Project Title

FreelanceShield: Fortifying Freelancers, Safeguarding Clients

Team Members

- Aayusha Odari
- Ujjwal Dhakal
- Ankit Rayamajhi
- Rajesh Bhusal

Project Description

FreelanceShield is designed to streamline the payment process for freelance projects using blockchain technology. The platform operates on a simple yet secure system where clients deposit funds into an escrow smart contract upon project commencement. These funds are then automatically released to freelancers upon project completion and approval.

The user journey begins with a freelancer creating a project contract and defining the payment terms. Clients then review the project terms, register on the system, and upon agreement, deposit the project fee into the smart contract. The platform ensures secure and timely payments without the need for intermediaries, enhancing trust and efficiency in the freelance economy.

Upon project completion, the freelancer submits their work through the platform. The client is notified and allowed to review the submitted work. If satisfied, the client approves the completion within the platform, triggering the smart contract to release the funds to the freelancer's wallet.

If there are disputes, the platform provides a resolution framework where the predefined conditions within the smart contract determine the outcome, ensuring fairness and adherence to the initial agreement.

Importance of Blockchain Technology in FreelanceShield

Blockchain technology makes FreelanceShield safer and more reliable than older online systems. It removes the need for middlemen by using smart contracts that automatically handle payments once work is approved. This means fewer chances for mistakes and no waiting for third parties to process transactions. Everyone can see the transaction details, which helps prevent misunderstandings and builds

trust between freelancers and clients. Plus, it cuts down on costs and saves time, making the whole process of getting paid for freelance work smoother and quicker.

User Roles

1. **Freelancer:**
   Sets up project contracts, defines payment terms, submits completed work, and receives payment upon client approval.
2. **Client:**
   Registers and reviews project contracts, deposits payment into the smart contract, approves completed work, and initiates dispute resolution if necessary.

System Development

In the development of FreelanceShield using Agile methodology, we initiated the project by assembling a proficient team and thoroughly analyzing requirements. System design involved creating architectural blueprints, focusing on blockchain integration for payment solutions. Using Figma, we prototyped the system, emphasizing payment automation and smart contract functionalities. Development included coding with Solidity for smart contracts and Ethereum for blockchain integration. Rigorous testing ensures reliability and security. Tools included Ethereum for blockchain, Solidity for smart contracts, specialized testing tools, and Figma for prototype design.

Prototype Design

Explore our UI design mockups on Figma here.

Methodology

In our project development, we conducted thorough research, delving into existing platforms and industry challenges through literature reviews, market analysis, and competitor assessments. FreelanceShield stands out by utilizing blockchain for secure payments and prioritizing user experience. We studied leading freelance platforms and emerging blockchain solutions to shape FreelanceShield's unique approach. Our prototype development journey began with wireframing, advancing to mockups and a clickable prototype. The system design of FreelanceShield emphasizes blockchain integration for secure payments, employing smart contracts and a decentralized structure for data integrity.

Project Block Diagram

1. Entities

a. Client:

A principal user responsible for commissioning and assigning project-based tasks to a freelancer within a professional engagement framework.

b. Freelancer:

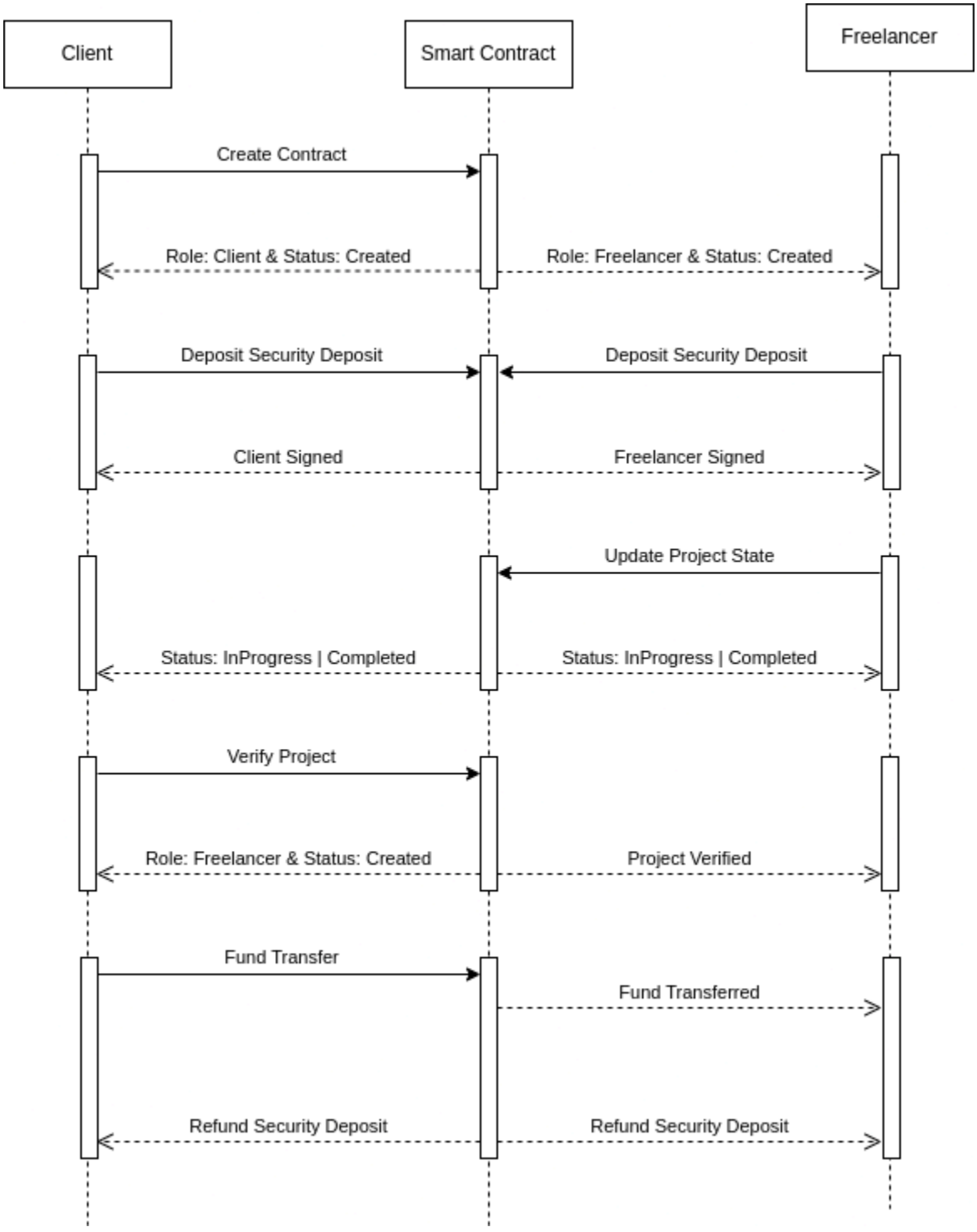
An operative user engaged in executing project tasks for diverse clients within the freelancing paradigm.

2. Data Stores

- Stores user details, profiles, and role-specific information.
- Records project terms and payment specifics.

-

3. Sequence Diagram



Problems Encountered

1. **Smart Contract Logic Complexity:** Developing and testing intricate smart contract conditions for fund release posed challenges.
2. **Transaction Scalability:** Addressing blockchain limitations for handling peak transaction volumes required scalable solutions.
3. **Security in Blockchain:** Collaborating with experts was crucial to fortify the platform against blockchain vulnerabilities.
4. **User Experience and Adoption:** Simplifying the interface for newcomers while maintaining complexity demanded user-centric design and educational efforts.
5. **Regulatory Compliance:** Continuous legal counsel engagement ensured compliance with evolving regulations across diverse jurisdictions.
6. **Balance of Transparency and Confidentiality:** Safeguarding project details while maintaining blockchain transparency involves encryption and access protocols.

Future Enhancements

1. **Smart Contract Templates:** Pre-designed contract structures for various freelance categories.
2. **Multi-Currency Support:** Enable transactions in different global currencies.
3. **Reputation System:** Allow users to rate and provide feedback for transparency.
4. **Customizable Smart Contracts:** Tailor contracts to specific project requirements.
5. **Automated Invoicing:** Generate invoices automatically at project milestones.
6. **Integration with Management Tools:** Connect with platforms for streamlined workflows.
7. **Blockchain Scaling Solutions:** Explore options for improved scalability and speed.
8. **Advanced Analytics:** Provide detailed project and financial insights.
9. **AI-Driven Contracts:** Develop contracts adaptable to project dynamics using AI.

GitHub Repository

FreelanceShield's GitHub Repository