SDG BLOCKCHAIN ACCELERATOR

Debugging and Testing Report – Template

1. Project Information

- **Project Name: Blackfrog**
- **Challenge & UNDP Office:** Rising ethnic tensions in the Balkans and Central Asia are fueled by misinformation and untrusted systems, alongside opaque mineral supply chains that hinder economic independence. **UNDP Istanbul Regional Hub**
- **Report Version:** v1.0

2. Testing Approach

Unit tests implementation are ongoing in Aiken to verify contract logic. The following tests are included :

- **Unit Testing:**
- **Contribution Logic**
  - test user_contribute_success()
  - test user_contribute_over_pay()
  - test user_contribute_under_pay()
  - test user_contribute_exceeds_target()
- **Admin Claim Logic**
  - test admin_claim_success()
  - test admin_claim_not_reached_target()
  - test admin_claim_not_admin()
- **Minting Logic**
  - test mint_success()
  - test mint_wrong_stt_quantity()
  - test mint_wrong_cf_quantity()
  - test edge_case_zero_cf_tokens()
  - test mint_config_utxo_not_consumed()

- **Integration Testing**:
  Deployed and tested on Cardano Preprod Testnet, simulating user interactions and contract flows under real network conditions.

- **Edge Cases**
- Contributions exceeding target
- Zero CF token edge case
- Minting with missing config UTxO

3. Error Logs

- No critical runtime errors encountered during integration testing.
- Aiken compiler and simulation logs were used for debugging during unit testing.
- Minor inconsistencies in CF token decimals logged for further investigation.

4. Discovered Issues

Issue ID	Description	Root Cause	Resolution	Status
001	Incorrect CF token decimal handling.	Unclear standard for token decimal precision in validator logic	Adjusted validator to follow Cardano-native token decimals convention	✅ Fixed
002	Contribution deadline not enforced	Initial validator lacked checks for contribution period	Added time-based validation (slot interval check) in validator logic	✅ Fixed
003	Owner withdraw logic missing	Missing branch in validator	Implemented ownerWithdraw redeemer in validator	✅ Fixed
004	No unit test coverage for validator	Validation only tested via off-chain integration	Plan to add Aiken unit tests for all redeemer branches	⏳ Pending

5. Optimization Notes

- Script efficiency: Refactored validator branches to reduce redundant checks and minimize execution units.

- Deadline checks: Added time-based validation at the contract level to reject late contributions without off-chain logic.
- Contribution handling: Adjusted validation logic to enforce minimum contribution rules directly on-chain.
- Future optimization: Plan to benchmark script memory usage and execution budget under high-volume transactions (stress testing).

6. Tools and Environments Used

- **Smart Contract Development: Aiken v1.1.17+c3a7fba**
  - Commands: aiken check, aiken build
- **Off-Chain Infrastructure: Node.js (with Lucid Evolution)**
  - Used for wallet management, UTxO tracking, transaction building, and submission.
- **Monitoring**: Blockfrost API (Preprod endpoint)
- **Network**: Cardano Preprod Testnet
- **Repository**: rwa-cardano-launchpad (https://github.com/Elizaproai/rwa-cardano-launchpad)

7. Remaining Issues / Next Steps

Pending improvements:

- Add unit tests in Aiken for all redeemer branches (contribute, refund, claimRWA, ownerWithdraw).
- Extend validator logic to support stablecoin contributions instead of ADA.
- Implement claim RWA token and burn functionality in validator + frontend.
- Optimize script execution units for scalability.

Next steps:

- Run stress tests on Cardano Preprod Testnet.

- Conduct security audit to verify resistance to replay/spoof attacks.
- Deploy prototype to Cardano Mainnet after validation.