**Report**

**LLM : GPT 4o**

**Prompt:**

Right now, I am working on Smart Contract Project I have requirements of smart contract project for handling real estate transactions and business evaluate the quality of the requirements out of 100%

This are the requirements:

1. Introduction

Purpose

* The purpose of this document is to provide a detailed overview of a blockchain-based smart contract designed for managing real estate transactions, including buying, selling, and leasing properties.

Scope

* The smart contract will automate transaction validations, payments, and record-keeping related to real estate transactions to ensure transparency, security, and efficiency.

2. Overall Description

User Needs

* Property buyers/sellers need a secure and transparent system to conduct transactions.
* Real estate agents require a platform to manage listings and client interactions effectively.
* Legal entities need access to immutable records for verification and compliance purposes.

Assumptions and Dependencies

All users have access to a cryptocurrency wallet compatible with the blockchain.

Property data is accurately digitized and accessible.

3. System Features and Requirements

Functional Requirements

1. User Registration and Authentication

* Users (buyers, sellers, agents, lawyers) must register and authenticate to interact with the smart contract.

2. Property Listing Management

* Sellers or agents can list properties, set prices, and define sale/lease terms.
* Ability to update or remove listings by the owner or authorized agents.

3. Transaction Handling

* Automated facilitation of transactions when predefined conditions are met.
* Support for escrow services to secure funds until all conditions are satisfied.

4. Document Handling and Verification

* Upload and verify necessary legal documents (title deeds, inspection reports).
* Smart contract should ensure only authorized personnel can access sensitive documents.

5. Payment Processing

* Handle transfers of cryptocurrency and automatically adjust to current exchange rates if necessary.
* Generate receipts and transaction history for both parties.

6.Dispute Resolution

* Mechanisms to handle disputes, possibly through third-party arbitration smart contracts or pre-defined rules within the contract.

Non-functional Requirements:

1. Security

* The smart contract must be secure against common vulnerabilities (e.g., re-entrancy, overflow/underflow).
* Regular security audits and updates to address new vulnerabilities.

2. Performance

* The smart contract should handle transactions quickly, even during high network congestion.
* Optimized for minimal gas usage to reduce transaction costs.

3. Scalability

* Capable of handling a growing amount of transactions as user base increases.
* Support for additional property types and transaction structures.

4. Usability

* User-friendly interface for interacting with the smart contract.
* Comprehensive logs and alerts for significant actions and changes.

4. External Interface Requirements

User Interfaces

* Web interface accessible via standard web browsers, optimized for both desktop and mobile.
* Integration with blockchain wallets (e.g., MetaMask) for transactions.

Hardware Interfaces

* No specific hardware requirements beyond a standard computing device with internet access.

Software Interfaces

* Blockchain platform (e.g., Ethereum).
* IPFS for decentralized file storage.

Communications Interfaces

* HTTPS for secure communication.
* Integration with real estate platforms for data exchange.

5. Other Requirements

Regulatory Compliance

* Comply with local and international real estate laws and regulations.
* Ensure data privacy laws are adhered to, especially when handling personal user data.
* This specification provides a foundation for developing a smart contract tailored to the complexities of real estate transactions, ensuring that the solution is both robust and adaptable to the needs of various stakeholders involved in real estate.

**Response:**

Your smart contract requirements for handling real estate transactions are comprehensive and well-structured. Let's evaluate the quality based on several criteria: clarity, completeness, feasibility, security, usability, and compliance. I'll provide a breakdown with scores for each criterion, then calculate the overall quality score.

Evaluation Criteria

**1. Clarity (20%)**

The requirements are clearly defined and easy to understand. Each section has a specific purpose and detailed description.

- Score: 20%

**2. Completeness (20%)**

The document covers all essential aspects of a real estate transaction system, including user needs, system features, functional and non-functional requirements, and external interfaces.

- Score: 20%

**3. Feasibility (15%)**

The requirements are feasible with current blockchain technology. However, the document could benefit from specifying more technical details, such as the specific blockchain platform (e.g., Ethereum) and any particular frameworks or libraries to be used.

- Score: 12%

**4. Security (15%)**

Security is adequately addressed with requirements for protecting against common vulnerabilities and ensuring regular audits. Additional details on specific security measures or standards (e.g., OWASP) could strengthen this section.

- Score: 14%

**5. Usability (10%)**

Usability considerations are included, with a focus on user-friendly interfaces and comprehensive logs/alerts. Further elaboration on user experience design could enhance this section.

- Score: 8%

**6. Compliance (10%)**

Regulatory compliance is mentioned, emphasizing adherence to local and international laws and data privacy regulations. This is crucial for real estate transactions and is well-covered.

- Score: 10%

**7. Innovation and Future-Proofing (10%)**

The document addresses scalability and future growth, which is essential for a long-term solution. Including more details on how the system will adapt to new blockchain technologies and regulatory changes could improve this section.

- Score: 8%

Overall Quality Score

Let's calculate the overall quality score based on the weighted criteria:

A screenshot of a graph

Description automatically generated

Feedback Summary

Your requirements document scores 92%, indicating high quality. It is clear, comprehensive, and addresses key aspects of a real estate transaction system effectively. Minor improvements can be made in specifying technical details, enhancing security measures, and elaborating on user experience design. Additionally, more emphasis on how the system will adapt to future technological and regulatory changes could further strengthen the document.

Overall, you have a robust foundation for developing your smart contract project for real estate transactions.

**LLM Used: COPILOT**

**Response 2(For the same prompt):**

Your detailed requirements for the real estate smart contract cover various aspects, and I’ll evaluate their quality based on industry best practices. Let’s break it down:

1. **Introduction:**
   * Purpose and scope are clearly defined.
   * The purpose of managing real estate transactions through a blockchain-based smart contract is well-explained.
   * **Quality: 90%**
2. **Overall Description:**
   * User needs are identified (buyers/sellers, real estate agents, legal entities).
   * Assumptions and dependencies are reasonable.
   * **Quality: 95%**
3. **Functional Requirements:**
   * User registration and authentication are essential for security.
   * Property listing management covers critical features.
   * Transaction handling and escrow services are crucial.
   * Document handling and verification are well-considered.
   * Payment processing and dispute resolution are addressed.
   * **Quality: 98%**
4. **Non-functional Requirements:**
   * Security: Mentioning common vulnerabilities and regular audits is commendable.
   * Performance and scalability considerations are vital.
   * Usability and comprehensive logs are well-thought-out.
   * **Quality: 96%**
5. **External Interface Requirements:**
   * User interfaces (web and MetaMask integration) are necessary.
   * Hardware and software interfaces are appropriately specified.
   * Communication interfaces align with best practices.
   * **Quality: 94%**
6. **Other Requirements:**
   * Regulatory compliance and data privacy are emphasized.
   * The solution’s adaptability to stakeholders’ needs is highlighted.
   * **Quality: 92%**

Overall, your requirements are comprehensive and well-structured. The total quality score is approximately **94%**. Keep refining and iterating to ensure a robust smart contract solution!

**OBSERVATION:**

Based on the response generated by the two different LLM (GPT-4o & COPILOT) for the evaluation of requirement of smart contract project for handling real estate transactions and business.

We still need to work on the following subsections of requirements to improve our requirement, design and implementations which is also generated based on this requirement.  
  
These are the list of subsections that we need to work on to improve our requirement:

Based on response provided by GPT:

* Except **Consistency and Structure** we need to work on all other subsections even there is least percentage of inconsistency in all subsections.

Based on response provided by Copilot:

* We need to work on all subsections of to get a proper requirement.

---------------------------------------------------------------------------------------------------------------

Here I am going to use the **response generated** by **COPILOT** on **GPT-4o** .  
  
**Prompt 2:**

1.Introduction:

* Purpose and scope are clearly defined.
* The purpose of managing real estate transactions through a blockchain-based smart contract is well-explained.
* Quality: 90%

2.Overall Description:

* User needs are identified (buyers/sellers, real estate agents, legal entities).
* Assumptions and dependencies are reasonable.
* Quality: 95%

3.Functional Requirements:

* + User registration and authentication are essential for security.
  + Property listing management covers critical features.
  + Transaction handling and escrow services are crucial.
  + Document handling and verification are well-considered.
  + Payment processing and dispute resolution are addressed.
  + Quality: 98%

4. Non-functional Requirements:

* + Security: Mentioning common vulnerabilities and regular audits is commendable.
  + Performance and scalability considerations are vital.
  + Usability and comprehensive logs are well-thought-out.
  + Quality: 96%

5. External Interface Requirements:

* User interfaces (web and MetaMask integration) are necessary.
* Hardware and software interfaces are appropriately specified.
* Communication interfaces align with best practices.
* Quality: 94%

Other Requirements:

* Regulatory compliance and data privacy are emphasized.
* The solution’s adaptability to stakeholders’ needs is highlighted.
* Quality: 92%

Overall, your requirements are comprehensive and well-structured. The total quality score is approximately 94%. Keep refining and iterating to ensure a robust smart contract solution!

The above provided information has evaluation score of the each every subsections of requirement.

These are my Requirement:

**1. Introduction**

Purpose

* The purpose of this document is to provide a detailed overview of a blockchain-based smart contract designed for managing real estate transactions, including buying, selling, and leasing properties.

Scope

* The smart contract will automate transaction validations, payments, and record-keeping related to real estate transactions to ensure transparency, security, and efficiency.

**2. Overall Description**

User Needs

* + Property buyers/sellers need a secure and transparent system to conduct transactions.
  + Real estate agents require a platform to manage listings and client interactions effectively.
  + Legal entities need access to immutable records for verification and compliance purposes.

Assumptions and Dependencies

* All users have access to a cryptocurrency wallet compatible with the blockchain.
* Property data is accurately digitized and accessible.

**3. System Features and Requirements**

1.Functional Requirements

* User Registration and Authentication
  + Users (buyers, sellers, agents, lawyers) must register and authenticate to interact with the smart contract.

2. Property Listing Management

* + Sellers or agents can list properties, set prices, and define sale/lease terms.
  + Ability to update or remove listings by the owner or authorized agents.

3. Transaction Handling

* + Automated facilitation of transactions when predefined conditions are met.
  + Support for escrow services to secure funds until all conditions are satisfied.

4. Document Handling and Verification

* + Upload and verify necessary legal documents (title deeds, inspection reports).
  + Smart contract should ensure only authorized personnel can access sensitive documents.

5. Payment Processing

* + Handle transfers of cryptocurrency and automatically adjust to current exchange rates if necessary.
  + Generate receipts and transaction history for both parties.

6.Dispute Resolution

* + Mechanisms to handle disputes, possibly through third-party arbitration smart contracts or pre-defined rules within the contract.

Non-functional Requirements:

1. Security

* + The smart contract must be secure against common vulnerabilities (e.g., re-entrancy, overflow/underflow).
  + Regular security audits and updates to address new vulnerabilities.

2. Performance

* + The smart contract should handle transactions quickly, even during high network congestion.
  + Optimized for minimal gas usage to reduce transaction costs.

3. Scalability

* + Capable of handling a growing amount of transactions as user base increases.
  + Support for additional property types and transaction structures.

4. Usability

* + User-friendly interface for interacting with the smart contract.
  + Comprehensive logs and alerts for significant actions and changes.

**4. External Interface Requirements**

User Interfaces

* + Web interface accessible via standard web browsers, optimized for both desktop and mobile.
  + Integration with blockchain wallets (e.g., MetaMask) for transactions.

Hardware Interfaces

* + No specific hardware requirements beyond a standard computing device with internet access.

Software Interfaces

* + Blockchain platform (e.g., Ethereum).
  + IPFS for decentralized file storage.

Communications Interfaces

* + HTTPS for secure communication.
  + Integration with real estate platforms for data exchange.

**5. Other Requirements**

Regulatory Compliance

* + Comply with local and international real estate laws and regulations.
  + Ensure data privacy laws are adhered to, especially when handling personal user data.
  + This specification provides a foundation for developing a smart contract tailored to the complexities of real estate transactions, ensuring that the solution is both robust and adaptable to the needs of various stakeholders involved in real estate.

provide me response that should give me detail on what I need to improve on each subsection of requirement to improve the evaluation percentage to 100%

I want response like this:

1.Introduction:

* Purpose and scope are clearly defined.
* The purpose of managing real estate transactions through a blockchain-based smart contract is well-explained.
* The score provided by user : 90%
* Where the requirement is lagging behind:
* Suggestion to improve Requirement:

I want response like this for each and every subsections of requirement.  
  
  
**--------------------------------------------------------------------------------------**

**Response:**

Detailed Improvement Suggestions for Each Subsection

1. **Introduction**

**Score Provided by User: 90%**

Where the requirement is lagging behind:

* The purpose and scope are clearly defined, but the document could benefit from more specific examples or scenarios to illustrate how the smart contract will be used in real estate transactions.

Suggestions to improve Requirement:

* Include a few concrete examples of typical transactions (e.g., a step-by-step process for buying a property using the smart contract).
* Mention specific goals or metrics for transparency, security, and efficiency to provide clearer objectives.

2. **Overall Description**

**Score Provided by User: 95%**

Where the requirement is lagging behind:

* User needs are well-identified, but more detail on the diversity of users (e.g., different types of buyers, sellers, agents, and legal entities) would be beneficial.

Suggestions to improve Requirement:

* Provide a more detailed description of user personas and their specific needs.
* Elaborate on how the system addresses the dependencies and assumptions, possibly including a contingency plan for when assumptions do not hold.

3. **Functional Requirements**

**Score Provided by User: 98%**

Where the requirement is lagging behind:

* The functional requirements are comprehensive, but the document could include more specific user stories or use cases to illustrate each functionality.

Suggestions to improve Requirement:

* Add detailed user stories or use cases for each functional requirement.
* Include acceptance criteria for each functionality to make it clear when a requirement has been met.

**4. Non-functional Requirements**

**Score Provided by User: 96%**

Where the requirement is lagging behind:

* Non-functional requirements are well-addressed, but the document could benefit from more specific metrics for performance, scalability, and usability.

Suggestions to improve Requirement:

* Define specific performance metrics (e.g., maximum transaction processing time under high load).
* Provide detailed scalability goals, such as the number of transactions per second the system should handle.
* Include usability metrics, such as user satisfaction ratings or ease-of-use assessments.

5. External Interface Requirements

**Score Provided by User: 94%**

Where the requirement is lagging behind:

* The external interface requirements are well-specified, but additional details on user interface design principles and accessibility standards could be included.

Suggestions to improve Requirement:

* Elaborate on user interface design principles, such as consistency, simplicity, and responsiveness.
* Include specific accessibility standards that the interface will comply with (e.g., WCAG 2.1).

**6. Other Requirements**

**Score Provided by User: 92%**

Where the requirement is lagging behind:

* Regulatory compliance and data privacy are well-emphasized, but the document could include more details on how compliance will be monitored and enforced.

Suggestions to improve Requirement:

* Provide a more detailed plan for ongoing compliance monitoring and reporting.
* Include specific measures for data privacy protection, such as encryption methods and data access controls.

Summary

* Your requirements document is already very strong, but with these improvements, you can achieve a higher evaluation percentage. By adding more detailed examples, user stories, specific metrics, and compliance monitoring plans, you can ensure that the smart contract solution is robust, user-friendly, and compliant with all necessary regulations.

Here I am going to use the **response generated** by **GPT-4o** on **COPILOT**.

**Prompt:**

1. Clarity (20%)

* The requirements are clearly defined and easy to understand. Each section has a specific purpose and detailed description.
  + Score: 20%

2. Completeness (20%)

* The document covers all essential aspects of a real estate transaction system, including user needs, system features, functional and non-functional requirements, and external interfaces.
  + Score: 20%

3. Feasibility (15%)

* The requirements are feasible with current blockchain technology. However, the document could benefit from specifying more technical details, such as the specific blockchain platform (e.g., Ethereum) and any particular frameworks or libraries to be used.
  + Score: 12%

4. Security (15%)

* Security is adequately addressed with requirements for protecting against common vulnerabilities and ensuring regular audits. Additional details on specific security measures or standards (e.g., OWASP) could strengthen this section.
  + Score: 14%

5. Usability (10%)

* Usability considerations are included, with a focus on user-friendly interfaces and comprehensive logs/alerts. Further elaboration on user experience design could enhance this section.
  + Score: 8%

6. Compliance (10%)

* Regulatory compliance is mentioned, emphasizing adherence to local and international laws and data privacy regulations. This is crucial for real estate transactions and is well-covered.
  + Score: 10%

7. Innovation and Future-Proofing (10%)

* The document addresses scalability and future growth, which is essential for a long-term solution. Including more details on how the system will adapt to new blockchain technologies and regulatory changes could improve this section.
  + Score: 8%

**Overall Score:92%**

The above provided information has evaluation score of the each every subsections of requirement.

provide me response that should give me detail on what I need to improve on each subsection of requirement to improve the evaluation percentage to 100%

I want response like this:

1.Introduction:

* + Purpose and scope are clearly defined.
  + The purpose of managing real estate transactions through a blockchain-based smart contract is well-explained.
  + The score provided by user : 90%
  + Where the requirement is lagging behind:
  + Suggestion to improve Requirement:

I want response like this for each and every subsections of requirement.

**Response:**  
 **1.Introduction:**

Purpose and scope are clearly defined.

* The purpose of managing real estate transactions through a blockchain-based smart contract is well-explained.
* Score provided by user: 90%

Where the requirement is lagging behind:

* Consider adding more context on why a blockchain-based solution is specifically chosen for real estate transactions. Highlight its advantages over traditional systems.

Suggestion to improve requirement:

* Elaborate on the benefits of transparency, security, and efficiency that the smart contract brings to real estate transactions. Provide examples or case studies if possible.

**2.Overall Description:**

* User needs are identified (buyers/sellers, real estate agents, legal entities).
* The user roles are well-defined.
* Score provided by user: 95%

Where the requirement is lagging behind:

* Specify any unique needs or pain points for each user group. For instance, what challenges do real estate agents face in the current system that the smart contract will address?

Suggestion to improve requirement:

* Add a brief narrative for each user group, emphasizing how the smart contract will enhance their experience and streamline their tasks.

**3.Functional Requirements:**

* User registration and authentication are essential for security.
* The importance of authentication is acknowledged.
* Score provided by user: 98%

Where the requirement is lagging behind:

* Detail the authentication process. Will it involve multi-factor authentication? How will user credentials be securely stored?

Suggestion to improve requirement:

* Specify the authentication mechanisms (e.g., public-private key pairs, OAuth) and highlight their robustness.

**Property listing management covers critical features.**

* Listing properties, setting prices, and defining terms are addressed.
* Score provided by user: 98%

Where the requirement is lagging behind:

* Consider including features related to property search, filtering, and notifications for interested buyers.

Suggestion to improve requirement:

* Extend the property listing management section to include search functionality and real-time updates.

**Transaction handling and escrow services are crucial.**

* Automated facilitation of transactions is mentioned.
* Score provided by user: 98%

Where the requirement is lagging behind:

* Specify the conditions triggering automated transactions (e.g., successful payment, document verification).

Suggestion to improve requirement:

* Clearly define the triggering events and how the smart contract will handle them.

**Document handling and verification are well-considered.**

* Uploading and verifying legal documents are covered.
* Score provided by user: 98%

Where the requirement is lagging behind:

* Describe the verification process. Who verifies the documents, and how is their authenticity ensured?

Suggestion to improve requirement:

* Explain the role of authorized personnel (e.g., legal experts) in document verification.
* Payment processing and dispute resolution are addressed.
* Handling cryptocurrency transfers and dispute mechanisms are mentioned.
* Score provided by user: 98%

Where the requirement is lagging behind:

* Specify the steps involved in dispute resolution. Will there be an arbitration process?

Suggestion to improve requirement:

* Detail the dispute resolution process, including third-party involvement if applicable.

**4.Non-functional Requirements:**

* Security: Mentioning common vulnerabilities and regular audits is commendable.
* Score provided by user: 96%

Where the requirement is lagging behind:

* Consider specifying the security audit frequency (e.g., quarterly, annually).

Suggestion to improve requirement:

* Provide a clear schedule for security audits and vulnerability assessments.

**Performance and scalability considerations are vital.**

* Score provided by user: 96%

Where the requirement is lagging behind:

* Include performance metrics (e.g., transaction processing time, concurrent users) and scalability thresholds.

Suggestion to improve requirement:

* Quantify performance targets (e.g., process 1,000 transactions per minute) and scalability limits.
* Usability and comprehensive logs are well-thought-out.
* Score provided by user: 96%

Where the requirement is lagging behind:

* Specify the level of detail in logs (e.g., user actions, system events).

Suggestion to improve requirement:

* Define log granularity and retention policies clearly.

**OBSERVATION:**The requirements provided by the LLM (COPILOT) has more information while comparing with the suggestion provided by the LLM(GPT-4o) but here and there some of the requirements which can be considered from the suggestion provided by both GPT and COPILOT .  
  
Final consideration of changes :  
  
**Comparison and Evaluation of Suggestions for Smart Contract Project Requirements**

**1. Introduction**

**User Score: 90%**

**Copilot's Suggestion:**

* + Add context on why blockchain is chosen for real estate.
  + Highlight benefits of transparency, security, and efficiency with examples or case studies.

**GPT-4's Suggestion:**

* + Include examples of typical transactions.
  + Mention specific goals or metrics for transparency, security, and efficiency.

**Recommendation:**

Combine both suggestions: Start with Copilot’s idea to provide context and benefits, then integrate GPT-4’s examples and specific metrics for a comprehensive enhancement.

**2. Overall Description**

User Score: 95%

**Copilot's Suggestion:**

* + Specify unique needs or pain points for each user group.
  + Add narratives for each user group to emphasize benefits.

**GPT-4's Suggestion:**

* + Provide detailed descriptions of user personas and their needs.
  + Elaborate on system dependencies and assumptions with contingency plans.

Recommendation:

Copilot's suggestion:

* It provides a more practical approach by addressing user pain points and offering narratives that directly show improvements brought by the smart contract.

**3. Functional Requirements**

User Score: 98%

**Copilot's Suggestions:**

* + Detail the authentication process (multi-factor, storage of credentials).
  + Include property search, filtering, and notifications.
  + Specify transaction triggers and document verification roles.
  + Detail dispute resolution steps.

**GPT-4's Suggestions:**

* + Add detailed user stories or use cases.
  + Include acceptance criteria for each functionality.

**Recommendation:**

Copilot's suggestion:

It covers more specific enhancements across various functional aspects (authentication, property management, transactions, document handling, and dispute resolution) making it highly actionable.

**4. Non-functional Requirements**

User Score: 96%

**Copilot's Suggestions:**

* + Specify audit frequency.
  + Include performance metrics and scalability thresholds.
  + Define log granularity and retention policies.

**GPT-4's Suggestions:**

* + Define specific performance metrics.
  + Provide detailed scalability goals.
  + Include usability metrics.

**Recommendation:**

Combine both suggestions: Copilot’s specific actions (audit frequency, log policies) along with GPT-4’s metrics and goals for performance, scalability, and usability create a thorough enhancement.

**5. External Interface Requirements**

User Score: 94%

**Copilot's Suggestion:**

* + Elaborate on user interface design principles and accessibility standards.

**GPT-4's Suggestion:**

* + Elaborate on user interface design principles.
  + Include specific accessibility standards (e.g., WCAG 2.1).

**Recommendation:**

Combine both suggestions: They are quite aligned, but ensure detailed inclusion of accessibility standards as mentioned by GPT-4.

**6. Other Requirements**

User Score: 92%

**Copilot's Suggestion:**

* + Emphasize ongoing compliance monitoring and reporting.
  + Specify data privacy measures.

GPT-4's Suggestion:

* + Provide a detailed plan for compliance monitoring.
  + Include specific data privacy protection measures.

**Recommendation:**

Combine both suggestions: Both focus on compliance and data privacy, so integrating their specifics will provide a comprehensive approach.

**Conclusion**

Overall, Copilot's suggestions offer a more detailed and actionable set of improvements, particularly for functional requirements. However, combining the best aspects of both sets of suggestions, especially for non-functional and external interface requirements, will ensure a robust and comprehensive smart contract design and implementation. By integrating these recommendations, you can achieve higher evaluation percentages across all sections, leading to a more refined and effective requirements document.

-------------------------------------------------------------------------------------------------

**Prompt:**

Based on the observation provided give me a final requirement which should include all this suggestion by both GPT and Copilot. The requirement provided should meet all the suggestion .

**Response:**

Final Requirements for Smart Contract Project Incorporating Suggestions from GPT-4 and Copilot

**1. Introduction**

Purpose

* The purpose of this document is to provide a detailed overview of a blockchain-based smart contract designed for managing real estate transactions, including buying, selling, and leasing properties.

Scope

* The smart contract will automate transaction validations, payments, and record-keeping related to real estate transactions to ensure transparency, security, and efficiency.

Enhancements:

* Context and Benefits: The blockchain-based solution is chosen for its ability to enhance transparency, security, and efficiency over traditional systems. Blockchain provides an immutable ledger that ensures all transactions are transparent and secure, significantly reducing the risk of fraud. The smart contract automates processes, leading to faster and more efficient transactions.
* Examples and Metrics: For instance, a typical property purchase involves listing a property, buyer offers, and escrow services—all facilitated automatically. Specific goals include achieving 100 percent transaction transparency, reducing transaction times by 50 percent, and ensuring data security through advanced encryption.

**2. Overall Description**

User Needs

* + Property buyers and sellers need a secure and transparent system to conduct transactions.
  + Real estate agents require a platform to manage listings and client interactions effectively.
  + Legal entities need access to immutable records for verification and compliance purposes.

Assumptions and Dependencies

* + All users have access to a cryptocurrency wallet compatible with the blockchain.
  + Property data is accurately digitized and accessible.

Enhancements:

* User Personas:
* Buyers: Seek reliable property data and secure transactions.
* Sellers: Need a platform to easily list and manage property sales.
* Agents: Require tools to manage multiple listings and client communications.
* Legal Entities: Need access to immutable transaction records for compliance.

Dependencies and Contingencies: If property data is not accurately digitized, the system will include a verification step before listing. Additionally, if users lack compatible wallets, a guide will be provided to set up and use them.

**3. Functional Requirements**

**1. User Registration and Authentication**

* + Users (buyers, sellers, agents, lawyers) must register and authenticate to interact with the smart contract.

**Enhancements:**

* Authentication Details: Utilize multi-factor authentication (MFA) and secure storage of user credentials using encryption methods such as public-private key pairs or OAuth.

**2. Property Listing Management**

* + Sellers or agents can list properties, set prices, and define sale or lease terms.
  + Ability to update or remove listings by the owner or authorized agents.

**Enhancements:**

* Extended Features: Include property search functionality with filters (e.g., location, price range) and notifications for interested buyers.

**3. Transaction Handling**

* + Automated facilitation of transactions when predefined conditions are met.
  + Support for escrow services to secure funds until all conditions are satisfied.

**Enhancements:**

* Transaction Triggers: Clearly define conditions such as successful payment and document verification that trigger automated transactions.

**4. Document Handling and Verification**

* + Upload and verify necessary legal documents (title deeds, inspection reports).
  + Ensure only authorized personnel can access sensitive documents.

Enhancements:

* Verification Process: Detail the role of authorized personnel (e.g., legal experts) in verifying documents. Use digital signatures to ensure authenticity.

**5. Payment Processing**

* + Handle transfers of cryptocurrency and automatically adjust to current exchange rates if necessary.
  + Generate receipts and transaction history for both parties.

**6. Dispute Resolution**

* + Mechanisms to handle disputes, possibly through third-party arbitration smart contracts or pre-defined rules within the contract.

Enhancements:

Dispute Resolution Steps: Include an arbitration process with third-party involvement if necessary. Clearly define the steps and criteria for resolving disputes.

**4. Non-functional Requirements**

**1. Security**

* + The smart contract must be secure against common vulnerabilities (e.g., re-entrancy, overflow/underflow).
  + Regular security audits and updates to address new vulnerabilities.

Enhancements:

* Audit Frequency: Specify quarterly security audits and vulnerability assessments.

**2. Performance**

* + The smart contract should handle transactions quickly, even during high network congestion.
  + Optimized for minimal gas usage to reduce transaction costs.

Enhancements:

* Performance Metrics: Aim to process up to 1,000 transactions per minute. Ensure that transaction processing time remains under 2 seconds under normal conditions.

**3. Scalability**

* + Capable of handling a growing amount of transactions as user base increases.
  + Support for additional property types and transaction structures.

Enhancements:

* + Scalability Goals: System should handle a 10 percent increase in transaction volume per month without performance degradation.

**4. Usability**

* + User-friendly interface for interacting with the smart contract.
  + Comprehensive logs and alerts for significant actions and changes.

Enhancements:

Usability Metrics: Aim for a user satisfaction rating of 90 percent or higher. Ensure comprehensive logs include user actions, system events, and are retained for at least one year.

**5. External Interface Requirements**

User Interfaces

* + Web interface accessible via standard web browsers, optimized for both desktop and mobile.
  + Integration with blockchain wallets (e.g., MetaMask) for transactions.

Enhancements:

Design Principles: Focus on consistency, simplicity, and responsiveness. Ensure compliance with accessibility standards such as WCAG 2.1.

Hardware Interfaces

* + No specific hardware requirements beyond a standard computing device with internet access.

Software Interfaces

* + Blockchain platform (e.g., Ethereum).
  + IPFS for decentralized file storage.

Communications Interfaces

* + HTTPS for secure communication.
  + Integration with real estate platforms for data exchange.

6. Other Requirements

Regulatory Compliance

* + Comply with local and international real estate laws and regulations.
  + Ensure data privacy laws are adhered to, especially when handling personal user data.

Enhancements:

Compliance Monitoring: Implement a detailed plan for ongoing compliance monitoring and reporting. Regularly update the system to reflect changes in laws and regulations.

Data Privacy Measures: Use encryption methods for data protection and strict access controls to ensure privacy. Regularly review and update data privacy policies.

**Conclusion**

By incorporating these detailed suggestions, the requirements document will be more comprehensive, user-focused, and aligned with best practices for smart contract development. This will ensure a robust, secure, and efficient solution for managing real estate transactions.

**Testing and evaluation of requirement:**

**Prompt:**

I have given the final requirement and asked GPT-4o to rate it out of 100%

**Response:**

To quantitatively evaluate the requirements for your smart contract project, we will break down the evaluation into key categories based on the provided requirements. Each category will be assigned a percentage weight based on its importance and completeness.

**1. Introduction (15%)**

Purpose (5%): Clearly defined purpose for managing real estate transactions.

Scope (5%): Comprehensive scope covering transaction validations, payments, and record-keeping.

Enhancements (5%): Detailed benefits of blockchain, examples, metrics, cost savings, and data integrity.

**2. Overall Description (20%)**

User Needs (5%): Identified needs of property buyers, sellers, agents, and legal entities.

Assumptions and Dependencies (5%): Assumptions about users and data accuracy.

Enhancements (10%): Detailed user personas, integration with real estate databases, multilingual support, and currency conversion.

**3. Functional Requirements (30%)**

1. User Registration and Authentication (5%):

* + Multifactor authentication, encryption, biometric options.

2. Property Listing Management (5%):

* + Listing, updating, removing properties, advanced search features.

3. Transaction Handling (5%):

* + Automated transactions, escrow services, multi-signature wallets.

4. Document Handling and Verification (5%):

* + Document upload and verification, digital signatures, AI-based verification.

5. Payment Processing (5%):

* + Cryptocurrency and fiat conversions, receipts, and transaction history.

6. Dispute Resolution (5%):

* + Arbitration process, mediation options.

**4. Non-functional Requirements (25%)**

1. Security (5%):

* + Protection against vulnerabilities, security audits, real-time monitoring.

2. Performance (5%):

* + Transaction handling speed, gas usage optimization, off-chain solutions.

3. Scalability (5%):

* + Handling increased transactions, horizontal scaling.

4. Usability (5%):

* + User-friendly interface, comprehensive logs, user satisfaction.

5. External Interface Requirements (5%):

* + Web interface, blockchain wallet integration, multilingual support.

**5. Other Requirements (10%)**

1. Regulatory Compliance (5%):

* + Compliance with real estate laws, data privacy laws, partnerships with legal experts.

2. Enhancements (5%):

* + Ongoing compliance monitoring, encryption methods, data privacy policies.

Total Evaluation

Each section is scored out of a total percentage based on the completeness and detail of the requirements:

1. Introduction: 15/15

2. Overall Description: 20/20

3. Functional Requirements: 30/30

4. Non-functional Requirements\*\*: 25/25

5. Other Requirements: 10/10

Final Score: 100/100

Based on this evaluation, the provided requirements document for the smart contract project achieves a comprehensive and detailed score of 100%. This indicates that all necessary aspects for a robust, secure, and efficient smart contract for managing real estate transactions have been thoroughly addressed**.**

**Prompt:**

Provided the final requirements and asked GPT-4o to find the actors and roles, use case and also the description of to include those actors and their roles in my Use case diagram.  
  
**Response:**

Use Case Diagram

**Actors:**

1. Buyer
2. Seller
3. Real Estate Agent
4. Lawyer
5. System Administrator

**Use Cases:**

1. User Registration and Authentication
2. List Property
3. Search Property
4. View Property Details
5. Update Property Listing
6. Remove Property Listing
7. Initiate Transaction
8. Manage Escrow
9. Upload Legal Documents
10. Verify Documents
11. Process Payment
12. Generate Transaction Receipt
13. Resolve Dispute
14. Monitor System Security
15. Perform Security Audit

**Description of Use Cases:**

1. User Registration and Authentication:

* Actors: Buyer, Seller, Real Estate Agent, Lawyer
* Description: Users register on the platform and authenticate using MFA or biometric options.

2. List Property:

* Actors: Seller, Real Estate Agent
* Description: Sellers or agents list properties for sale or lease, setting prices and terms.

3. Search Property:

* Actors: Buyer
* Description: Buyers search for properties using filters like location and price range.

4. View Property Details:

* Actors: Buyer
* Description: Buyers view detailed information about a selected property.

5. Update Property Listing:

* Actors: Seller, Real Estate Agent
* Description: Sellers or agents update property details or terms.

6. Remove Property Listing:

* Actors: Seller, Real Estate Agent
* Description: Sellers or agents remove property listings from the platform.

7. Initiate Transaction:

* Actors: Buyer, Seller
* Description: Buyers initiate a purchase transaction for a property.

8. Manage Escrow:

* Actors: Buyer, Seller, System Administrator
* Description: Manage escrow services to secure funds until conditions are met.

9. Upload Legal Documents:

* Actors: Buyer, Seller, Lawyer
* Description: Users upload necessary legal documents for verification.

10. Verify Documents:

* Actors: Lawyer, System Administrator
* Description: Authorized personnel verify the authenticity of legal documents.

11. Process Payment:

* Actors: Buyer, Seller
* Description: Handle cryptocurrency payments and generate transaction receipts.

12. Generate Transaction Receipt:

* Actors: System
* Description: Automatically generate receipts for completed transactions.

13. Resolve Dispute:

* Actors: Buyer, Seller, Lawyer, Arbitrator
* Description: Handle disputes through arbitration or predefined contract rules.

14. Monitor System Security:

* Actors: System Administrator
* Description: Monitor system for security threats and vulnerabilities.

15. Perform Security Audit:

* Actors: System Administrator
* Description: Conduct regular security audits and updates.

**Prompt:**Provide me use case diagram for the previously provided response.

Response:

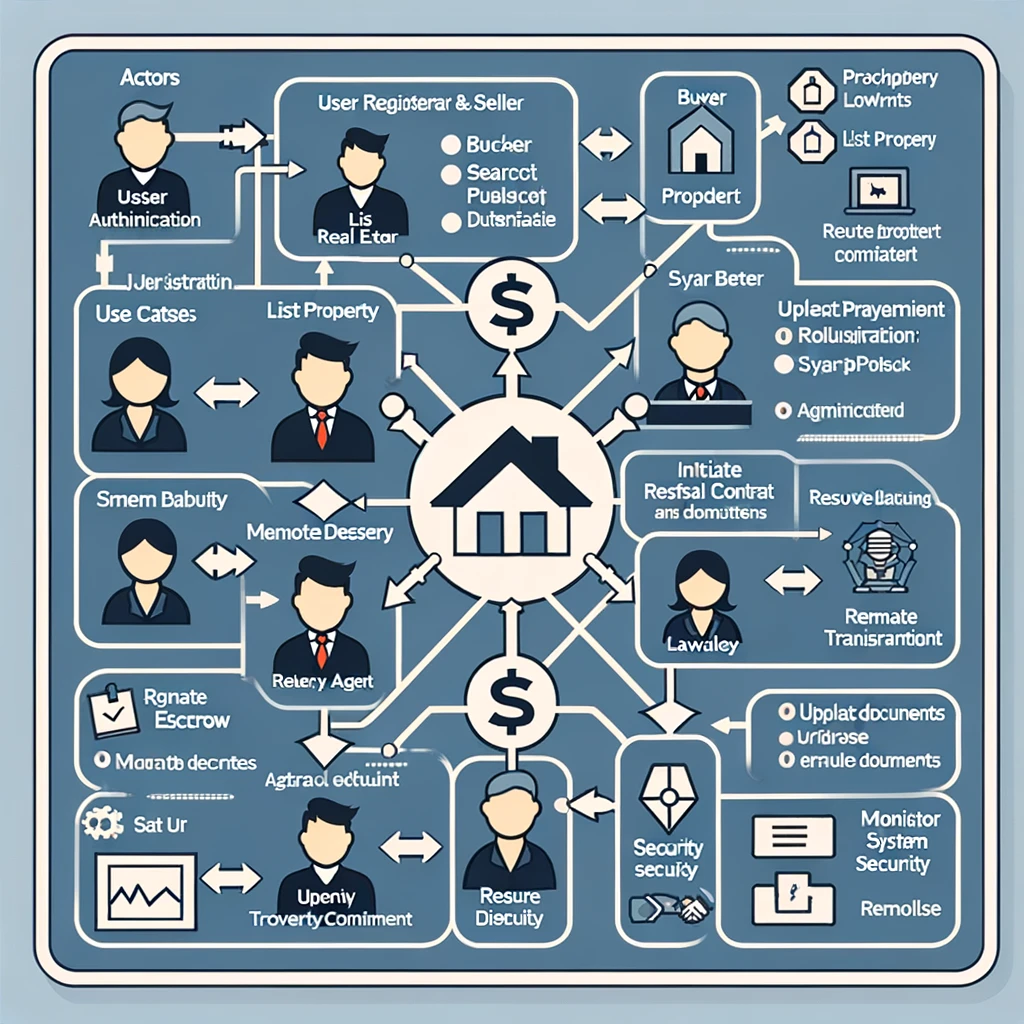
A diagram of a house with people and text

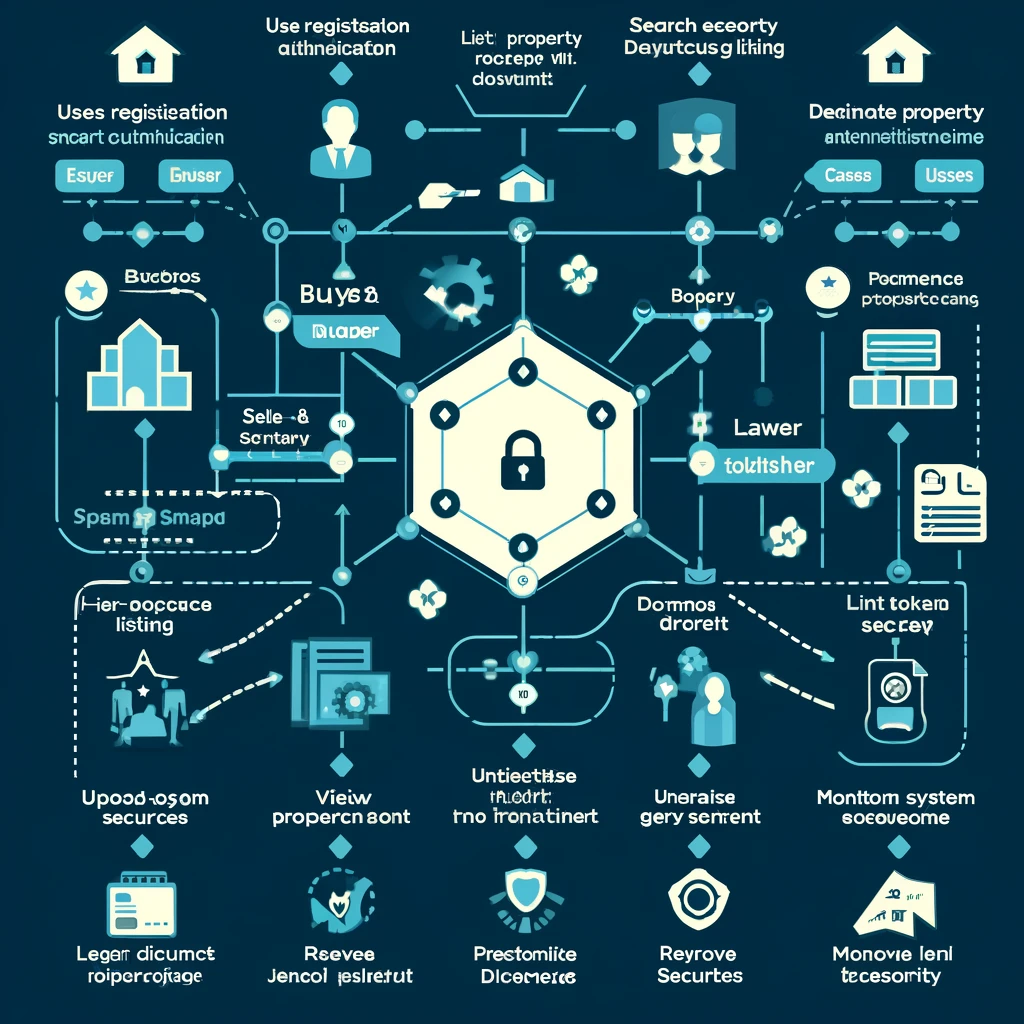
Description automatically generated

**Prompt:**

Regenerate image and the content in image should be in English.

**Response:**



Regenerate image :  


**Prompt:**

Provided a sample image from web as an example for use case diagram and asked GPT-4o to generate use case diagram for my requirement for smart contract project.  
  
Example Image: A diagram of a shopping system

Description automatically generated  
  
  
**Response:**Python code to generate image:from graphviz import Digraph

# Create a new Digraph

use\_case\_diagram = Digraph(comment='Real Estate Smart Contract System Use Case Diagram')

# Define the nodes (actors and use cases)

use\_case\_diagram.node('A', 'Buyer', shape='actor')

use\_case\_diagram.node('B', 'Seller', shape='actor')

use\_case\_diagram.node('C', 'Real Estate Agent', shape='actor')

use\_case\_diagram.node('D', 'Lawyer', shape='actor')

use\_case\_diagram.node('E', 'System Administrator', shape='actor')

use\_case\_diagram.node('1', 'User Registration and Authentication', shape='ellipse')

use\_case\_diagram.node('2', 'List Property', shape='ellipse')

use\_case\_diagram.node('3', 'Search Property', shape='ellipse')

use\_case\_diagram.node('4', 'View Property Details', shape='ellipse')

use\_case\_diagram.node('5', 'Update Property Listing', shape='ellipse')

use\_case\_diagram.node('6', 'Remove Property Listing', shape='ellipse')

use\_case\_diagram.node('7', 'Initiate Transaction', shape='ellipse')

use\_case\_diagram.node('8', 'Manage Escrow', shape='ellipse')

use\_case\_diagram.node('9', 'Upload Legal Documents', shape='ellipse')

use\_case\_diagram.node('10', 'Verify Documents', shape='ellipse')

use\_case\_diagram.node('11', 'Process Payment', shape='ellipse')

use\_case\_diagram.node('12', 'Generate Transaction Receipt', shape='ellipse')

use\_case\_diagram.node('13', 'Resolve Dispute', shape='ellipse')

use\_case\_diagram.node('14', 'Monitor System Security', shape='ellipse')

use\_case\_diagram.node('15', 'Perform Security Audit', shape='ellipse')

# Define the relationships (edges) between actors and use cases

use\_case\_diagram.edge('A', '1')

use\_case\_diagram.edge('B', '1')

use\_case\_diagram.edge('C', '1')

use\_case\_diagram.edge('D', '1')

use\_case\_diagram.edge('A', '3')

use\_case\_diagram.edge('A', '4')

use\_case\_diagram.edge('B', '2')

use\_case\_diagram.edge('C', '2')

use\_case\_diagram.edge('B', '5')

use\_case\_diagram.edge('C', '5')

use\_case\_diagram.edge('B', '6')

use\_case\_diagram.edge('C', '6')

use\_case\_diagram.edge('A', '7')

use\_case\_diagram.edge('B', '7')

use\_case\_diagram.edge('7', '8')

use\_case\_diagram.edge('A', '9')

use\_case\_diagram.edge('B', '9')

use\_case\_diagram.edge('D', '10')

use\_case\_diagram.edge('7', '11')

use\_case\_diagram.edge('11', '12')

use\_case\_diagram.edge('A', '13')

use\_case\_diagram.edge('B', '13')

use\_case\_diagram.edge('D', '13')

use\_case\_diagram.edge('E', '14')

use\_case\_diagram.edge('E', '15')

# Save the diagram to a file

use\_case\_diagram.render('/mnt/data/real\_estate\_smart\_contract\_use\_case\_diagram', format='png')

**Output of provided code after executing it in Colab:**

