SDG BLOCKCHAIN ACCELERATOR

Debugging and Testing Report – Template

1. Project Information

- **Project Name:** Unicorn

- **Challenge & UNDP Office:** UNDP OP CCIT

- **Document Version:** 1

2. Testing Approach

(Explain how you tested your Aiken smart contracts. Include both unit tests and testnet/emulator tests.)

- **Unit Testing:** Describe test coverage, structure of Aiken unit tests, and edge cases checked.

- **Integration Testing:** Describe emulator or testnet runs.

- **Edge Cases:** Mention invalid datum/redeemer, unauthorized signatures, insufficient collateral, etc.

Entry:

- We did not use Smart Contracts in our PoC
- Added unit testing on Back-End Services for creating Cardano donations, validating transaction info for the project

3. Error Logs

- We encountered and handled the following error states during Cardano donation integration:
- Wallet connection errors
- Issue: User unable to connect with Yoroi or Nami.
- Handling: Added fallback error messages and modal for wallet connection failure.
- Token balance fetching errors
- Issue: wallet.getBalance() sometimes returns unexpected decimals or empty array.
- Handling: Added helper getAdaBalance(wallet) and validation to safely extract lovelace balance.
- Transaction submission errors
- Error: TxSubmitFail with details like BabbageOutputTooSmallUTxO (caused by sending < 1 ADA).

- Handling: Added normalizeAmount() and minimum ADA check (MIN_ADA = 1) before building transaction.
- BigInt conversion errors
- Error: RangeError: The number 9.58 cannot be converted to a BigInt.
- Handling: Implemented helper toUnits(value, decimals) to safely convert decimal numbers (e.g. 8.5 ADA → 8500000 lovelace).
- Wrong network errors
- Error: WrongNetwork Testnet (fromList [Addr Mainnet ...]).
- Handling: Added wallet network ID check (wallet.getNetworkId()) to detect Mainnet vs Preprod mismatch.
- Anonymous donation edge case
- Issue: External link still displayed for anonymous donations.
- Handling: Updated conditional rendering to hide transaction link if donation.anonymous = true.

4. Resolved Issues

• *Wallet not connecting (Yoroi/Nami)*

  • *Cause: Missing error handling for unavailable wallet APIs.*

  • *Resolution: Added modal + fallback error messages when useWallet() fails.*

  • *Status: ✅ Resolved*

• *Balance not showing*

  • *Cause: wallet.getBalance() returning empty or decimals mismatch.*

  • *Resolution: Added getAdaBalance(wallet) helper to safely extract lovelace balance.*

  • *Status: ✅ Resolved*

• *Transaction rejected (BabbageOutputTooSmallUTxO)*

  • *Cause: Attempted to send < 1 ADA (violates min-ADA requirement).*

  • *Resolution: Added normalizeAmount() and MIN_ADA check before transaction build.*

  • *Status: ✅ Resolved*

• *BigInt conversion error (RangeError: 9.58 cannot be converted)*

  • *Cause: Directly passing decimals to BigInt.*

• Resolution: Implemented toUnits(value, decimals) helper to safely convert to smallest unit.

• Status: ✅ Resolved

• Wrong network (WrongNetwork Testnet vs Mainnet)

• Cause: Mismatch between wallet and destination address network.

• Resolution: Added wallet.getNetworkId() check to ensure correct network before donation.

• Status: ✅ Resolved

• Anonymous donation link visible

• Cause: Condition did not exclude anonymous transactions.

• Resolution: Updated conditional rendering to hide ExternalLink when donation.anonymous = true.

• Status: ✅ Resolved

5. Optimization Notes

- *Gas / Fee Estimation*
- *• Instead of hardcoding a buffer (e.g., 2 ADA), integrate a lightweight helper to fetch min UTxO and fee estimates dynamically via Blockfrost or MeshJS.*
- *• This avoids false “insufficient balance” alerts when the fee is lower.*
- *• Reusable Conversion Helpers*
- *• Standardized functions like normalizeAmount() and toUnits() ensure consistent handling of decimals, commas, and BigInt conversions across ADA and native tokens.*
- *• Prevents duplicate logic and recurring RangeError issues.*
- *• Wallet Network Detection*
- *• Added wallet.getNetworkId() check to auto-detect Mainnet vs Preprod/Testnet.*
- *• Optimizes UX by preventing users from sending to the wrong network and producing avoidable errors.*
- *• Responsive Image Loading*
- *• Replaced static <Image width={1920} /> with Next.js fill and CSS object-fit: cover.*
- *• Reduces layout shifts and optimizes image delivery across devices.*
- *• Conditional Rendering Cleanup*
- *• Donation links are now conditionally hidden for anonymous transactions, reducing clutter and avoiding privacy leaks.*
- *• Improved Error Logging*
- *• Standardized console error messages for transactions, wallet connections, and balance fetching.*

- *• Easier debugging across environments.*

6. Tools and Environments Used

- *Tools and Environments Used*
- *• Development Environment*
- *• Next.js (v14+) for frontend dApp development.*
- *• TypeScript & styled-components for strongly typed UI logic and styling.*
- *• Node.js (v18+) for local development.*
- *• Cardano SDKs & Libraries*
- *• MeshJS (@meshsdk/core, @meshsdk/react) for wallet connection, transaction building, and signing.*
- *• Blockfrost API for fetching token metadata, balances, and network info.*
- *• Testing & Debugging Tools*
- *• Postman for GraphQL API testing (e.g., createDonation mutations).*
- *• Console logging in browser DevTools for tracing wallet balances, transaction hashes, and API responses.*
- *• External APIs*
- *• Coingecko API for fetching ADA → USD conversion rates.*
- *• MuesliSwap API for Cardano token price discovery in ADA.*
- *• Wallets*
- *• Yoroi Wallet (extension) for transaction signing and testing ADA/native token transfers.*
- *• Also tested with Nami/Eternl for broader compatibility.*
- *• Networks*
- *• Cardano Mainnet (for real token metadata and production-like environment).*
- *• Cardano Preprod Testnet (for safe testing with test tokens).*

7. Remaining Issues / Next Steps

- *Transaction Fee Estimation*
- *• Need a more reliable way to estimate Cardano network fees before submission.*
- *• Current workaround uses buffer checks (hasSufficientBalance), but lacks precision.*
- *• Anonymous Donations Handling*
- *• UI condition for anonymous Cardano/EVM donations requires refinement (preventing transaction link exposure).*
- *• Token Handling Improvements*
- *• Standardize conversion functions for ADA (toUnits, normalizeAmount) and ensure consistent BigInt usage across ADA and native tokens.*
- *• Verify decimals handling for all supported Cardano tokens.*
- *• Wallet Detection & UX*
- *• Improve error handling when no wallet is connected or when the user is on the wrong network (Mainnet vs Preprod).*
- *• Add responsive UI feedback if wallet balance or selected token info fails to load.*
- *• Testing Coverage*

- *• Extend test cases for token transfers beyond ADA (e.g., SHEN, test tokens).*
- *• Validate donations with multiple Cardano wallets (Yoroi, Nami, Eternl).*
- *• API Integration*
- *• Ensure GraphQL mutation (createCardanoDonation) aligns with backend schema (types and variable mapping).*
- *• Implement proper error messaging for unauthorized or invalid transaction submissions.*
- *• Next Steps*
- *• Deploy fixes to staging for end-to-end donation flow testing.*
- *• Gather feedback from test users (using Yoroi/Nami wallets) on transaction UX.*
- *• Finalize minimum-ADA validation rules and sync with backend enforcement*