FreelanceX Technical Documentation – EasyA Hackathon

Jonathan McKesey

Julian Cruzet

Paulo Chan

The Hai Nguyen

Ruel Gutierrez

📖 I. Narrative

Project name: FreelanceX

What are we building: FreelanceX is a decentralized escrow platform built on **Stellar Soroban smart contracts.** It automates freelance payments based on verifiable work milestones, eliminating middlemen and reducing disputes.

What does it fix: Freelancers and clients in developing economies often face delayed/failed payments, high fees from intermediaries and a lack of trust between unknown parties

Value proposition: Our solution utilizes Smart contracts and Stellar passkeys for simple job request creation, job acceptance and payment. Furthermore, by using DAOs for disputes ensures that neither the freelancer nor the client can exploit the decentralized and anonymous nature of our product.

🔨 II. MVP

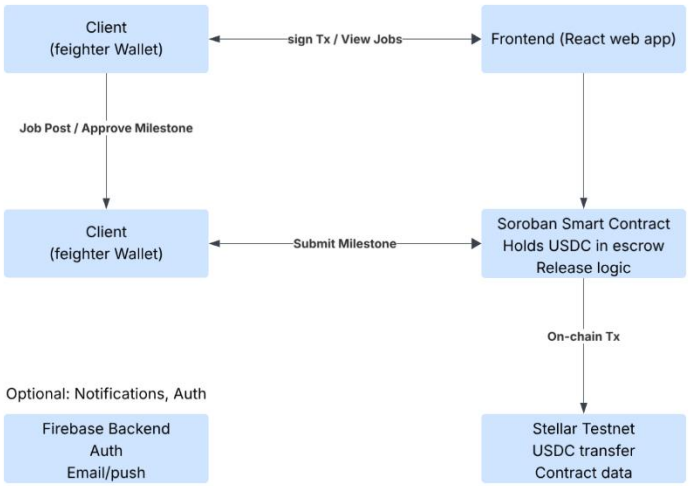
🔗 https://github.com/JulianCruzet/FreelanceX-Stellar-Escrow-dApp

💡 MVP Functionality Summary (as of 2025-05-16):

- Fully interactive front-end
- Stellar passkey integration
- Working test net transactions
- Barebones smart contract functionality

📄 III. Technical documentation

🔧 Architecture overview



🔍 In-depth component breakdown

Component	Description	Tools/Libs Used
Frontend UI	Wallet connection, UI interaction	React, Tailwind, Passkeys Kit
Smart Contracts	Core logic (staking, token, escrow, etc.)	Soroban, Rust
Wallet/Auth	Key management, sign-in flow	Stellar Passkeys Kit
Deployment	Used for contract launch and app visibility	LaunchTube

⚙️ IV. Key Design Decisions

Why Soroban on Stellar?

- Stellar is optimized for low fees and high throughput, making it ideal for microtransactions like milestone payments
- Soroban brings Turing-complete smart contracts to Stellar with wasm-based execution on rust safety

Why PasskeyKit?

- Integrates modern passwordless login using device biometrics or secure credentials.
- Enhances security while removing barriers to entry, especially in low-trust, global freelance markets

Escrow Milestone Model:

- We chose milestone-based release because it's the simplest trustless structure to reduce payment disputes without needing oracles or court-like arbitration
- A DAO dispute system (planned in v2) gives the model neutrality without off-chain bias.

⚠️ V. Challenges and Tradeoffs

Challenge	Resolution/Tradeoff
Soroban contract documentation is still maturing	Used examples and tested heavily on test net
Passkeys integration complexity	Used Stellar's SDK
Dispute logic can't be enforced trustlessly (yet)	Will add DAO-based arbitration in v2

🧪 VI. Testnet Details

Component	Status	Notes
Soroban Contract	✅ Deployed	
Frontend	❎ Local	
Test Token Used	✅ USDC (Testnet)	

📦 VII. Deployment Instructions

```
# Clone the repo
git clone https://github.com/JulianCruzet/FreelanceX-Stellar-Escrow-dApp.git
cd FreelanceX-Stellar-Escrow-dApp

# Install dependencies
npm install

# Run the frontend
npm run dev

# Deploy contract to Soroban testnet
soroban contract deploy --wasm ./contracts/escrow_contract.wasm --network testnet
```

🧩 VIII. Future plans

- Input sanitization
- DAO arbitration system in v2
- Reputation and KYC Tokens
- Model PWA Optimization
- Cross-chain Freelance Credentials (e.g., Polkadot, Bahamut)
- Job Completion NFT as verifiable proof on-chain