

User Story

Decentralized Milestone Payment Protocol

Project Name:

PIVOX – Payment Infrastructure for Verified Operations and eXecution

Value Proposition:

PIVOX provides a decentralized, transparent, and programmable infrastructure for managing freelance contracts and milestone-based payments. PIVOX eliminates disagreements, delays, and the need for centralised escrow services by ensuring that freelancers are paid only after milestones are approved by both parties through smart contracts and token vaults on Solana. Without the need for middlemen, the platform streamlines the flow of verifiable, trustless, and timely payments.

Product-Market Fit:

Freelance and remote work is now global — yet most freelancer–client engagements rely on trust, informal communication, or high-fee platforms. Unresolved problems include ambiguous milestones, late payments, and escrow mismanagement. PIVOX directly tackles this by providing a distributed option where programmable approvals, completely auditable contracts, and automated, milestone-triggered fund releases let both freelancers and customers keep control.

Target User Profiles:

- The "Independent Deliverer" Freelancer

Values financial security, clear scope, and prompt payment. Likes dealing with verifiable contracts under which money is locked upfront and released only following milestone approval.

- The "Accountable Builder" Client

Wants to pre-fund work for credibility but keep control over when funds are released. Values spending control, milestone verification, and systematic contracts.

- The "Coordinator" Project Manager

Oversees deliverables across several contractors and seeks a clear, trustless method to authorise and release milestone-based payments.

User Story ID: PX-001

1. User Persona

Name: Ananya

Role: Freelancer

Goal: Securely deliver design work under a pre-funded contract and get paid once milestones are approved.

2. User Story

As a freelancer, I want to work under contracts where payment is locked upfront and released only after both parties approve a milestone, so that I don't have to chase clients for payment.

3. Acceptance Criteria

Functionality:

- Freelancers can view contracts and milestones linked to them.
- Freelancers can approve completed milestones via wallet.
- Payments are automatically released once both parties sign.

User Interaction:

- Freelancers connect wallets to view project state, approval status, and contract terms.
- They receive tokens in their wallet once a milestone is approved.

Security:

- Milestone approval and contract state are stored immutably on-chain.
- Vaults are PDA-controlled and funds cannot be accessed without contract-based triggers.
- Only the assigned freelancer and client can interact with their contract state.

4. Priority:

High

5. Technical Notes (for Developers)

Dependencies:

- Anchor smart contract with SPL Token integration
- PDA-based vault and program signer
- Phantom wallet integration

Considerations:

- Use `findProgramAddress` to lock vault under program control
- Prevent replay or duplicate payout with a `released` flag
- Optional off-chain file storage (Arweave) for milestone proof

User Story ID: PX-002

1. User Persona

Name: Rajeev

Role: Client

Goal: Safely commit funds for a freelance project and approve milestone payouts only after work is verified.

2. User Story

As a client, I want to fund a project vault in advance and approve milestone payments when deliverables are met, so that I can manage payouts responsibly and prevent premature releases.

3. Acceptance Criteria

Functionality:

- Clients can initiate a contract, set milestone value, and deposit tokens.
- Clients can approve milestone completions.
- Payments release only after dual approval.

User Interaction:

- Clients view contract dashboard and click to approve milestones.
- Status updates occur immediately upon transaction confirmation.

Security:

- Client tokens are locked in a secure vault PDA.
- Approval checks confirm both parties signed before release.
- No third party or contract override is allowed.

4. Priority:

High

5. Technical Notes (for Developers)

Dependencies:

- SPL Token vault account derived via PDA
- USDT/USDC mint compatibility

Considerations:

- Implement seed validation for each signer
- Ensure precise match of vault authority to contract authority
- Future roadmap: allow milestone batch support per contract

User Story ID: PX-003

1. User Persona

Name: Sara

Role: Project Manager

Goal: Oversee multiple contributors and approve milestone payments across contracts transparently.

2. User Story

As a project manager, I want to manage multiple freelancer contracts from a single dashboard and track milestone approvals in real time, so that I can coordinate work and payouts effectively.

3. Acceptance Criteria

Functionality:

- Multi-wallet dashboard view of contracts and their milestone states.
- Can view who has approved and trigger reminders if pending.

User Interaction:

- Logs in with wallet and views a summary table.
- Milestone statuses show "Pending," "Approved," or "Released."

Security:

- Project managers can only view data, not approve/release funds unless assigned to the contract.

4. Priority:

Low

5. Technical Notes (for Developers)

Dependencies:

- Read-only contract data hooks
- Frontend indexer for dashboard visibility

Considerations:

- Dashboard UI should poll on-chain state or use events

- Visual cues for approval thresholds (e.g. dual-signed = green)

User Story ID: PX-004

1. User Persona

Name: Amit

Role: Freelancer

Goal: Earn rewards for early adoption and successful milestones.

2. User Story

As a freelancer, I want to earn **\$PIVOX points** for every successfully completed contract, so I feel motivated to use PIVOX more and build a track record of trust.

3. Acceptance Criteria

Functionality:

- Freelancers earn \$PIVOX points for successfully completing contracts with verified milestones.
- Points accumulate and can be redeemed for **gas fee discounts** or **priority onboarding** for future contracts.

User Interaction:

- Freelancers can track their accumulated \$PIVOX points from their dashboard.
- Points are automatically awarded when a contract milestone is marked as completed and both parties approve it.

Security:

- Points are non-transferable and are tied to the freelancer's wallet address.
- Points are stored on-chain and can't be altered.

4. Priority:

Very Low (Future Implementation)

5. Technical Notes (for Developers)

Dependencies:

- Creation and tracking of \$PIVOX points (via SPL tokens or a custom token).
- Smart contract integration to automatically issue points after milestone completion.
- Display functionality in the freelancer's dashboard.

User Story ID: PX-005

1. User Persona

Name: Pranav

Role: Client

Goal: Receive incentives for using PIVOX over traditional escrow services.

2. User Story

As a client, I want to receive **discounts or perks** for using PIVOX to manage milestone payments, so I am motivated to adopt this platform over traditional escrow or payment services.

3. Acceptance Criteria

Functionality:

- Clients who use PIVOX for their first **5 contracts** receive a **gas fee rebate** or **reduced service fees**.
- Clients also earn **PIVOX reputation points** for each successful contract, which improves their **reputation score**.

User Interaction:

- Clients will be notified when they become eligible for perks.
- Perks are displayed on the dashboard, such as "5 contracts left to earn a rebate."

Security:

- Gas fee rebates or discounts are automatically calculated when transactions are initiated.
- Reputation scores are updated after each successful contract completion.

4. Priority:

Very Low (Future Implementation)

5. Technical Notes (for Developers)

Dependencies:

- Smart contract logic to track first-time clients and apply rebates or discounts.
- Reputation tracking based on the client's contract history.
- Wallet integration to provide fee reimbursements.