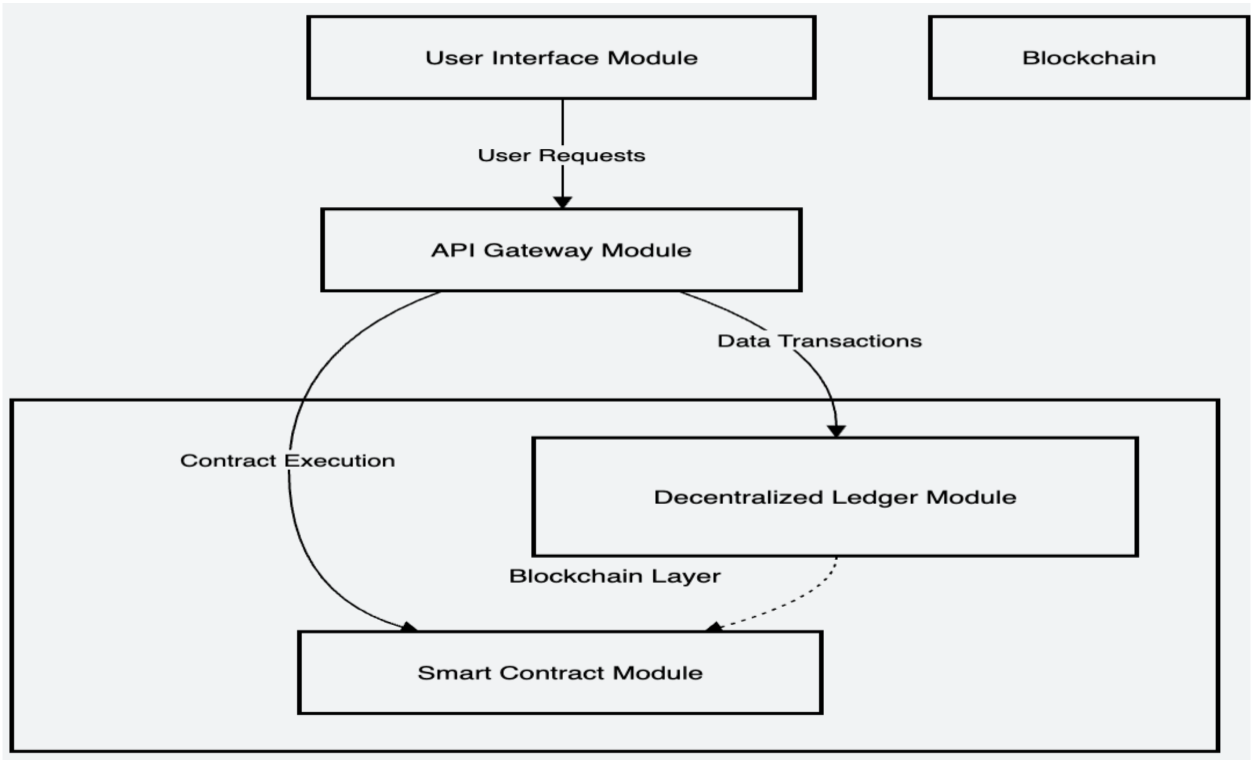

Figure 6. Key Modules and Design

3.5 System Schematics

The system is designed to achieve smooth and transparent land registration, transfer, and verification processes in a secure, transparent way via blockchain technology. As a result, it provides a frontend interface for the user to interact, a backend API for the requests to be processed, and a smart contract deployed to the blockchain addressing the record keeping for the purpose in an immutable manner. Additionally, IPFS is used for decentralised document storage, and Postgis is used for geospatial data storage with a Postgis database at the back of this system. The following is the system schematic diagram showing the component and actor interactions.

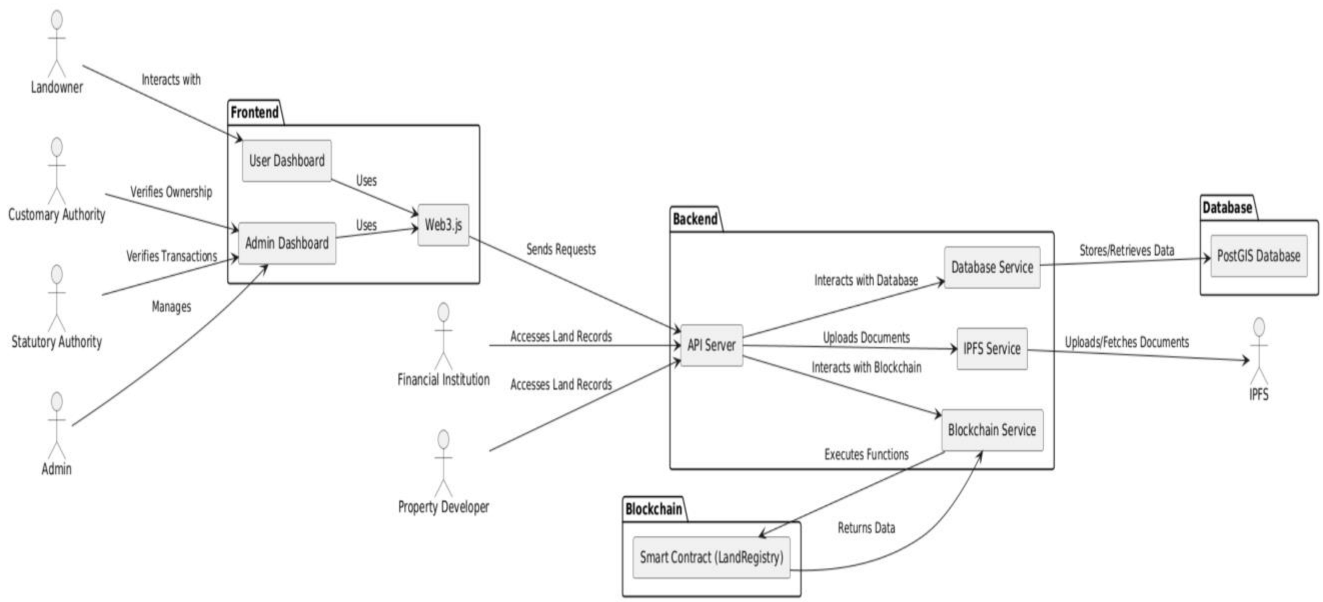

Figure 7. System Design

3.6 Key Modules and Design

The Assase Aban system is designed based on four key modules, which are involved in four functionalities essential for the blockchain-based framework for land administration. All these modules work together to provide transparency, security, interoperability, and user accessibility while avoiding confusion and clearly separating concerns about modularity, maintainability, and further scalability.

3.6.1 Decentralized Ledger Module

That means the Decentralized Ledger Module manages tamper-proof records of all the land transactions. The transaction data is stored in this module and is guaranteed for immutability and transparency with the Ethereum blockchain, particularly located in Sepolia Testnet. The immutability of the blockchain and the transactions it hosts means that land transactions, registration, transfer of ownership, and verification processes are recorded. These transactions are validated and executed by smart contracts so that data integrity and unauthorized changes will be prevented. The trust model of the system relies on the absence of single points of failure and human-induced manipulation, which is made possible by the decentralized ledger.

3.6.2 Smart Contract Module

The Smart Contract Module automates essential operations within the system, such as land transfers, title registrations, and ownership verifications. Developed using Solidity, the smart contracts encapsulate the business logic required for secure and transparent land administration. The module's core functions include registerLand for registering new land parcels along with location and document hashes, transferLand for transferring ownership between stakeholders, and verifyLand for confirming ownership status and updating records. The LandRegistry contract serves as the central smart contract, ensuring that all operations adhere to pre-defined rules and are automatically enforced without manual intervention, thereby enhancing reliability and efficiency.

3.6.3 API Gateway Module

The API Gateway Module acts as an intermediary between the user-facing interfaces, the blockchain network, and the backend database. Built using Node.js and Express.js, this module handles incoming API requests from both frontend components and external systems. Key routes include /api/land/register for registering land by interacting simultaneously with the blockchain and the PostgreSQL/PostGIS database, /api/land/lands for retrieving all registered land records, and /api/land/:landId for fetching specific land details directly from the blockchain. The module integrates with blockchain services via the ethers.js library and ensures secure and reliable database interactions, thereby enabling seamless system interoperability.

3.6.4 User Interface Module

The User Interface Module provides user-friendly dashboards and interactive tools for all system stakeholders, including landowners, administrators, and statutory authorities. Developed using HTML, CSS, Vanilla JavaScript, Leaflet.js, and Bootstrap, this module delivers an accessible and responsive user experience. Key functionalities include an Admin Dashboard that allows administrators to view all registered lands, verify land status, and execute ownership transfers, and a User Dashboard where individual landowners can view their properties and register new land parcels. The User Interface Module communicates with the backend API to submit user inputs and retrieve real-time processed data for visualization, facilitating transparency and ease of use in land administration operations.

The modular design of these components ensures that the Assase Aban system remains adaptable to future enhancements, such as integrating biometric verification for identity management or extending functionality to support mortgage and leasing operations, without disrupting existing system stability.

3.7 Summary

Assase Aban provides an architecture and design of Assase Aban, a robust and scalable land administration system, which is user-centric. Every module aligns with the system's terms and conditions, including its integrity, transparency, and land transaction responsibility. Future enhancements, like integration of other services or rollout of other land governance models, are made possible by the modular design.

Chapter 4: Implementation

4.1 Introduction

The Asaase Aban blockchain-based land administration system is a system of implementation that will provide a satellite view of Ghana's ever-changing land governance by affording web application functionality using geospatial databases, blockchain technology and modern web development tools. This section highlights the tools, technologies and processes used to develop this system, specifically illustrating how the tools and technologies allow the system to be transparent, efficient and function seamlessly. This implementation follows the architecture design principles and can provide a robust, scalable solution for registering, transferring and verifying land. Each module implements the system to meet specific challenges in land governance within the compliance of Ghanaian laws and customary practices.

4.2 System Implementation Overview

The Asaase Aban system leverages advancements across database, backend, frontend, and blockchain technologies to move beyond traditional land administration models, delivering a transparent and efficient decentralized platform.

Database tools such as PostgreSQL and its PostGIS extension manage land and user data, while pgAdmin facilitates database visualization and administration. **Backend development** is driven by Node.js and Express.js, with additional support from libraries like Multer for file uploads, pg for database connections, and Web3.Storage for decentralized document storage via IPFS.

On the **frontend**, the system employs HTML, CSS, JavaScript, and Bootstrap to build a responsive user interface, while Leaflet.js enables dynamic map visualizations and Font Awesome enhances UI aesthetics. Ethers.js facilitates blockchain interactions within the frontend. **Blockchain components** include the Ethereum Sepolia Testnet, Solidity smart contracts for land management, and Hardhat for contract development, deployment, and testing. Together, these technologies enable Asaase Aban to provide a scalable, secure, and user-centric land administration solution.

4.3 Blockchain Network Setup

The blockchain network serves as the foundational infrastructure of the Asaase Aban system, enabling decentralized, immutable, and verifiable land transaction records. The setup employs the Ethereum blockchain, leveraging the Sepolia Testnet for development and testing purposes, smart contracts developed in Solidity, and development tools such as Hardhat and Ethers.js for deployment, interaction, and system integration. The architecture ensures reliability, transparency, and scalability in land registry operations.

Network Configuration. The network configuration is built upon the Ethereum blockchain using its Sepolia Testnet, selected for its public accessibility and compatibility with production-grade Ethereum tooling. Smart contracts are deployed on the Sepolia network, with deployment operations managed through the Hardhat development environment. The primary smart contract, *LandRegistry.sol*, is deployed via scripts in /blockchain/blockchain/scripts/deploy.js. The backend and frontend components interact with the deployed smart contract using the *ethers.js* library, facilitating seamless blockchain transactions. The backend service, responsible for blockchain operations, resides in /backend/src/services/blockchainService.js, while the frontend connection is established through /frontend/js/web3.js. To enable secure and reliable access to the Sepolia network, the system utilises Alchemy as its Ethereum node provider, configured through the environment variable SEPOLIA_RPC_URL within the project's .env file.

Smart Contract Deployment: The deployment of smart contracts is a multi-step process designed to ensure the correctness and readiness of the blockchain infrastructure. Initially, the smart contract code is compiled using Hardhat with the command:

npx hardhat compile

Following compilation, the *LandRegistry.sol* contract is deployed to the Sepolia Testnet using:

npx hardhat run blockchain/scripts/deploy.js network sepolia.

Upon successful deployment, the contract address is automatically stored in the environment file under the CONTRACT_ADDRESS variable for consistent access across the backend and frontend services.

Ledger Initialisation: Unlike permissioned blockchain networks that require manual genesis block creation, the Ethereum network inherently manages the genesis block. Ledger initialization within Assase Aban is achieved through deploying and activating the smart contract. Landowners initialize land records by invoking the registerLand (string location, string documentHash) function within *LandRegistry.sol*, thereby immutably recording parcel details onto the blockchain ledger.

Interaction with the Blockchain; Interaction with the blockchain is facilitated both through the frontend and backend services using the *ethers.js* library.

Frontend Interaction: The frontend application enables users to interact with the smart contract via browser-based Ethereum providers such as MetaMask. The interaction workflow involves creating a blockchain connection and instantiating the bright contract object using the following example:

```
const provider = new ethers.BrowserProvider(window.ethereum);
const contract = new ethers.Contract(contractAddress, contractABI, signer);
```

Figure 8. A code snippet showing the Frontend interaction with the blockchain.

This setup enables users to register new land, transfer ownership, and verify land ownership directly from the user interface.

Backend Interaction: The backend server executes blockchain operations on behalf of the system using authenticated wallet credentials. For instance, when a land registration is initiated from the frontend, the backend interacts with the smart contract using:

```
const contract = new ethers.Contract(contractAddress, contractABI, wallet);
const tx = await contract.registerLand(location, documentHash);
```

Figure 9. A code snippet showing the backend interaction with the smart contract

This dual interaction model ensures both real-time responsiveness for users and secure, automated transaction handling for administrative and system processes.

The comprehensive blockchain network setup, combining Sepolia Testnet deployment, Hardhat-based contract management, and ethers.js integration, ensures that the Assase Aban system provides a secure, transparent, and efficient foundation for modernized land administration.

4.4 Smart Contract Development

Smart contracts form the core automation engine of the Asaase Aban system, enabling decentralized, transparent, and efficient management of land administration processes.

Developed in Solidity and deployed on the Ethereum Sepolia Testnet, the primary contract,

LandRegistry.sol, encapsulates essential functionalities such as property registration, ownership transfer, and land verification.

The registerLand function immutably records land location and associated document hashes (e.g., IPFS links) on the blockchain, ensuring verifiability and tamper-proof records. Ownership transfers are securely handled through the transferLand function, which enforces conditions such as prior verification and owner authentication to maintain transactional integrity. Verification of land parcels is restricted to authorized verifiers via the verifyLand function, reinforcing trust and preventing unauthorized alterations. All operations automatically update the blockchain ledger, ensuring real-time, decentralized record management.

Through its smart contract framework, Asaase Aban advances a scalable, secure, and transparent model for modern land governance, minimizing human error and reducing reliance on manual intermediaries.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

contract LandRegistry {
    struct Land {
        address owner;
        string location;
        string documentHash;
        bool verified;
    }

    mapping(uint256 => Land) public lands;
    uint256 public landCounter;

    event LandRegistered(uint256 landId, address indexed owner, string location);

    // ✅ Register a new land parcel
    function registerLand(string memory location, string memory documentHash) public {
        landCounter++;
        lands[landCounter] = Land({
            owner: msg.sender,
            location: location,
            documentHash: documentHash,
            verified: false
        });

        emit LandRegistered(landCounter, msg.sender, location);
    }
}
```

Figure 10. A code snippet showing the smart contract code for land title registration

4.5 API Gateway Development

The API gateway serves as a bridge between the blockchain system, the database, and external applications. It provides RESTful endpoints for interacting with the blockchain, managing land records, and facilitating user operations. The development process leverages **Express.js** for building the backend server and integrates with the Ethereum blockchain and PostgreSQL database.

1. Setup

Technology:

Express.js: A lightweight web framework for building RESTful APIs.

Ethers.js: For interacting with the Ethereum blockchain.

pg (PostgreSQL Client): For database operations.

```
npm install express body-parser ethers pg dotenv cors
```

Figure 11. Code Snippet showing how to install all dependencies for the backend.

Configure environment variables in the .env file:

```
SEPOLIA_RPC_URL=<Your_Sepolia_RPC_URL>
PRIVATE_KEY=<Your_Private_Key>
CONTRACT_ADDRESS=<Your_Contract_Address>
DB_USER=<Your_Database_User>
DB_PASSWORD=<Your_Database_Password>
DB_HOST=<Your_Database_Host>
DB_PORT=<Your_Database_Port>
DB_NAME=<Your_Database_Name>
```

Figure 12. A code snippet for the .env file for the whole setup

2. API Endpoints

The API gateway provides the following key endpoints:

Register Land:

Route: POST /api/land/register

1. **Description**: Registers a new land parcel by interacting with the blockchain and storing details in the database.

2. **File**: /backend/src/routes/landRouters.js

```
router.post("/register", upload.single("document"), async (req, res) => {
    try {
        const { owner, location } = req.body;
        const documentUrl = await uploadDocument(req.file.path); // Upload file to IPFS

        if (!documentUrl) {
            return res.status(500).json({ error: "Failed to upload document" });
        }

        const land = await registerLand({ owner, location, document: documentUrl });
        res.json({ message: "Land registered successfully!", land });
    } catch (error) {
        console.error("❌ Error registering land:", error);
        res.status(500).json({ error: "Internal Server Error" });
    }
});
```

Figure 13. A code Snippet showing the API endpoint.

3. Get All Lands:

Route: GET /api/land/lands

Description: Retrieves all registered lands from the database.

File: /backend/src/routes/landRouters.js

```
router.get("/lands", getLands);
```

Figure 14. A code snippet of a function for getting all lands.

4.Get Land Details:

Route: GET /api/land/:landId

Description: Fetches details of a specific land parcel from the blockchain and database.

File: /backend/src/routes/landRouters.js

```
router.get("/land/:landId", getLandDetails);
```

Figure 15. A code snippet of a function for getting all lands details.

5.Transfer Land Ownership:

Route: POST /api/land/transfer-land

Description: Transfers ownership of a land parcel to a new owner.

File: /backend/src/routes/userLands.js

```
router.post("/transfer-land", async (req, res) => {
    const { landId, newOwner, currentOwner } = req.body;
    try {
        const transaction = await transferLandOnBlockchain(landId, newOwner);
        res.json({ success: true, message: "Land transferred successfully", transaction });
    } catch (err) {
        console.error("❌ Error transferring land:", err.message);
        res.status(500).json({ error: "Internal Server Error" });
    }
});
```

Figure 16. A code snippet showing how a land is transferred.

6. Integration with Blockchain

File: /backend/src/services/blockchainService.js

Description: Handles blockchain interactions using ethers.js.

```
const provider = new ethers.providers.JsonRpcProvider(process.env.SEPOLIA_RPC_URL);
const wallet = new ethers.Wallet(process.env.PRIVATE_KEY, provider);
const contract = new ethers.Contract(process.env.CONTRACT_ADDRESS, contractABI, wallet);

async function registerLandOnBlockchain(location, documentHash) {
    const tx = await contract.registerLand(location, documentHash);
    await tx.wait();
    return tx.hash;
}
```

Figure 17. A code snippet showing blockchain interaction using ethers.js

7. Integration with Database

File: /backend/src/database.js

Description: Manages database operations using the PostgreSQL client.

```
const pool = new Pool({
    user: process.env.DB_USER,
    password: process.env.DB_PASSWORD,
    host: process.env.DB_HOST,
    port: process.env.DB_PORT,
    database: process.env.DB_NAME,
});

module.exports = {
    query: (text, params) => pool.query(text, params),
};
```

Figure 18. A code snippet of a database operation using the PostgreSQL client

4.6 User Interface Development

The user interface of the Asaase Aban system serves as the primary gateway for landowners, administrators, and stakeholders, providing seamless interaction with the decentralized land registry. Designed with accessibility, usability, and responsiveness in mind, the web-based dashboard connects directly to backend services and the blockchain network, offering real-time land administration functionalities in an intuitive environment. Developed using Vanilla JavaScript (ES6+) within a modular architecture, the interface ensures maintainability, scalability, and future-proofing.

Key libraries integrated include Leaflet.js for dynamic map visualizations, Ethers.js for blockchain interactions, and HTML5, CSS3, and Font Awesome for responsive and visually engaging design. Core features of the interface include land registration via geolocation and IPFS document uploads, ownership transfer between stakeholders, administrator-based land verification, dynamic map visualization of registered parcels, and GPS-based search and existence checks. Through its integration of blockchain and geospatial tools, the Asaase Aban user interface delivers a secure, efficient, and user-centric experience, advancing the modernization of land administration practices.

This is the user interface

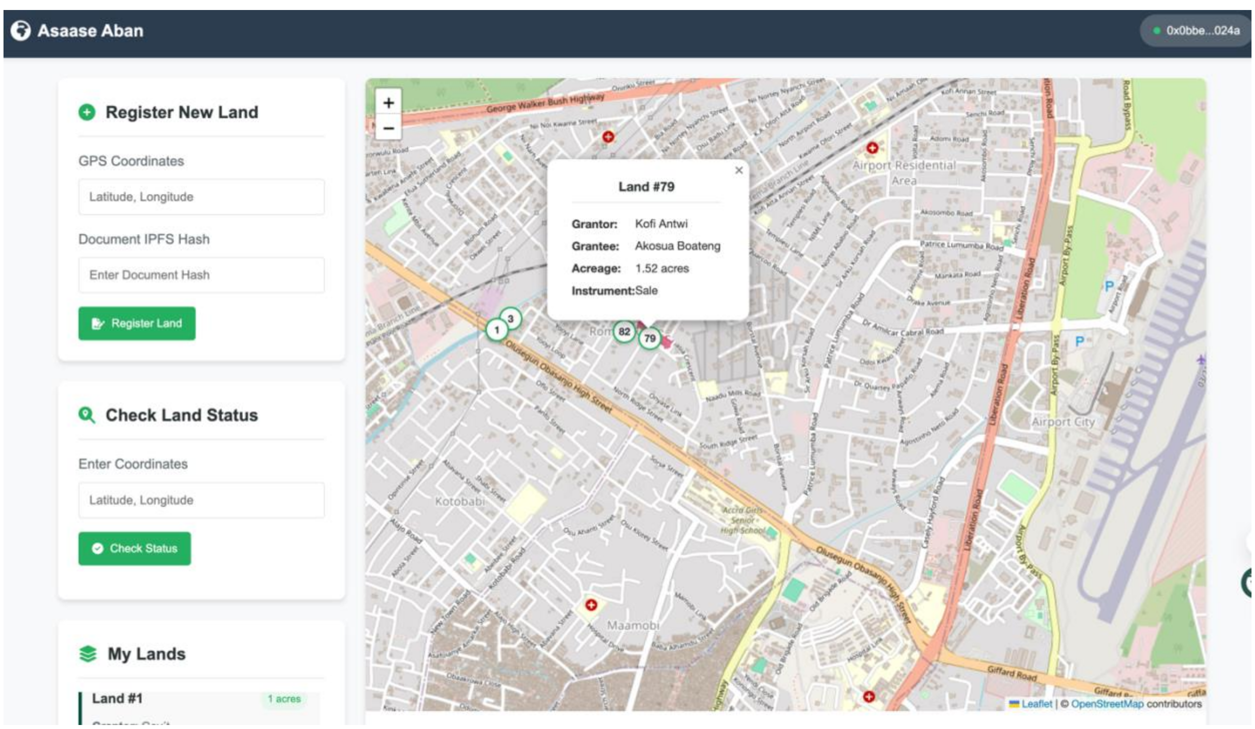

Figure 19. A picture of the User interface

This is the admin interface

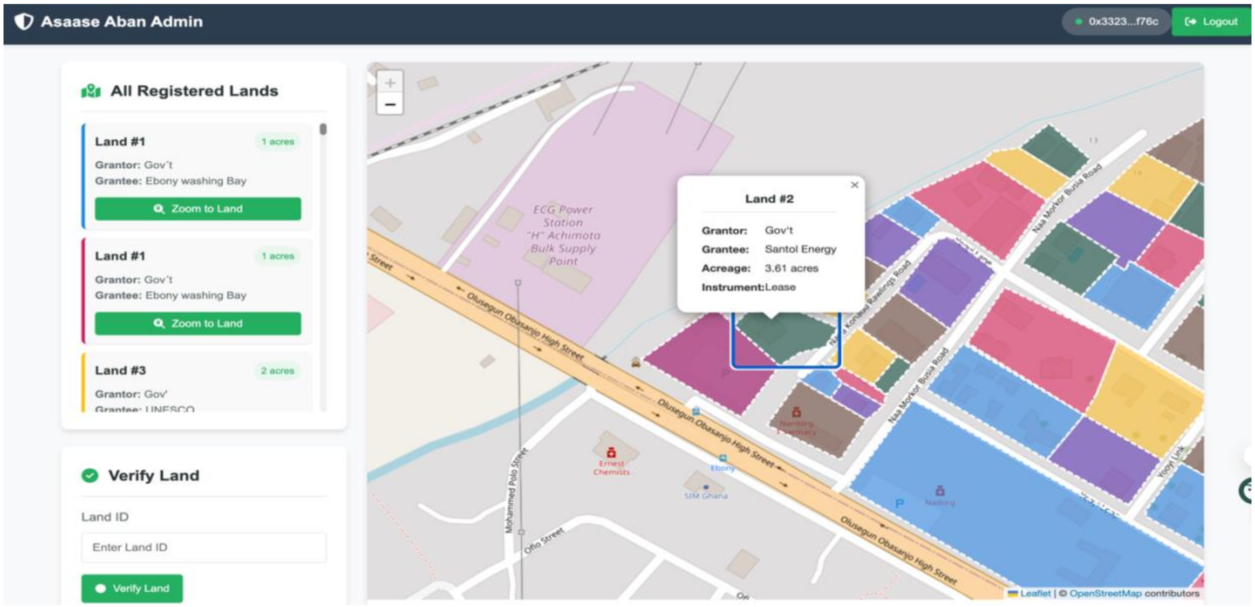

Figure 20. A picture of the Admin interface

4.7 PostgreSQL Database Integration

The PostgreSQL database forms a critical component of the Assase Aban system's backend infrastructure, providing structured storage for metadata related to land records, user accounts, and ownership relationships. Complementing the blockchain's immutable ledger, the database enables efficient querying, advanced geospatial analysis, and comprehensive administrative oversight. Its integration strengthens the overall system's performance, scalability, and accessibility.

Database System The database utilizes PostgreSQL, a robust open-source relational database system well-suited for managing structured transactional data. To extend its capabilities to spatial data management, the PostGIS extension is incorporated, enabling the system to efficiently store, index, and query land parcel boundaries and geolocation attributes. Together, PostgreSQL and

PostGIS provide a powerful platform for handling both attribute and spatial components of land administration.

Tables and Structure

The Asaase Aban system's backend database is structured around key relational tables to manage land metadata, geospatial data, and user associations efficiently. The **LandParcel** table stores core land information, including owner wallet addresses, geolocation coordinates, IPFS hashes for document referencing, verification status, and registration timestamps (/backend/db.sql). The **Romman** table manages spatial boundaries and transactional metadata, such as grantor and grantee details, parcel acreage, transaction cost, and the legal instrument governing the transaction.

The **User_Land** table establishes relationships between users and their associated land parcels, linking blockchain wallet addresses to land IDs for efficient query handling. The **Users** table maintains user information, specifically blockchain wallet identifiers. Referential integrity across the database is enforced through foreign key constraints, ensuring consistent linkage between user accounts and land records. This relational structure, combined with geospatial indexing, supports scalable, secure, and efficient land data management within the Asaase Aban platform.

Key Features

The Asaase Aban system employs a secure, normalized PostgreSQL database structure to minimize redundancy, ensure data integrity, and support scalable growth. Enhanced with PostGIS, the database enables advanced geospatial operations such as boundary verification and spatial analysis, facilitating dynamic mapping for users and administrators. Seamless backend integration via Node.js service modules ensures efficient CRUD operations and secure transaction processing. The system also supports filtered retrieval of land records based on ownership, location, and verification status, improving operational efficiency. Additionally, integration with GeoServer allows dynamic publishing of land layers for comprehensive spatial visualization. Through these combined technologies, Asaase Aban establishes a robust, scalable, and efficient data management framework, strengthening its decentralized land administration platform.

Example of Postgres Query

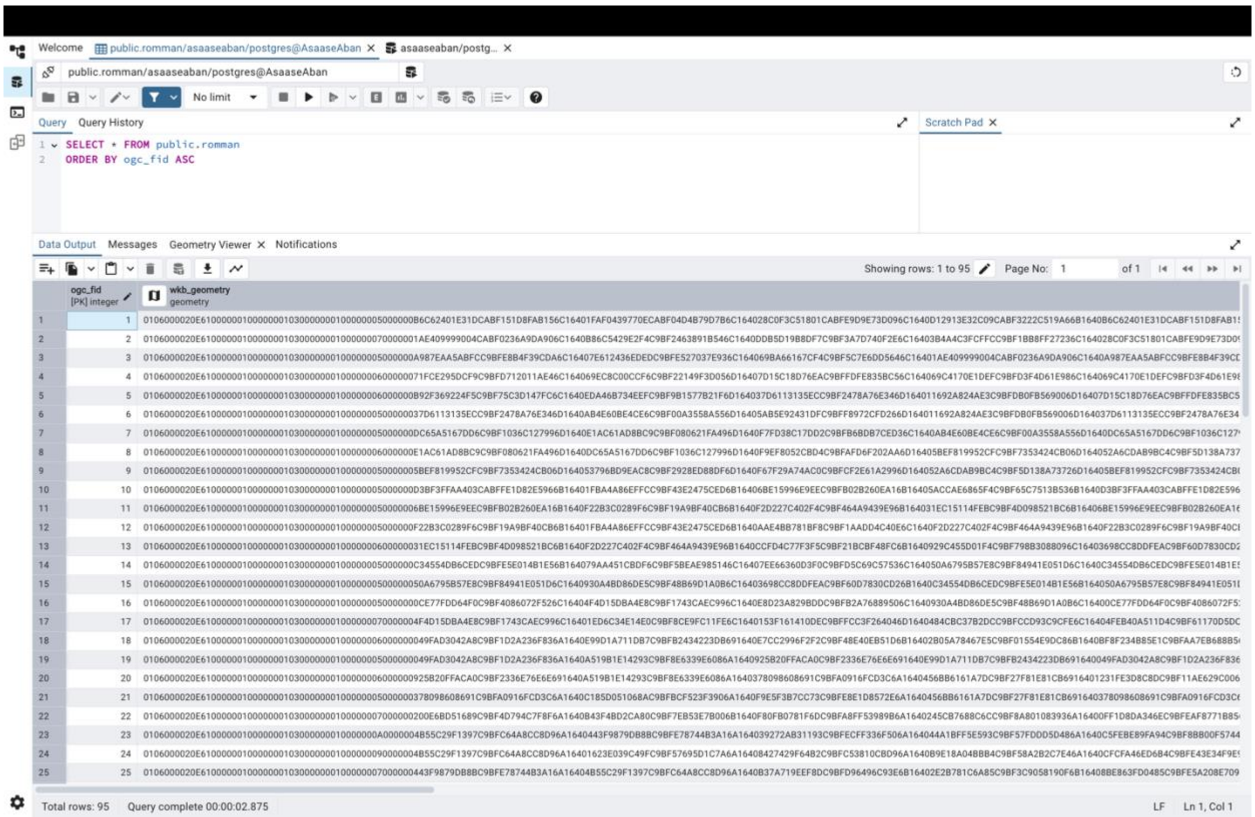

Figure 21. A pciture of Postgres query and the results.

4.8 Decentralized File Storage Integration

To enhance transparency, trust, and resilience, the Asaase Aban system integrates decentralized file storage alongside its blockchain-based transaction records. Critical land documents, such as deeds, sales agreements, and transfer proofs, are securely stored using the

InterPlanetary File System (IPFS), a peer-to-peer decentralized storage protocol. By distributing documents across a global network rather than relying on centralized servers, IPFS ensures tamper-proof verification, improves data availability, and protects against document loss or fraud, reinforcing the integrity of the land registration process.The Asaase Aban platform implements a decentralized document management workflow to ensure secure and tamper-proof handling of land-related documents. Landowners or administrators initiate document uploads through the frontend interface (/frontend/pages/user.html), submitting critical files such as deeds or ownership agreements during registration processes. Upon submission, the backend server utilizes the Web3.Storage client to store documents on the IPFS network, generating a unique cryptographic Content Identifier (CID) for each file (/backend/src/utils/ipfs.js).

The IPFS hash is systematically integrated into both the blockchain and the PostgreSQL database. On the blockchain, the CID is embedded within the registerLand transaction via smart contract execution (/blockchain/contracts/LandRegistry.sol), ensuring immutability. Simultaneously, the CID is stored in the backend database (/backend/db.sql) alongside other land metadata for administrative retrieval. When needed, the IPFS hash allows for secure, decentralized retrieval of original documents, ensuring verifiability, integrity, and resilience against tampering in land ownership verification processes.

Key Features

The Asaase Aban system extends decentralization and immutability to land-related documents by integrating decentralized file storage via IPFS. Each land registration records a permanent IPFS hash, ensuring that documents remain tamper-proof and verifiable. By distributing documents across multiple IPFS nodes, the system enhances resilience against data loss and centralized attacks. IPFS-linked documents serve as immutable proof of ownership during verification, administrative reviews, and dispute resolutions. Furthermore, by anchoring IPFS hashes on the blockchain during registration, the platform reinforces the transparency, auditability, and integrity of both transaction data and associated legal documents, creating a comprehensive and secure land administration framework.

```
async function uploadDocument(filePath) {
    const fileBuffer = fs.readFileSync(filePath);
    const fileName = filePath.split("/").pop();
    const file = new File([fileBuffer], fileName);
    const cid = await client.put([file]);
    return `https://${cid}.ipfs.w3s.link/${fileName}`;
}
```

Figure 22. A code snippet that handles file uploads to IPFS using Web3.Storage

```
CREATE TABLE IF NOT EXISTS public."LandParcel"
(
    id serial NOT NULL,
    owner text NOT NULL,
    location text NOT NULL,
    "ipfsHash" text NOT NULL,
    verified boolean NOT NULL DEFAULT false,
    "createdAt" timestamp(3) NOT NULL DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT "LandParcel_pkey" PRIMARY KEY (id)
);
```

Figure 23. A code snippet showing how IPFS hash for each land document is stored.

4.9 System Diagram

The system architecture of the Assase Aban platform is visually represented through an updated interaction flow diagram, illustrating the relationships and communication pathways between the User Interface, API Gateway, Database, and Blockchain layers. This diagram highlights how these key components collaborate to deliver a seamless, decentralized, and transparent land administration experience for stakeholders.

4.9.1 System Interaction Flow

The Asaase Aban system architecture integrates four core components to ensure secure, transparent, and efficient land administration workflows. The **User Interface** acts as the primary access point for landowners, administrators, and stakeholders, enabling land registration, ownership transfers, and verification activities. It sends user inputs to the backend and displays retrieved blockchain and database records, such as land parcels and ownership history.

The **API Gateway**, developed with Node.js and Express.js, serves as the intermediary between the frontend, blockchain network, and PostgreSQL database. It processes user requests, executes smart contract functions, and manages data storage and retrieval operations. The **PostgreSQL/PostGIS database** underpins backend storage, handling structured metadata for land parcels, user accounts, and geospatial queries for visualization and search functionalities.

Finally, the **Blockchain layer**, built on the Ethereum Sepolia Testnet with Solidity smart contracts, ensures the immutability of land transaction records. It manages critical operations such as land registration, ownership transfer, and verification while storing IPFS document hashes to guarantee tamper-proof access to land ownership evidence. Together, these components form a cohesive, decentralized system architecture that underpins the Asaase Aban platform's functionality.

Key Components Depicted in the Diagram

The Asaase Aban system employs a modular, layered architecture to deliver decentralized land administration services with high scalability, security, and usability. The **User Dashboard** allows landowners to register new land parcels and view their owned lands, while the **Admin Dashboard** enables administrators to verify land records, manage ownership transfers, and oversee transaction histories. **Map visualization**, powered by Leaflet.js, dynamically displays registered land parcels and associated metadata. **Backend services** include an API server that routes requests to blockchain or database services, a blockchain service using Ethers.js to interact with smart contracts, a database service managing structured land data in PostgreSQL/PostGIS, and an IPFS service for decentralized document storage and retrieval. **Blockchain components** center on the *LandRegistry* smart contract, which governs registration, ownership transfer, and land verification processes. The **database structure** is organized across tables such as *LandParcel* (storing land metadata), *Romman* (managing geospatial data and transactions), *User_Land* (linking users to land parcels), and *Users* (storing authentication data). Together, these components ensure the Asaase Aban system functions as a robust, transparent, and efficient platform for modern land governance..

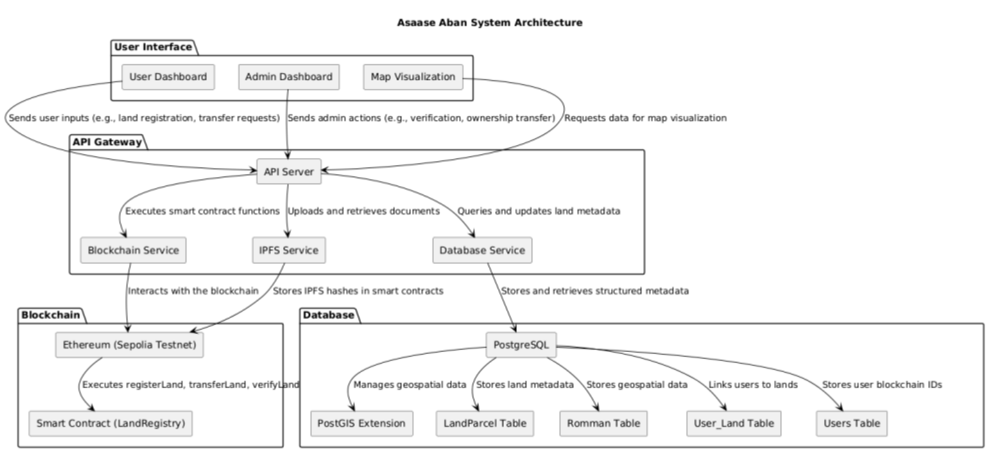

Figure 24. System Interaction Diagram

4.10 Integration with External Systems

To enhance interoperability and extend its functional reach, the Asaase Aban system is designed for seamless integration with external property management platforms, financial institutions, and government oversight systems. Integration is achieved through a combination of RESTful APIs, blockchain-based verification mechanisms, and decentralized document retrieval via IPFS, ensuring secure, transparent, and efficient interactions.

The system's Node.js and Express.js-based API Gateway exposes well-defined RESTful endpoints, allowing external systems to query registered lands, specific land details, and user-associated land assets. Property management platforms and financial institutions can leverage these APIs for ownership verification, transaction history review, and due diligence in mortgage assessments. Beyond API access, external systems can interact directly with Ethereum smart contracts, retrieving immutable land records through functions like getLandDetails and verifyLand, thereby enhancing data integrity and trust. Additionally, IPFS integration allows external entities to securely retrieve land documents via cryptographic CIDs, eliminating reliance on centralized storage and preserving document authenticity.

Use Cases

The Asaase Aban system enables seamless integration with property management platforms, financial institutions, and government agencies, allowing for secure querying of land records, validation of collateralized assets, and centralized oversight. Secured by authentication mechanisms and designed for scalability, the platform supports high-volume interactions while maintaining blockchain-backed transparency and integrity, positioning it as a reliable solution for modern land administration across sectors.

Chapter 5: Testing and Validation

5.1 Testing and Validation Overview

The testing and validation phase was crucial in ensuring the Asaase Aban system met the required standards of reliability, security, and usability for a blockchain-based land registry. Comprehensive testing was conducted across smart contracts, backend services, APIs, database integrations, and the user interface to verify that all functional and non-functional requirements were satisfied. A combination of unit testing, integration testing, and user acceptance testing (UAT) was employed, allowing for iterative improvements and optimization. Special focus was placed on validating security, data immutability, user interactions, and interoperability. This chapter outlines the testing methodologies, key findings, and refinements made during the development process.

5.1.1 Smart Contract Testing

The blockchain smart contract functionality, central to the Asaase Aban platform, underwent extensive unit testing using the Hardhat development environment, with Chai assertion libraries employed for detailed validation of contract behavior. Tests were implemented within the LandRegistry.test.js file to rigorously evaluate all critical aspects of the LandRegistry.sol smart contract. The testing environment utilized Hardhat for contract deployment and testing, Ethers.js for blockchain interactions, and Chai for crafting expressive and flexible test cases.

The unit tests comprehensively covered key smart contract functionalities to ensure data integrity, security, and enforcement of business logic. Land registration workflows were validated to confirm that submitted land metadata, including location and IPFS document hashes, was correctly recorded and associated with user wallet addresses. Land verification processes were tested to ensure that only authorized verifiers could approve land records, with unauthorized attempts properly rejected. Ownership transfer functionality was carefully assessed to guarantee that only verified land parcels could be transferred by rightful owners. Document verification was tested to confirm the integrity of off-chain references through IPFS hashes, while authority management features were evaluated to ensure administrative privileges were securely assigned. Negative testing also confirmed that unauthorized users could not manipulate the contract, and that all invalid operations reverted with appropriate error messages.

The outcomes of smart contract testing validated that the LandRegistry.sol contract operated securely, reliably, and in strict adherence to defined business rules. Potential vulnerabilities such as unauthorized access or data inconsistencies were effectively mitigated, strengthening the overall trustworthiness and resilience of the Asaase Aban blockchain system.

```javascript
const { expect } = require("chai");
const { ethers } = require("hardhat");

describe("LandRegistry", function () {
    let LandRegistry, landRegistry;
    let owner, user1, user2, authority;

    beforeEach(async function () {
        [owner, user1, user2, authority] = await ethers.getSigners();
        LandRegistry = await ethers.getContractFactory("LandRegistry");
        landRegistry = await LandRegistry.deploy();
    });

    it("Should register a new land parcel", async function () {
        await expect(landRegistry.connect(user1).registerLand("5.6037, -0.1870", "Qm1234IPFSHash"))
            .to.emit(landRegistry, "LandRegistered")
            .withArgs(1, user1.address, "5.6037, -0.1870");

        const land = await landRegistry.getLandDetails(1);
        expect(land.owner).to.equal(user1.address);
        expect(land.location).to.equal("5.6037, -0.1870");
        expect(land.documentHash).to.equal("Qm1234IPFSHash");
    });

    it("Should allow only verified authority to verify land", async function () {
        await landRegistry.connect(owner).addVerifiedAuthority(authority.address);
        await landRegistry.connect(user1).registerLand("5.6037, -0.1870", "Qm1234IPFSHash");

        await expect(landRegistry.connect(authority).verifyLand(1))
            .to.emit(landRegistry, "LandVerified")
            .withArgs(1, true);

        const land = await landRegistry.getLandDetails(1);
        expect(land.verified).to.equal(true);
    });
```

```
it("Should not allow unverified users to verify land", async function () {
    await landRegistry.connect(user1).registerLand("5.6037, -0.1870", "Qm1234IPFSHash");

    await expect(landRegistry.connect(user2).verifyLand(1)).to.be.revertedWith("Not an authorized entity");
});

it("Should transfer ownership", async function () {
    await landRegistry.connect(user1).registerLand("5.6037, -0.1870", "Qm1234IPFSHash");
    await landRegistry.connect(owner).addVerifiedAuthority(authority.address);
    await landRegistry.connect(authority).verifyLand(1);

    await expect(landRegistry.connect(user1).transferLand(1, user2.address))
        .to.emit(landRegistry, "OwnershipTransferred(uint256,address)")
        .withArgs(1, user2.address);

    const land = await landRegistry.getLandDetails(1);
    expect(land.owner).to.equal(user2.address);
});

it("Should not allow unverified land to be transferred", async function () {
    await landRegistry.connect(user1).registerLand("5.6037, -0.1870", "Qm1234IPFSHash");

    await expect(landRegistry.connect(user1).transferLand(1, user2.address))
        .to.be.revertedWith("Land must be verified");
});

it("Should allow only land owner to transfer ownership", async function () {
    await landRegistry.connect(user1).registerLand("5.6037, -0.1870", "Qm1234IPFSHash");
    await landRegistry.connect(owner).addVerifiedAuthority(authority.address);
    await landRegistry.connect(authority).verifyLand(1);

    await expect(landRegistry.connect(user2).transferLand(1, user2.address))
        .to.be.revertedWith("Not the landowner");
});

it("Should allow owner to add and remove verified authority", async function () {
    await landRegistry.connect(owner).addVerifiedAuthority(authority.address);
    expect(await landRegistry.verifiedAuthorities(authority.address)).to.equal(true);

    await landRegistry.connect(owner).removeVerifiedAuthority(authority.address);
    expect(await landRegistry.verifiedAuthorities(authority.address)).to.equal(false);
});
```

Figure 25. A code snippet of the Smart Contract.

5.1.2 Backend API Testing

The backend API of the Asaase Aban platform, acting as the communication bridge between the user interface, the blockchain network, and the PostgreSQL database, underwent rigorous unit and functional testing to validate its correctness, reliability, and robustness. Automated API tests were conducted using a dedicated testAPI.js script, employing the Axios HTTP client library to simulate real-world interactions with API endpoints. The testing environment included Node.js as the backend runtime, Axios for issuing programmatic HTTP requests, and Chai for selected assertion checks on response structures and error handling.

The API testing phase comprehensively covered key backend operations to ensure transaction integrity and secure data retrieval workflows. Land data retrieval endpoints, such as /api/land/lands and /api/land/:landId, were tested to confirm accurate access to land metadata and geospatial records from the database. Basic user authentication workflows were validated to ensure that API requests correctly authorized users based on their blockchain wallet addresses. End-to-end testing of land registration workflows verified that processes interfaced smoothly with blockchain smart contracts and that corresponding records were accurately updated in the database.

Additionally, data persistence tests confirmed synchronization between blockchain transactions and off-chain database records, maintaining consistency across the system layers. Negative testing scenarios assessed error handling robustness by simulating invalid inputs, unauthorized requests, and system failures, validating the proper return of HTTP status codes (400, 401, 404, 500) and informative error messages. Overall, backend API testing confirmed that all primary endpoints performed reliably, securely, and efficiently, supporting the Asaase Aban platform's goal of maintaining transaction integrity and user trust across a decentralized land administration system.

```
const registerLandOnBlockchain = async (location, documentHash) => {
    const tx = await contract.registerLand(location, documentHash);
    await tx.wait();
    return tx.hash;
};
```

Figure 26. A code snippet showing a function that registers lands on the blockchain.

5.2 Frontend-Backend Integration Testing

Frontend-backend integration testing was a critical phase in validating the Asaase Aban platform's communication flow between the user interface and backend API services. These tests ensured that user actions on the frontend reliably triggered backend operations and that appropriate data, responses, and feedback were consistently delivered, maintaining a seamless and intuitive user experience. The testing environment leveraged Vanilla JavaScript for managing event-driven interactions, Axios for handling HTTP requests and responses, and browser developer tools for validating API outputs, monitoring network traffic, and verifying UI rendering accuracy. Testing confirmed that land registration data, ownership records, and verification statuses were correctly retrieved and dynamically rendered on user dashboards and administrative panels, with map visualizations remaining synchronized with backend records. API response handling was rigorously evaluated for both successful and failed interactions, ensuring that relevant notifications or error messages were appropriately displayed. Additionally, user authentication flows were tested to verify that blockchain wallet-based authentication securely governed user-specific actions, such as land registration and ownership management. Negative testing scenarios, including server errors and unauthorized access attempts, validated the platform's robust error handling without system crashes or data corruption. Overall, integration testing confirmed that Asaase Aban delivers a resilient, secure, and user-centered platform experience through strong frontend-backend coordination.

```
export async function checkUserRole(walletAddress) {
    try {
        const response = await fetch(`http://localhost:8000/api/user/${walletAddress}`);
        if (!response.ok) {
            throw new Error(`HTTP error! status: ${response.status}`);
        }
        const userData = await response.json();
        return userData.user_role;
    } catch (error) {
        console.error("Error checking user role:", error);
        return 0;
    }
}
```

Figure 27. A code snippet showing the checkUserRole function

5.2.3 Map Integration Testing

Map integration testing was conducted to validate the accurate visualization of geospatial land data within the Assase Aban system. Given the critical role of geospatial information in land administration, it was essential to ensure that land parcel boundaries and metadata were displayed correctly, interactively, and reliably on the frontend user dashboards.

Testing Environment and Tools:

The testing of the Asaase Aban platform's map visualization components relied on a set of key tools. **Leaflet.js** was employed as the primary JavaScript library for rendering interactive maps, providing dynamic visualization of land parcels. **GeoJSON** served as the standard format for encoding geospatial data structures, enabling consistent representation of land boundaries and properties. Additionally, **browser developer tools** were utilized to monitor the parsing of GeoJSON data, assess the rendering of map layers, and verify the behavior of interactive features, ensuring a responsive and accurate user experience.

```
const geoLayer = L.geoJSON(feature, {
    style: {
        fillColor: color,
        weight: 3,
        opacity: 0.9,
        color: "#FFFFFF",
        dashArray: "5",
        fillOpacity: 0.6,
    },
    onEachFeature: (feature, layer) => {
        // Add popup with land info
        layer.bindPopup(popupContent);
    }
}).addTo(map);
```

Figure 28. A code snippet showing a design of a registered land

Testing efforts for the Asaase Aban platform focused extensively on validating the integration between the backend geospatial database and the frontend map visualization. Tests confirmed that GeoJSON data retrieved from the PostgreSQL/PostGIS database was correctly parsed and accurately mapped onto the Leaflet.js interface, ensuring that the geometry, properties, and metadata of land parcels remained consistent with backend records. Visual inspections further verified that land boundaries were precisely rendered in terms of shape, size, and positioning, with any distortions or errors systematically addressed.

Interactive map features, including information popups and hover effects, were rigorously tested to ensure that key land metadata—such as ownership details, verification status, and acreage—was reliably displayed, thereby enhancing user engagement and information accessibility. In addition, zooming, panning, and navigation functionalities were evaluated to confirm smooth, responsive performance across various map scales, preserving data fidelity in both clustered and dispersed parcel views. The successful outcomes of these tests validated the robustness, accuracy, and usability of the map visualization component, which is central to the user interaction experience within the Asaase Aban system.

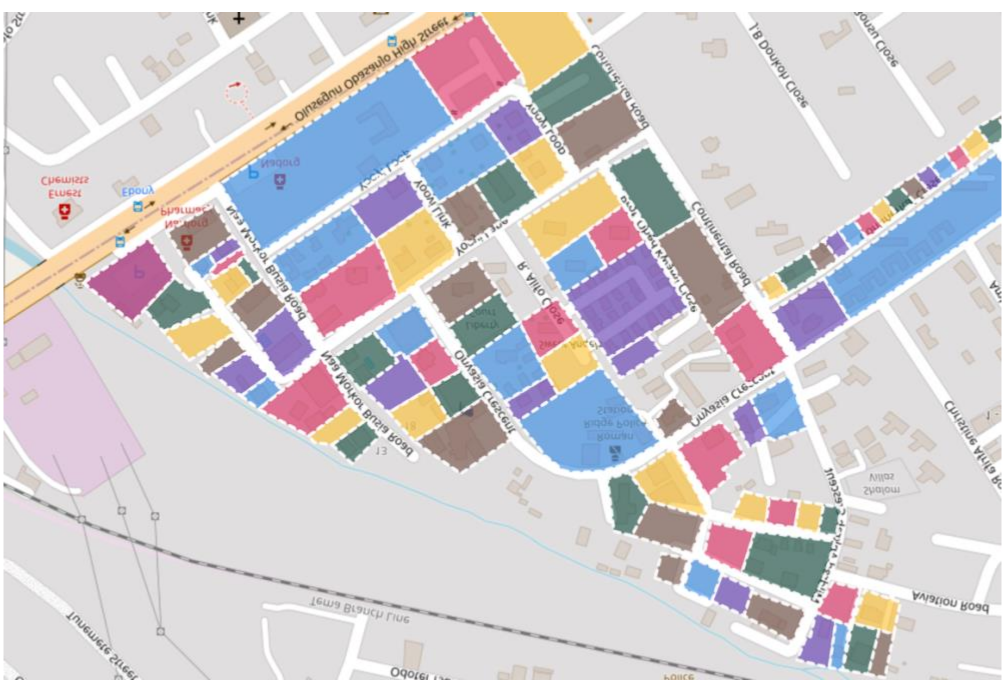

Figure 29. A picture of registered lands.

Outcomes: Map integration testing confirmed that the Assase Aban system's interactive map components provided accurate, responsive, and user-friendly visualization of land parcels. The successful parsing, rendering, and interactivity of GeoJSON data strengthened the system's spatial data representation capabilities, supporting efficient land record exploration and verification for all stakeholders.

5.3 System Testing

System testing was conducted to validate the end-to-end functionality of the Assase Aban platform, ensuring that all integrated modules, frontend, backend, blockchain, database, and external services, operated cohesively under real-world usage scenarios. The focus was on simulating complete user workflows to verify that the system met functional, performance, and security expectations across all critical land administration processes.

Testing Environment and Tools:

The Asaase Aban platform integrates several key technologies to ensure secure, scalable, and efficient land administration workflows. MetaMask, a browser-based Ethereum wallet, is utilized for user authentication and blockchain transaction signing, providing a seamless bridge between users and the decentralized network. The system's core business logic and backend services are managed through a Node.js API server, which handles interactions between the frontend, blockchain, and database components. For backend storage, a PostgreSQL database enhanced with PostGIS extensions is employed to store land metadata and user information, enabling spatial data management. Blockchain transactions are validated and recorded on the Ethereum Sepolia Testnet, ensuring data integrity and transparency during the development and testing phases.

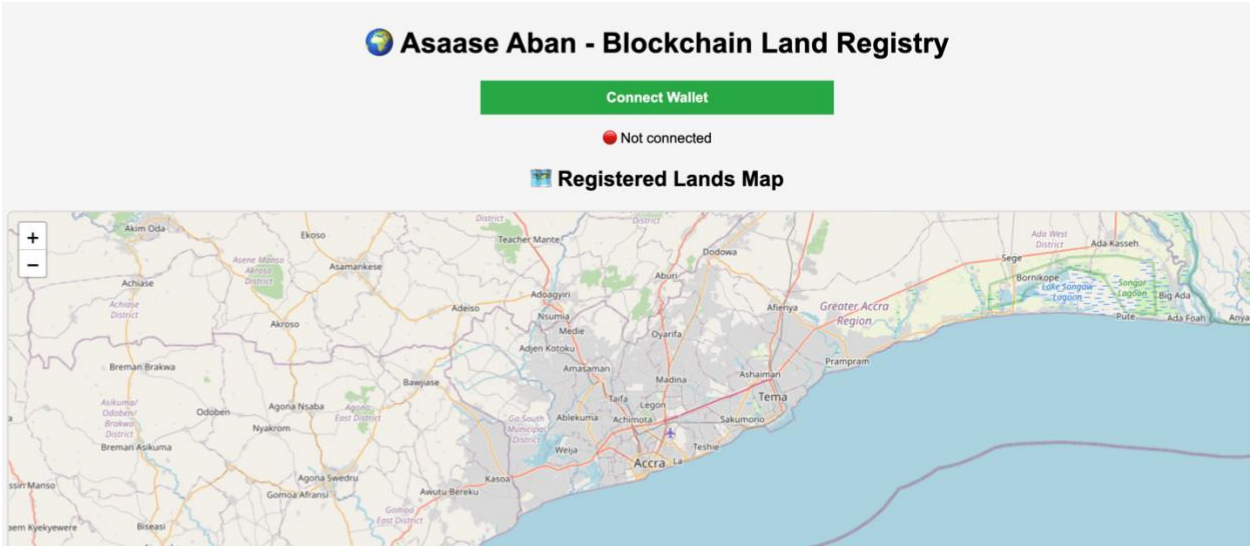

Figure 30. A picture of the landing page of AsaaseAban

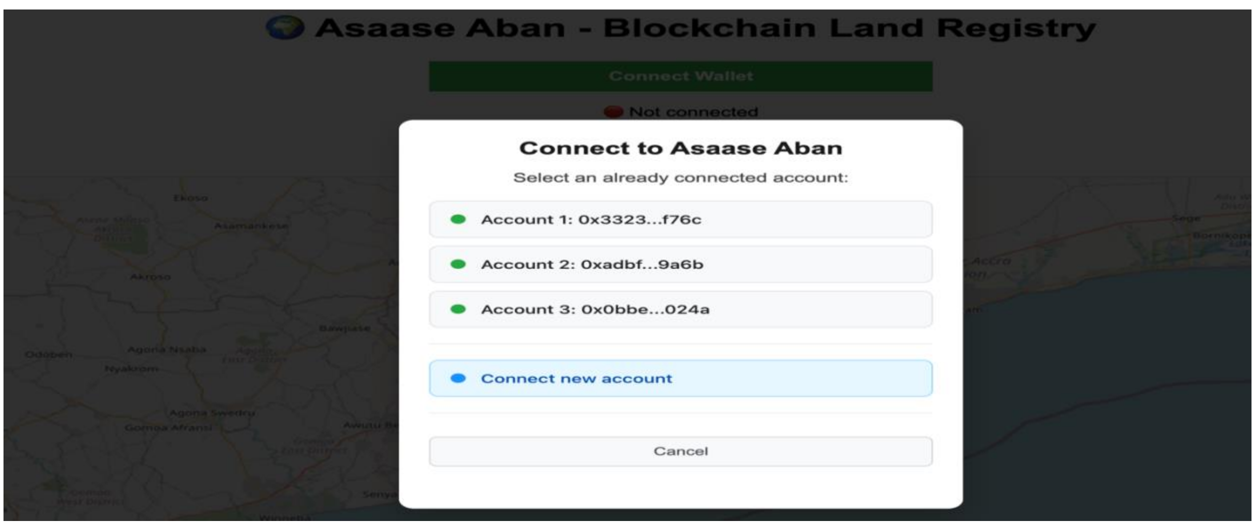

Figure 31. A picture of showing which different metamask account wallet to connect to.

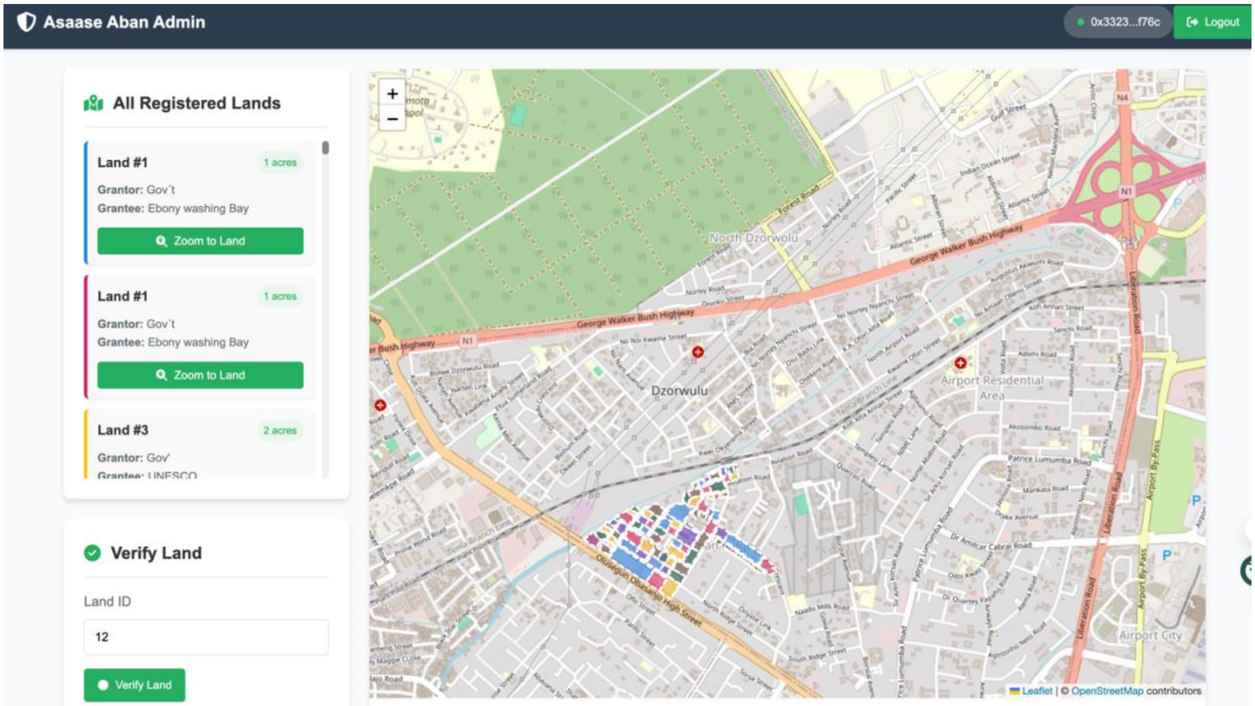

Figure 32. A picture showing the admin interface after the Admin connected the metamask wallet.

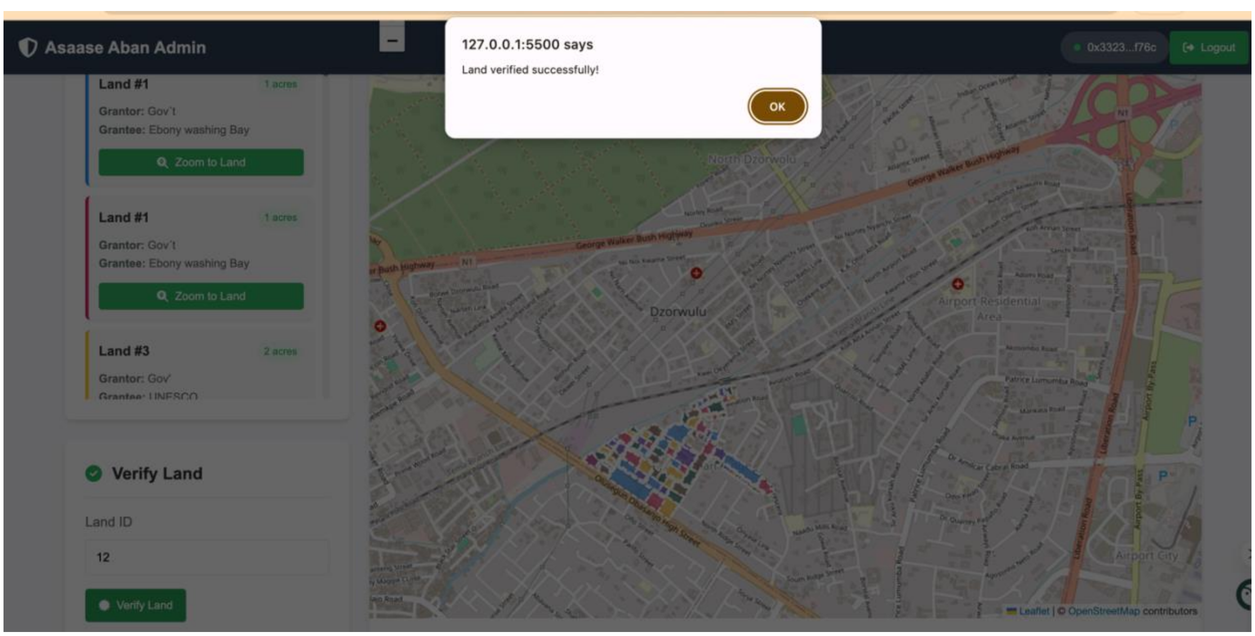

Figure 33. A picture showing Admin successfully verified a new land to complete the registration.

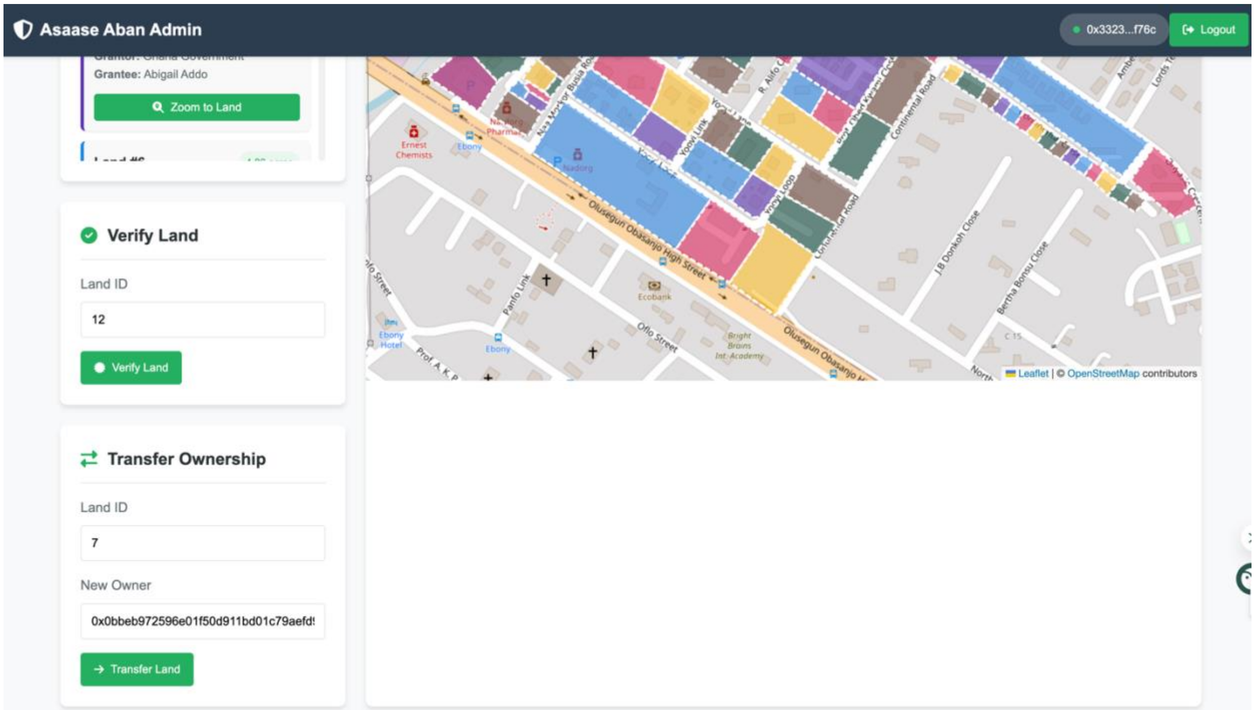

Figure 34. A picture showing a land of ID of 7 that is going to be transferred to the given address below it

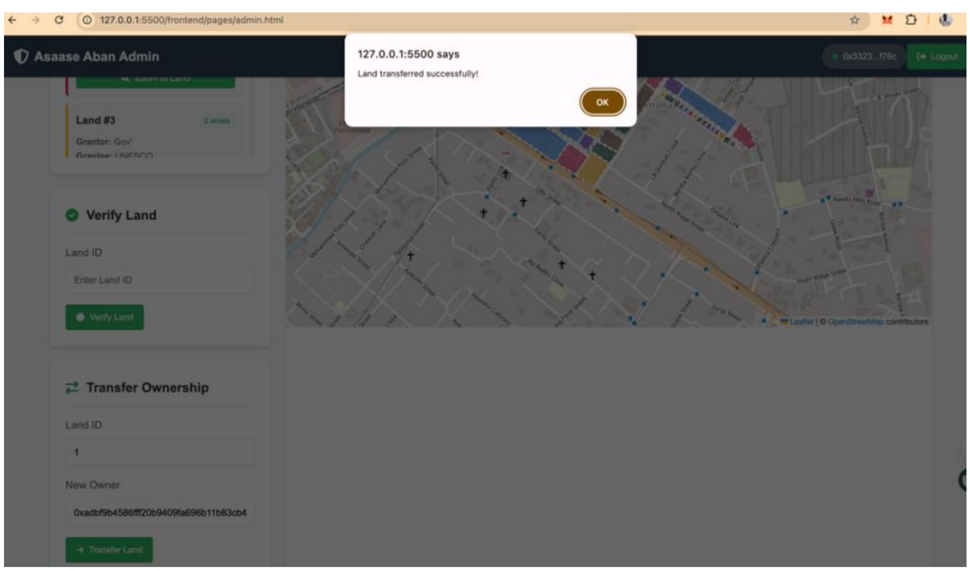

Figure 35. A picture of a land that has been successfully transferred.

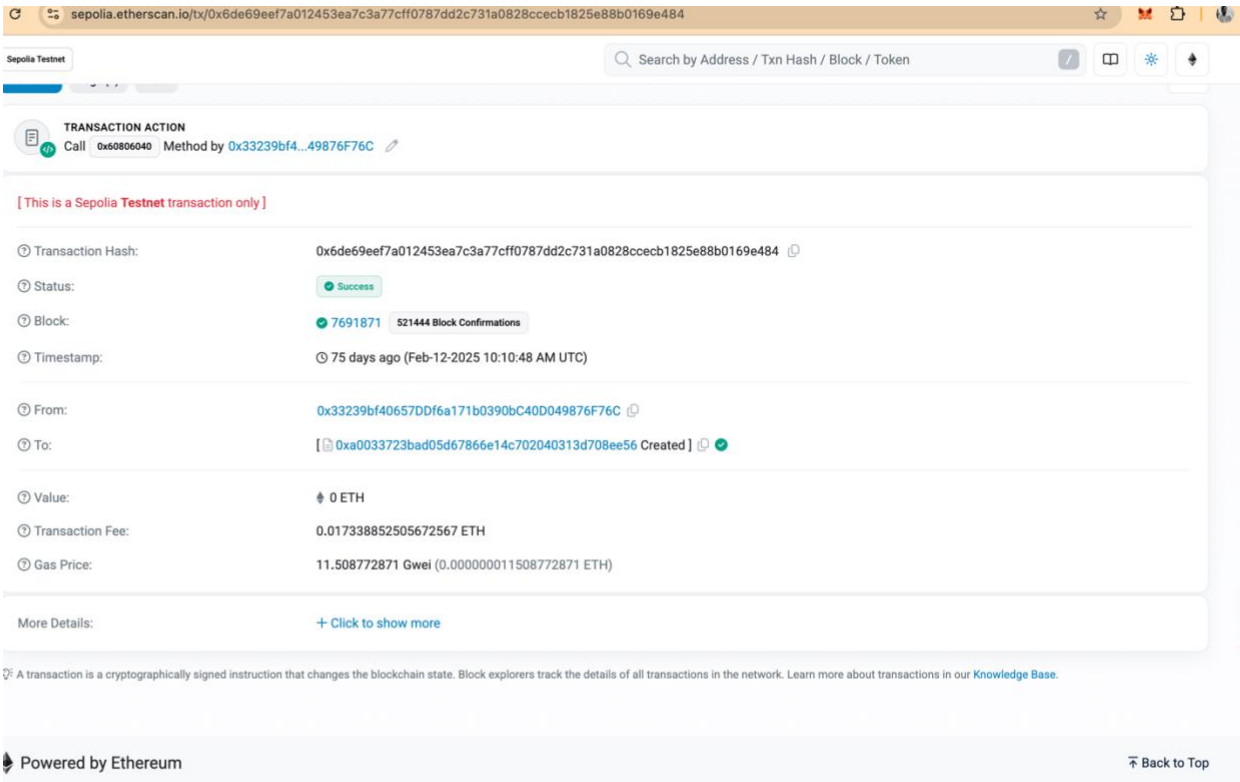

Figure 36. A picture showing the transaction of land on the blockchain using the sepolia Testnet.

System testing for the Asaase Aban platform included a comprehensive evaluation of key user workflows to ensure seamless operation and consistent synchronization between the blockchain and database components. User registration and login functionalities were carefully tested, confirming that users could successfully connect their MetaMask wallets and that public wallet addresses were accurately retrieved. Additionally, workflows related to user role assignment were validated, ensuring that administrative privileges were correctly granted to designated users and that access controls were effectively enforced based on user roles. These testing outcomes reinforced the platform's commitment to delivering a secure, intuitive, and robust user experience.

Land Registration Flow:

Testing of the data entry and registration functionalities on the Asaase Aban platform emphasized the importance of ensuring input accuracy and system integrity. Frontend input fields were rigorously tested to validate the correctness, completeness, and proper formatting of user-provided data before initiating land registration requests. Blockchain transaction testing confirmed the successful invocation of the `registerLand` smart contract function on the Ethereum Sepolia Testnet, ensuring the integrity of transactions recorded on-chain.

Additionally, backend validation confirmed that corresponding land metadata was accurately created and stored in the PostgreSQL database. This synchronization between blockchain and off-chain records ensures data consistency, reinforcing the platform's reliability and supporting seamless land registration workflows.

Land Verification:

Testing of the administrative functionalities within the Asaase Aban platform focused on validating land parcel verification processes and document authenticity checks. The admin dashboard interface was tested to ensure that administrators could efficiently select and verify land parcels. Additionally, the system's integration with IPFS was evaluated, confirming that land-related documents could be reliably retrieved and validated for authenticity.

Further testing verified that once a land parcel was successfully verified, its status was accurately updated both on-chain via smart contracts and off-chain within the PostgreSQL database. These results demonstrate the platform's ability to maintain consistency between blockchain and database records, ensuring transparency, authenticity, and reliable land status management through the administrative workflows.

Land Transfer:

System testing for the Asaase Aban platform focused on validating the ownership transfer processes and ensuring seamless data synchronization across all system components. Ownership transfer workflows were tested through smart contract function calls to confirm that property rights could be securely and accurately reassigned between users. Further testing verified that successful transfers resulted in synchronized updates between the blockchain ledger and the PostgreSQL database, maintaining consistency and data integrity across platforms.

The outcomes of the testing phase affirmed that the Asaase Aban application achieved full end-to-end integration. Core user workflows, covering land registration, management, verification, and ownership transfer, were executed reliably, demonstrating the system's ability to maintain transactional integrity and deliver a smooth user experience. These results confirmed that the platform effectively meets its objectives of decentralized, transparent, and efficient land administration.

.

5.4 Browser and Device Compatibility

The application was tested across multiple environments:

Table 1. *Browser and Device Compatibility*

Browser	**Desktop**	**Mobile**	**Results**
Chrome	✓	✓	Fully functional
Firefox	✓	✓	Fully functional
Safari	✓	⚠️	Minor UI issues on older iOS
Edge	✓	✓	Fully functional

5.5 User Acceptance Testing

User Acceptance Testing (UAT) served as a critical final phase in validating the Asaase Aban platform, ensuring that it met the functional requirements, usability standards, and operational workflows of its intended users. The primary objective was to collect direct feedback from key stakeholder groups, evaluate system effectiveness from a user-centric perspective, and identify areas for further refinement before broader deployment.

The UAT participants included land administration officials, individual property owners, and real estate professionals. Land administration officials tested the system's verification and registry features, while property owners engaged with land registration, ownership verification, and transfer workflows. Real estate professionals assessed functionalities related to land record retrieval, mapping, and property verification processes.

Testing activities covered core areas of the platform, such as land registration, ownership verification, land transfers, and map-based search and visualization features. Both guided walkthroughs and independent testing sessions were conducted to ensure comprehensive feedback. Insights gathered from UAT informed critical system improvements, strengthening Asaase Aban's readiness for large-scale adoption in the land governance ecosystem.

Key Findings and Feedback:

User Acceptance Testing (UAT) for the Asaase Aban platform yielded valuable insights that informed final refinements and prioritized future enhancements. Key findings revealed high user satisfaction with the system's dynamic map visualization features powered by Leaflet.js, with participants particularly appreciating the spatial display of land boundaries, ownership details, and verification statuses. Additionally, users strongly valued the transparency provided by blockchain verification, highlighting the tamper-proof recording of land transactions on Ethereum as a significant trust-enhancing feature.

However, participants also identified areas for improvement. Many suggested the addition of advanced filtering options within the dashboard to facilitate more precise searches based on criteria such as location, verification status, acreage, and transaction history. Furthermore, mobile device users noted minor usability challenges, recommending improved mobile responsiveness and more touch-friendly interface elements to enhance accessibility on smaller screens.

Overall, the UAT phase validated that Asaase Aban meets core user expectations regarding functionality, transparency, and ease of use. The feedback gathered served as a critical guide for prioritizing post-deployment feature improvements, ensuring that the platform continues to evolve in alignment with real-world user needs and experiences.

5.6 Testing Challenges and Solutions

Despite the overall success of the testing and validation phase, the Assase Aban system encountered several technical challenges that required iterative problem-solving and adaptation. Addressing these challenges was critical to ensuring system robustness, resilience, and a consistent user experience across different operational scenarios.

5.6.1 GeoJSON Data Parsing

During testing, Asaase Aban faced challenges in parsing GeoJSON data from the PostGIS database due to variations in coordinate systems, missing fields, and formatting inconsistencies, which occasionally disrupted map rendering. To solve this, a robust parsing strategy was introduced, including support for multiple GeoJSON formats, validation checks for malformed data, and a fallback mechanism that displayed parcel metadata when visual rendering failed. These enhancements greatly improved the system's resilience to geospatial data inconsistencies, ensuring a reliable and accessible user experience.

```
// Parse the GeoJSON string into an object
let geojsonData;
try {
    if (typeof land.geojson === "object") {
        geojsonData = land.geojson;
    } else {
        geojsonData = JSON.parse(land.geojson);
    }
} catch (e) {
    console.error("Failed to parse GeoJSON:", e, land.geojson);
    createLandListItem(land, index, allLandsList);
    return;
}
```

Figure 37. A code snippet showing a challenge encountered when Geojson data was trying to retrieve data form the Postgis database.

5.7 Test Results Summary

Table 2.

Test Results Summary

Component	Test Cases	Pass Rate	Notes
Smart Contract	12	100%	All security controls validated
Backend API	8	95%	Minor issue with error handling
Frontend UI	15	93%	Responsive design improvements needed
Database	5	100%	Performance optimizations identified
Map Integration	7	90%	GeoJSON parsing improvements made

Chapter 6: Conclusion

6.1 Project Summary

The Asaase Aban project developed a blockchain-based land administration system to improve transparency, security, and efficiency in land governance. By combining Ethereum smart contracts, a PostGIS-enabled database, and a responsive web interface, the platform provides a robust, scalable solution for land registration, ownership verification, and property transfers, effectively addressing the complexities of Ghana's dual land tenure system.

Key Features of the Asaase Aban System:

The Asaase Aban project successfully developed a blockchain-based land administration system tailored to address transparency, security, and efficiency challenges in traditional land governance, especially within Ghana's dual tenure system. Leveraging Ethereum smart contracts, the platform securely records land transactions, ensuring data immutability. It features an interactive map powered by Leaflet.js using GeoJSON data for dynamic visualization of land parcels, and a user-friendly web interface with MetaMask integration for intuitive user access. A Node.js and Express.js backend connects the frontend, blockchain, and PostgreSQL/PostGIS database, creating a scalable and modular infrastructure. Role-based access control ensures system security by distinguishing between administrators and landowners. Through these innovations, Asaase Aban showcases how blockchain and decentralized technologies can transform land governance, promote economic development, and build stakeholder trust in emerging economies.

6.2 Recommendation

Based on the Asaase Aban project's development, testing, and user feedback, several recommendations are proposed to strengthen the system's usability, robustness, and stakeholder engagement. Key improvements include enhancing mobile responsiveness for better smartphone and tablet access, expanding GIS data support to accommodate formats like KML and Shapefiles, and improving error handling and logging for smoother diagnostics. Implementing structured user training programs would help users better understand blockchain-based processes, while stronger data validation mechanisms would ensure higher data quality during land transactions. By addressing these areas, Asaase Aban can evolve into an even more scalable, reliable, and user-centered platform for modernizing land administration through blockchain technology.

6.3 Limitations and Challenges

While the Asaase Aban project achieved its core objectives, several challenges emerged during development and testing. Blockchain transaction delays on the Ethereum testnet affected user workflows, highlighting the need for better UI feedback. Issues with inconsistent GeoJSON data and initial PostgreSQL schema constraints required additional validation and revisions. Limited browser compatibility, particularly with older Safari versions, caused UI inconsistencies, while user adoption was hindered by limited familiarity with blockchain technology. Addressing these technical and adoption challenges through improvements and education will help Asaase Aban evolve into a more resilient and accessible land administration platform.

6.4 Future Work

The successful implementation of the Asaase Aban system marks a major step in modernizing land administration in Ghana, but several future enhancements are proposed to strengthen its impact. Key areas for development include optimizing system performance by reducing blockchain transaction latency and database query times to enhance user responsiveness. Integrating Asaase Aban with national land administration systems is also crucial for centralized oversight, reducing data duplication, and encouraging broader adoption.

User experience can be improved through advanced search and filtering features, while accessibility would be expanded by developing a mobile app for Android and iOS, allowing landowners and administrators to access services on the go. To increase flexibility and security, supporting multiple blockchain networks is recommended. Additionally, integrating artificial intelligence could provide powerful insights, enabling predictive analytics for land valuation and automated anomaly detection in transactions.

By pursuing these enhancements, Asaase Aban can evolve into a more powerful, scalable, and intelligent platform, further transforming land governance and fostering trust, efficiency, and innovation in Ghana's land administration system.

6.5 Conclusion

The Asaase Aban project highlights the transformative potential of blockchain technology in addressing critical challenges in land administration, such as fraud, inefficiency, and data insecurity. By combining blockchain's transparency with a robust backend and user-friendly interface, the system offers a resilient and scalable solution. Overcoming technical hurdles through iterative development, Asaase Aban has laid a strong foundation for future innovations. With further enhancements like mobile access, advanced GIS integration, and national system collaboration, it is well-positioned to revolutionize land management in Ghana and beyond, promoting secure, transparent, and efficient land transactions.

References

[1] "(PDF) Toward Smart Land Management: Land Acquisition and the Associated Challenges in Ghana. A Look into a Blockchain Digital Land Registry for Prospects." Accessed December 12, 2024. https://www.researchgate.net/publication/349686944_Toward_Smart_Land_Management_Land_Acquisition_and_the_Associated_Challenges_in_Ghana_A_Look_into_a_Blockchain_Digital_Land_Registry_for_Prospects.

[2] "(PDF) Skin Lands in Ghana and Application of Blockchain Technology for Acquisition and Title Registration." ResearchGate, October 22, 2024. https://doi.org/10.1108/JPPEL-12-2019-0062.

[3] Al-dulaimy, Tiba W., and Saad N. Alsaad. "Blockchain-Based Secured Estate Property Registration System: Baghdad Estate Registration as a Case Study," 2024. https://doi.org/10.1007/978-3-031-62871-9_12.

[4] "Physical Access Control Management System Based on Permissioned Blockchain | Request PDF." In ResearchGate. Accessed December 12, 2024. https://www.researchgate.net/publication/327781133_Physical_Access_Control_Management_System_Based_on_Permissioned_Blockchain.

[5] "Land Registry System Using Blockchain Technology." In *ResearchGate*, 2024. https://doi.org/10.1109/INNOCOMP63224.2024.00035.

[6] Mintah, Kwabena, Festival Godwin Boateng, Kingsley Tetteh Baako, Eric Gaisie, and Gideon Kwame Otchere. "Blockchain on Stool Land Acquisition: Lessons from Ghana For." Accessed December 12, 2024. https://ideas.repec.org//a/eee/lauspo/v109y2021ics0264837721003586.html.

[7] "New Technology Interventions Including Blockchain Technology in Land Record and Registry Management in India | Request PDF." In *ResearchGate*. Accessed December 12, 2024. https://doi.org/10.1145/3428502.3428521.

[8] ResearchGate. "A Proof of Concept of the Integration of Blockchain with an ISO 19152:2012 Based Land Administration System | Request PDF," December 9, 2024. https://doi.org/10.1145/3581971.3581984.

[9] “(PDF) Smart Contract Application for Managing Land Administration System Transactions.” *ResearchGate*, December 9, 2024. https://doi.org/10.1109/ACCESS.2022.3164444.

[10] Ameyaw, Prince Donkor, and Walter Timo de Vries. “Toward Smart Land Management: Land Acquisition and the Associated Challenges in Ghana. A Look into a Blockchain Digital Land Registry for Prospects.” *Land* 10, no. 3 (March 1, 2021): 239. https://doi.org/10.3390/land10030239.

[11] “A Blockchain-Based Land Titling Project in the Republic of Georgia: Rebuilding Public Trust and Lessons for Future Pilot Projects.” *ResearchGate*, October 22, 2024. https://doi.org/10.1162/inov_a_00276.

[12] “Sweden Tests Blockchain Technology for Land Registry | Reuters.” Accessed December 12, 2024. https://www.reuters.com/article/us-sweden-blockchain/sweden-tests-blockchain-technology-for-landregistry-idUSKCN0Z22KV/.

[13] Mintah, Frank, Clifford Amoako, and Kwasi Kwafo Adarkwa. “The Fate of Urban Wetlands in Kumasi: An Analysis of Customary Governance and Spatio-Temporal Changes.” *Land Use Policy* 111 (December 1, 2021): 105787. https://doi.org/10.1016/j.landusepol.2021.105787.

[14] Shinde, Disha, Snehal Padekar, Siddharth Raut, Abdul Wasay, and S. S. Sambhare. “Land Registry Using Blockchain - A Survey of Existing Systems and Proposing a Feasible Solution.” *2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, September 2019, 1–6. https://doi.org/10.1109/ICCUBEA47591.2019.9129289.

[15]

[16] "Blockchain technology for land tenure security and agricultural financing in rural Nigeria and Benin," accessed April 26, 2025. [Online]. Available: https://cwas.uestc.edu.cn/info/1042/3558.htm

[17] M. A. Mansoor, M. Ali, A. Mateen, M. Kaleem, and S. Nazir, "Blockchain technology for land registry management in developing countries," *2023 2nd International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE)*, Lahore, Pakistan, Nov. 27, 2023, IEEE, 2023. [Online]. Available: https://doi.org/10.1109/ETECTE59617.2023.10396736