**RENTAL PROPERTY SYSTEM**

**A Project Report**

##### Submitted in partial fulfillment of the requirements for the award of the Degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

**By**

**AMAN SANJAY RAI**

Seat Number: 21302C0037 **Under the esteemed guidance of**

**Ms. KALYANI RAIKAR**

**Assistant Professor, Department of Information Technology**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY**

**(Affiliated to University of Mumbai) MUMBAI, 400 037 MAHARASHTRA**

**2023 - 2024**

**VIDYALANKAR SCHOOL OF INFORMATION TECHNOLOGY**

**(Affiliated to University of Mumbai) MUMBAI-MAHARASHTRA-400037**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project entitled, **BLOCKCHAIN RENTAL PROPERTY SYSTEM SMART CONTRACTS**, is bonafied work of **Aman Sanjay Rai** bearing Seat No: 21302C0037 submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

**Internal Guide Coordinator**

**Internal Examiner External Examiner**

**Date: College Seal Principal**

ABSTRACT

As for the housing rental market, there are many landlords, arbitrary charges, false rental information and so on, and it is difficult for the government to record and supervise the housing rental situation. Based on the block chain technology of encryption algorithm, this paper constructs a housing leasing Alliance chain. The landlord and tenant sign lease agreements through smart contracts, determine the lease relationship, pay and collect rent automatically on a regular basis, and return the rental right when due. In the design of the Alliance chain network, the certification department node and the record management node are introduced to authenticate the authenticity of the house source and personal identity information, and backup the lease contract information, register the housing rental situation, and achieve the effect of safe leasing. This method eliminates the mediation, less cost, clear housing rental information, and is conducive to government market supervision.

**ACKNOWLEDGEMENT**

We would like to express our special thanks and gratitude to our project guide **Ms. Kalyani Raikar** for guiding us to do the project work on time and giving us all support and guidance, which made complete our project duly. We are extremely thankful to her for providing such nice support and guidance.

We are also thankful for and fortunate enough to get constant encouragement, support and guidance from the teachers of information Technology who helped us in successfully completing our project work.

**DECLARATION**

I hereby declare that the project entitled, **BLOCKCHAIN RENTAL PROPERTY SYSTEM SMART CONTRACTS** done at Vidyalankar School of Information Technology, has not been in any case duplicated to submit to any other universities for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fufillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum

Name and Signature of the Student

**Table of Contents**

[Chapter 1 Introduction](about:blank) 2

[1.1 Background](about:blank) 12

[1.2 Objectives](about:blank) 12

[1.3 Purpose, Scope , Applicability (Feasibility Study)……………………………….07](about:blank)

[Chapter 2 Survey of Technologies…………………………………………………………09](about:blank)

[Chapter 3 Requirements and Analysis](about:blank) 11

[3.1 User Requirements 11](about:blank)

[3.2 Technical Requirements 11](about:blank)

[3.3 Legal and Regulatory Considerations 12](about:blank)

[3.4 Bussiness Requirements 12](about:blank)

3.5 User experience and interface………………………………………….................12

3.6 Future proofing……………………………………………………………………13

3.7 Conclusion………………………………………………………………………...13

[Chapter 4 System Design 14](about:blank)

[4.1 Basic Modules 14](about:blank)

[4.2 Data Design (Table Design) 14](about:blank)

[4.2.1 Schema Design 14](about:blank)

[4.2.2 Data Integrity and Constraints](about:blank) 15

[4.3 Diagrams 16](about:blank)

[4.3.1 E-R Diagram /Block Diagram 16](about:blank)

[4.3.2 Software backend Class Diagram 17](about:blank)

[4.3.3 Use Case Diagram 18](about:blank)

[4.3.4 Data flow Diagram 19](about:blank)

[4.3.5 Architectural design 20](about:blank)

[4.3.6 User Interface design 21](about:blank)

[4.3.7 Security issues 22](about:blank)

[References 24](about:blank)

**Chapter 1: Introduction**

**1.1 Background**

With the development of social economy and changes of people’s concepts, not only the housing rental market’s demand has increased significantly but also the number of renters has increased year by year. Nevertheless, the housing rental market is experiencing frequent occurrences. For example, the proliferation of false listings, the asymmetry of information in the leasing market, the serious leakage of customer information, and the low efficiency of industry transactions have caused chaos in the housing leasing market. The abovementioned problems have severely affected the vigorous development of the housing rental market and increased the additional burden of renters. Solving the various problems in the housing rental market plays a very important role in meeting the housing needs of residents and promoting stable economic and social development.

**1.2 Objectives**

The primary objectives of this exploration into the NFT marketplace are multifaceted:

**1.2.1 Introducing Trusted Third Party**

We propose a scheme for house leasing based on blockchain technology. By introducing a trusted third party, we ensure the authenticity of rental information. Solve the problem of information asymmetry between homeowners and tenants in the housing rental market. In addition, our scheme has no housing intermediary, which saves the cost of housing intermediary.

**1.2.2 Fairness In the Transactions**

The introduction of smart contract technology has improved the fairness of house leasing transactions. The conditions for triggering the smart contract are met, then the transaction is automatically executed, which improves the fairness of industry transactions.

**1.3 Purpose, Scope, Applicability (Feasibility Study)**

**1.3.1 Purpose**

The purpose of this study is to share the listing information, preventing large companies having too many users to control the market price.

**1.3.2 Scope**

The scope of this study encompasses:

**Market Analysis**:

A detailed examination of the current state and future projections of rental property system.

**Technological Evaluation:**

An in-depth assessment of the blockchain and smart contract technologies supporting real estate.

**Chapter 2: Survey of Technologies in the NFT Marketplace**

**2.1 Blockchain Technology**

Blockchain technology has been increasingly utilized in various industries, including real estate and property management. The use of smart contracts in rental property systems based on blockchain technology can offer numerous benefits, such as increased transparency, efficiency, and security.

**2.1.1 Smart Contracts:**

* **Definition: Smart contracts are self-executing contracts with the terms directly written into code. In the context of rental property systems, smart contracts can automate and enforce the terms of a rental agreement.**
* **Benefits: Increased automation, reduced reliance on intermediaries, faster and more secure transactions, and decreased chances of fraud.**

**2.2 Blockchain Platforms:**

* **Ethereum: Ethereum is a popular blockchain platform for developing decentralized applications (DApps) and smart contracts. Its ERC-20 and ERC-721 standards are often used for creating tokens representing assets like real estate.**
* **Binance Smart Chain (BSC): Binance Smart Chain is another blockchain platform that supports smart contracts. It has gained popularity due to its lower transaction fees compared to Ethereum.**

**2.3 Tokenization of Real Estate:**

* **Definition: Tokenization involves representing ownership or rights to an asset (such as real estate) as tokens on a blockchain. Each token represents a share or fraction of the property.**
* **Benefits: Increased liquidity, fractional ownership, and easier transfer of property ownership.**

**2.4 Identity Verification:**

* + **On-chain Identity: Integrating on-chain identity verification systems ensures that the parties involved in a rental agreement are who they claim to be.**
  + **Privacy Considerations: Balancing the need for identity verification with privacy concerns is crucial in real estate transactions.**

**2.5 Interoperability and Standards:**

* **Interoperability: Ensuring that different blockchain systems and smart contracts can seamlessly interact with each other can enhance the overall efficiency of the rental property system.**
* **Standards: The industry may adopt standards for smart contracts and tokenization to ensure compatibility and ease of integration**

**2.6 Oracles:**

* **Definition: Oracles are systems or services that provide external information (e.g., real-world events, market prices) to smart contracts on the blockchain.**
* **Importance: Oracles enable smart contracts to interact with real-world data, which is crucial for applications like rental agreements that may require external information.**

**2.7 User Experience:**

* **Wallet Integration: Integrating user-friendly cryptocurrency wallets can make it easier for users to engage with the blockchain-based rental property system.**
* **UI/UX Design: Intuitive user interfaces and experiences can contribute to the adoption of blockchain solutions in real estate.**

**Chapter 3: Requirements and Analysis in the NFT Marketplace**

**3.1 User Requirements**

**3.1.1 Landlords:**

* **Ability to list properties on the blockchain.**
* **Set rental terms, including pricing, deposit requirements, and rules.**
* **Receive and track rental payments automatically via smart contracts.**
* **Access a user-friendly interface for managing property listings and transactions.**

**3.1.2 Tenants:**

* **Browse available rental properties on the blockchain.**
* **Execute rental agreements through smart contracts.**
* **Make rental payments using cryptocurrency.**
* **Access a transparent history of rental transactions.**

**3.2 Technical Requirements**

* **3.2. Automated Lease Execution:**
  + **Ensure that the smart contract can execute and enforce the terms of the lease automatically.**
  + **Include conditions for security deposits, rent payments, and penalties for breaches.**
* **Tokenization of Property:**
  + **Define how real estate assets will be represented as tokens on the blockchain.**
  + **Determine the standards (e.g., ERC-20, ERC-721) for tokenization.**
* **Payment Handling:**
  + **Design smart contracts to handle cryptocurrency payments securely.**
  + **Implement automatic rent collection and distribution.**

**3.2.2 Security and Privacy:**

* **Identity Verification:**
  + **Establish mechanisms for on-chain identity verification of both landlords and tenants.**
  + **Consider privacy implications and compliance with data protection regulations.**
* **Secure Smart Contracts:**
  + **Ensure smart contracts are secure and audited to prevent vulnerabilities.**
  + **Implement access controls to restrict unauthorized modifications.**

**3.3 Legal and Regulatory Considerations**

**3.3.1. Legal Frameworks:**

* + **Ensure the system complies with existing real estate and rental regulations.**
  + **Incorporate features to adapt to changing legal requirements.**

**3.3.2.KYC/AML Compliance:**

* + **Integrate Know Your Customer (KYC) and Anti-Money Laundering (AML) checks as needed.**

**3.4 Business Requirements**

* **3.4.1 Transaction Throughput:**
  + **Assess the expected transaction volume and choose a blockchain platform that can handle the required throughput.**
  + **Consider scalability solutions if needed.**

**3.5 User Experience and Interface**

**3.5.1 Efficient Onboarding:**

* + **Streamlined registration and onboarding processes.**
  + **Clear instructions and informative prompts.**

**3.5.2 Intuitive Dashboard:**

* + **Clean, organized layout.**
  + **Prioritized information with visual elements.**

**3.5.3 Seamless Property Listing:**

* + **Simple interface for landlords to list properties.**
  + **Real-time previews and drag-and-drop features for property details.**

**3.5.4 User-Friendly Smart Contract Execution:**

* + **Clear and concise execution interface.**
  + **Tooltips and step-by-step guidance.**

**3.5.5 Smooth Rent Payment:**

* + **Simplified cryptocurrency payment process.**
  + **Transparent payment history display.**

**3.5.6 Tokenization and Ownership:**

* + **Clearly represented tokenized properties.**
  + **User-friendly transfer and ownership views.**

**3.5.7 Efficient Identity Verification:**

* + **User-friendly document submission interface.**
  + **Clear communication of verification status.**

**3.5.8 Effective Notifications:**

* + **Notification center for updates.**
  + **Customizable preferences and push notifications.**

**3.5.9 Accessible Help and Support:**

* + **Comprehensive FAQ section and visible support button.**
  + **In-app chat support and chatbots for quick assistance.**

**3.5.10 Accessibility Focus:**

* + **Design with accessibility standards.**
  + **Navigable platform for users with disabilities.**

**3.5.11 Documentation and Tutorials:**

* + **Easily accessible user manuals and guides.**
  + **Video tutorials and tooltips for complex processes.**

**3.5.12 Security and Trust:**

* + **Display of security indicators.**
  + **Two-factor authentication and ongoing security updates.**

**3.6 Future-Proofing**

**3.6.1 Ensure modular and flexible design for easy integration of emerging technologies.**

**Prioritize interoperability to adapt to evolving blockchain standards.**

**Implement upgradable smart contracts to accommodate future updates seamlessly.**

**Stay compliant with changing regulations and engage with the community for continuous improvement.**

**3.7 Conclusion**

In conclusion, a blockchain rental property system with smart contracts holds significant promise for revolutionizing the real estate industry. By leveraging decentralized technology, it enhances transparency, security, and efficiency in property transactions. However, successful implementation requires careful consideration of user experience, regulatory compliance, and future-proofing strategies. Continued collaboration with stakeholders, adherence to evolving standards, and a commitment to security and innovation will be instrumental in realizing the full potential of blockchain smart contracts in the rental property ecosystem.

**Chapter 4: System Design**

**4.1 Basic Modules**

1. **Tokenization Module:**
   * **Converts real estate assets into blockchain tokens for fractional ownership.**
2. **Lease Execution Module:**
   * **Automates and enforces lease agreements through smart contracts.**
3. **Payment Module:**
   * **Handles cryptocurrency-based rent payments and secure distribution.**
4. **Identity Verification Module:**
   * **Ensures on-chain identity verification for landlords and tenants.**
5. **Notification Module:**
   * **Sends alerts for important events, such as upcoming payments or lease changes.**
6. **Dashboard Module:**
   * **Provides a user-friendly interface for managing properties and leases.**
7. **Oracle Integration Module:**
   * **Fetches real-world data to inform smart contract decisions.**
8. **Regulatory Compliance Module:**
   * **Ensures adherence to legal and regulatory requirements.**
9. **Security Module:**
   * **Safeguards the system with encryption, access controls, and audits.**
10. **Documentation and Training Module:**
    * **Offers user manuals, guides, and educational materials.**

**4.2 Data Design (Table Design)**

**4.2.1 Schema Design**

1. **Property Table:**
   * **Attributes:**
     + **PropertyID (Primary Key)**
     + **Address**
     + **Size**
     + **Amenities**
     + **Images**
     + **OwnerID (Foreign Key referencing Identity Table)**
2. **Token Table:**
   * **Attributes:**
     + **TokenID (Primary Key)**
     + **PropertyID (Foreign Key referencing Property Table)**
     + **OwnershipDistribution**
3. **Lease Table:**
   * **Attributes:**
     + **LeaseID (Primary Key)**
     + **PropertyID (Foreign Key referencing Property Table)**
     + **TenantID (Foreign Key referencing Identity Table)**
     + **LandlordID (Foreign Key referencing Identity Table)**
     + **StartDate**
     + **EndDate**
     + **Terms**
     + **Status**
4. **Payment Table:**
   * **Attributes:**
     + **PaymentID (Primary Key)**
     + **LeaseID (Foreign Key referencing Lease Table)**
     + **Amount**
     + **Date**
     + **PayerID (Foreign Key referencing Identity Table)**
     + **PayeeID (Foreign Key referencing Identity Table)**
     + **Status**
5. **Identity Table:**
   * **Attributes:**
     + **IdentityID (Primary Key)**
     + **Name**
     + **ContactDetails**
     + **KYCStatus**
     + **AMLStatus**
6. **Notification Table:**
   * **Attributes:**
     + **NotificationID (Primary Key)**
     + **RecipientID (Foreign Key referencing Identity Table)**
     + **Message**
     + **Timestamp**
     + **Status**
7. **Dashboard Analytics Table:**
   * **Attributes:**
     + **AnalyticsID (Primary Key)**
     + **UserID (Foreign Key referencing Identity Table)**
     + **PropertyCount**
     + **LeaseCount**
     + **PaymentHistory**
     + **OtherMetrics**
8. **Oracle Data Table:**
   * **Attributes:**
     + **OracleDataID (Primary Key)**
     + **DataType**
     + **Source**
     + **Timestamp**
     + **RelevantInformation**
9. **Regulatory Compliance Table:**
   * **Attributes:**
     + **ComplianceID (Primary Key)**
     + **LeaseID (Foreign Key referencing Lease Table)**
     + **Status**
     + **RegulatoryChanges**
     + **AdaptationMeasures**
10. **Security Table:**
    * **Attributes:**
      + **SecurityID (Primary Key)**
      + **AccessLog**
      + **AuditTrail**
      + **EncryptionKeys**
      + **SecurityConfigurations**
11. **Documentation and Training Table:**
    * **Attributes:**
      + **DocID (Primary Key)**
      + **Title**
      + **Content**
      + **ResourceType**
      + **UserFeedback**

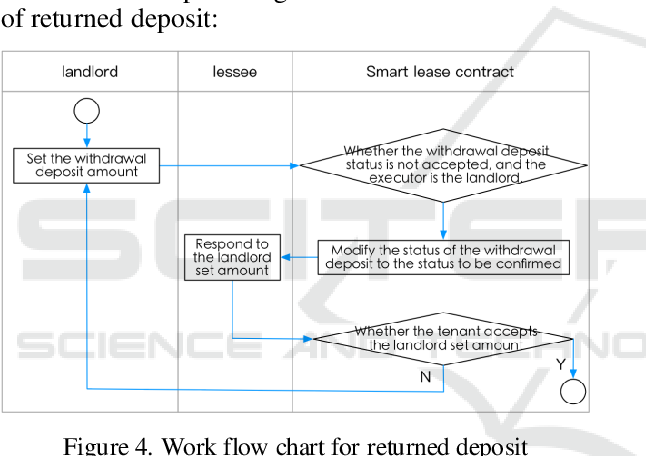
**4.2.2 Data Integrity and Constraints**

Ensuring data integrity and implementing constraints is crucial for the effective and secure functioning of any web application, including crowdfunding platforms. Here are some key considerations for data integrity and constraints in the context of a crowdfunding web application:

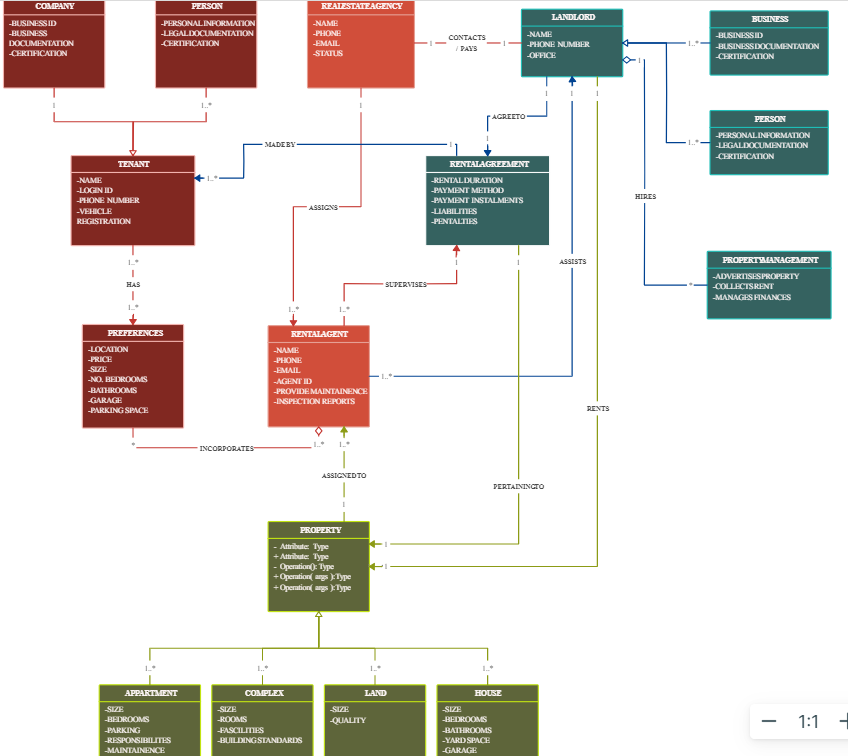
1. **Primary Key Constraints:** Ensure unique identification in each table for data integrity.
2. **Foreign Key Constraints:** Establish relationships between tables to maintain referential integrity.
3. **Unique Constraints:** Restrict fields to unique values, preventing duplicates.
4. **Check Constraints:** Specify conditions for allowable data values, validating accuracy.
5. **Default Constraints:** Set default values for fields to ensure consistency.
6. **Transaction Constraints (ACID):** Implement ACID properties for secure and reliable transactions.
7. **Immutable Smart Contracts:** Design contracts to be unchangeable once deployed for historical preservation.
8. **Encryption Mechanisms:** Apply encryption to sensitive data, ensuring confidentiality.
9. **Consensus Mechanisms:** Use consensus algorithms to validate transactions and ensure agreement.
10. **Event Logging and Auditing:** Record system events for traceability, accountability, and data integrity.

**4.3 Diagrams**

**4.3.1.E-R diagram/Block diagram.**



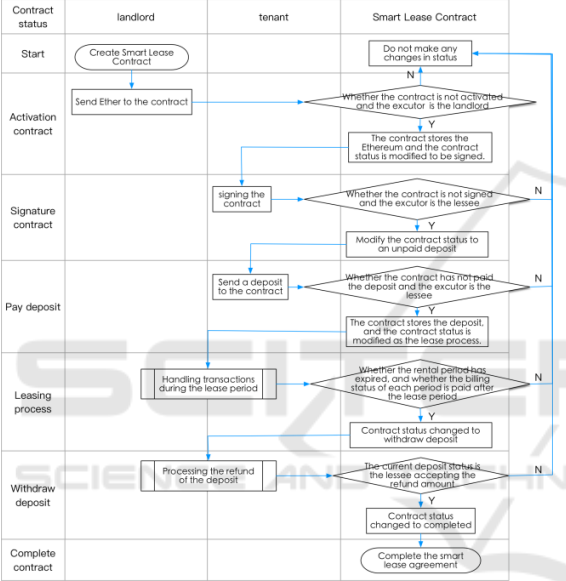
**4.3.2. Software backend class diagram**

****

**4.3.3. USE CASE Diagram**

****

**4.3.4. Data flow diagram**

****

**4.3.5. ARCHITECTURAL DESIGN**

**OVERVIEW**

1. **Blockchain Layer:**
   * **Choose Ethereum for smart contract execution.**
2. **Tokenization Module:**
   * **Use ERC-20/ERC-721 to tokenize real estate assets.**
3. **Lease Management Module:**
   * **Develop smart contracts for automated lease execution.**
4. **Payment Processing Module:**
   * **Implement transparent cryptocurrency-based rent payments.**
5. **Identity Verification Module:**
   * **Integrate KYC/AML checks for secure on-chain identity verification.**
6. **Data Storage Layer:**
   * **Utilize a decentralized database with encryption for secure data storage.**
7. **Security Layer:**
   * **Conduct regular smart contract audits and enforce access controls.**
8. **User Interface (UI) Layer:**
   * **Develop an intuitive web/mobile interface for landlords and tenants.**
9. **Notification Layer:**
   * **Implement a module for system-generated notifications and alerts.**
10. **Documentation and Help Layer:**
    * **Provide user manuals, guides, and support mechanisms.**
11. **External Integrations:**
    * **Fetch real-world data through oracles.**
    * **Integrate compliance mechanisms for regulatory changes.**
12. **Scalability and Future-Proofing:**
    * **Design for scalability and modular updates to accommodate growth**

**4.3.7 SECURITY ISSUES**

1. **Smart Contract Vulnerabilities:**
   * **Risk:** Coding errors or vulnerabilities in smart contracts can lead to exploits or attacks.
   * **Mitigation:** Regular code audits, testing, and adherence to best practices can help minimize vulnerabilities.
2. **Oracle Exploitation:**
   * **Risk:** Reliance on external data sources (oracles) may expose the system to manipulation or inaccuracies.
   * **Mitigation:** Carefully vet oracle providers, implement data verification mechanisms, and consider multiple oracles for redundancy.
3. **Identity Management Risks:**
   * **Risk:** Compromised user identity or KYC data can lead to fraudulent transactions or unauthorized access.
   * **Mitigation:** Implement strong encryption for identity data, conduct regular security audits, and ensure secure KYC/AML processes.
4. **Transaction Privacy Issues:**
   * **Risk:** Public blockchain transactions can potentially expose sensitive information.
   * **Mitigation:** Use privacy-focused techniques or off-chain solutions to protect sensitive transaction details.
5. **Scalability Concerns:**
   * **Risk:** Scalability issues can lead to delays, congestion, or higher transaction costs.
   * **Mitigation:** Choose a blockchain platform with sufficient scalability features and regularly assess network congestion.
6. **Regulatory Compliance:**
   * **Risk:** Evolving regulations may pose challenges in compliance for blockchain-based systems.
   * **Mitigation:** Stay informed about legal developments, design the system with adaptability in mind, and seek legal advice.
7. **Centralization Risks:**
   * **Risk:** Overreliance on centralized components may undermine the decentralized benefits of blockchain.
   * **Mitigation:** Distribute critical components and functionalities across the network to enhance decentralization.
8. **Double-Spending Attacks:**
   * **Risk:** Attempted manipulation of transactions to spend the same cryptocurrency multiple times.
   * **Mitigation:** Implement consensus mechanisms and transaction verification processes to prevent double-spending.
9. **Social Engineering Attacks:**
   * **Risk:** Phishing or other social engineering attacks can target users and compromise their accounts.
   * **Mitigation:** Educate users about security best practices, implement multi-factor authentication, and use secure communication channels.
10. **Governance and Upgrade Risks:**
    * **Risk:** Issues related to decentralized governance or challenges in upgrading smart contracts.
    * **Mitigation:** Establish a clear governance model, consider upgradeability in smart contract design, and involve the community in decision-making.

**REFERENCES**

Yu, Y., Dong, Y. & Guo, X., 2018. Pricing for sales and per-use rental services with vertical differentiation. European Journal of Operational Research, 270(2): 586–598. Zhuang, W., Chen, J. & Fu, X., 2017. Joint dynamic pricing and capacity control for hotels and rentals with advanced demand information. Oper. Res. Lett. Yu, X.L., Xu, X. & Bin Liu, 2017. EthDrive-A Peer-to-Peer Data Storage with Provenance. CAiSE-Forum-DC. Wilkinson, S., 2016. Storj A Peer-to-Peer Cloud Storage Network. Available at: https://storj.io/storj.pdf. Cheng, Z. et al., 2018. Deterministic Proof of Work. arXiv.org, cs.CR. Gupta, D., Saia, J. & Young, M., 2017. Proof of Work Without All the Work: Computationally Efficient Attack-Resistant Systems. arXiv.org, cs.DC. Y. Yuan, F.-Y. Wang, 2016. Blockchain: The State of the Art and Future Trends[J]. Acta Automatica Sinica, 42(4): 481-494. Kim, H.M., Laskowski, M. & Nan, N., 2018. A First Step in the Co-Evolution of Blockchain and OntologiesTowards Engineering an Ontology of Governance at the Blockchain Protocol Level. CoRR, cs.CY. Nakamoto S, 2009. Bitcoin: a peer-to-peer electronic cash system. Available at: https://bitcoin.org/bitcoin.pdf. Ethereum White Paper, 2015. A next-generation smart contract and decentralized application platform. Available at: https://github.com/ethereum/wiki/wiki/ White-Paper. Wood, D., 2014. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Available at: https://gavwood.com/paper.pdf. David Johnston, 2015. The General Theory of Decentralized Applications, Dapps. Available at: https://github.com/DavidJohnstonCEO/DecentralizedA pplications. Kim, H., M., 2018. A perspective on blockchain smart contracts: reducing uncertainty and complexity in value exchange. Social Science Electronic Publishing, 1-6. Cong, W., Z., 2018. Blockchain Disruption and Smart Contracts. Social Science Electronic Publishing, 7-10. Adler, J. et al., 2018. Astraea: A Decentralized Block-chain Oracle. arXiv.org, cs.CR. de Pedro, A.S., Levi, D. & Cuende, L.I., 2017. Witnet: A Decentralized Oracle Network Protocol. arXiv.org, cs.CR.