Marcus Rappenborg Kjærsgaard
23-11-2025

Multi-Currency Wallet Simulator

Black Box Test Design

This document describes the black-box test design for the Multi-Currency Wallet Simulator. Black-box testing uses the SRS as test basis and checks the system from the outside via its inputs and outputs (e.g. with equivalence partitioning, 3-boundary value analysis, decision tables, and state transition testing).

Equivalence Partitioning

Equivalence partitioning divides an input domain into partitions (valid and invalid) where any value inside a partition is assumed to be handled identically by the system. For each partition we select one or a few representative test values.

Wallet Balance EP

We apply EP to wallet balance to separate negative, zero, positive, and non-numeric values, so we can test each behaviour-relevant range with a single representative.

Partition type	Partition	Test case values
Invalid	] MIN DOUBLE, -0.01]	-1000000
Valid	0.00	0.00
Valid	[0.01, MAX DOUBLE [	+1000000
Invalid	Non-numeric / null	“abc”, null

Transaction Amount EP

We apply EP to transaction amounts to distinguish clearly invalid values (negative, zero, non-numeric) from valid positive amounts used in operations.

Partition type	Partition	Test case values
Invalid	] MIN DOUBLE, -0.01]	-1000000
Invalid	0.00	0.00
Valid	[0.01, MAX DOUBLE [	+1000000
Invalid	Non-numeric / null	“abc”, null

Currency EP

We apply EP to currency codes to distinguish supported currencies from unsupported or malformed values with a small set of representative examples.

Allowed currencies: DKK, EUR, USD

Partition type	Partition	Test case values
Valid	Supported currency code	“DKK”, “EUR”, “USD”

Invalid	Unsupported currency code	“GBP”
Invalid	Non-currency code / null	“dkk”, null

(3-value) Boundary Value Analysis

This Boundary Value Analysis (BVA) focuses on values at and around the edges between equivalence partitions. For this project 3-value BVA is applied for better testing, based on the requirements and mitigations in SRS and Risk Assessment.

Wallet Balance BVA

Partition type	Partition	Boundary values	Test case values
Invalid	] MIN DOUBLE, -0.01]	-0.01	-0.02, -0.01, 0.00
Valid	0.00	0.00	-0.01, 0.00, 0.01
Valid	[0.01, MAX DOUBLE [	0.01	0.00, 0.01, 0.02

Transaction Amount BVA

Partition type	Partition	Boundary values	Test case values
Invalid	] MIN DOUBLE, -0.01]	-0.01	-0.02, -0.01, 0.00
Invalid	0.00	0.00	-0.01, 0.00, 0.01
Valid	[0.01, MAX DOUBLE [	0.01	0.00, 0.01, 0.02

Decision Tables

Decision tables are used in this project to describe how combinations of input conditions affect whether money is moved and how transactions are recorded. Each column (R1, R2, …) represents a rule (one combination of condition outcomes).
Each row under Conditions is a Boolean property or none (Y/N/-), and each row under Actions shows which outcome applies for that rule

Below, three decision tables are defined for the main money-moving operations: deposit, withdraw, and exchange

Deposit – Decision Table

Deposit checks whether money may be added to an existing wallet.

	R1	**R2**	**R3**	**R4**	**R5**
Conditions					
Wallet exists	Y	N	Y	Y	Y
Status ACTIVE	Y	-	N	Y	Y
Amount valid	Y	-	-	N	Y
Currency supported & matches wallet	Y	-	-	-	N

Actions					
Increase balance	X				
Record successful transaction	X				
Record failed transaction		X	X	X	X

Withdraw – Decision Table

Withdraw checks whether money may be taken out of a wallet without violating the “no negative balance” rule.

	R1	**R2**	**R3**	**R4**	**R5**
Conditions					
Wallet exists	Y	N	Y	Y	Y
Status ACTIVE	Y	-	N	Y	Y
Amount valid	Y	-	-	N	Y
Amount ≤ current balance	Y	-	-	-	N
Actions					
Decrease balance	X				
Record successful transaction	X				
Record failed transaction		X	X	X	X

Exchange – Decision Table

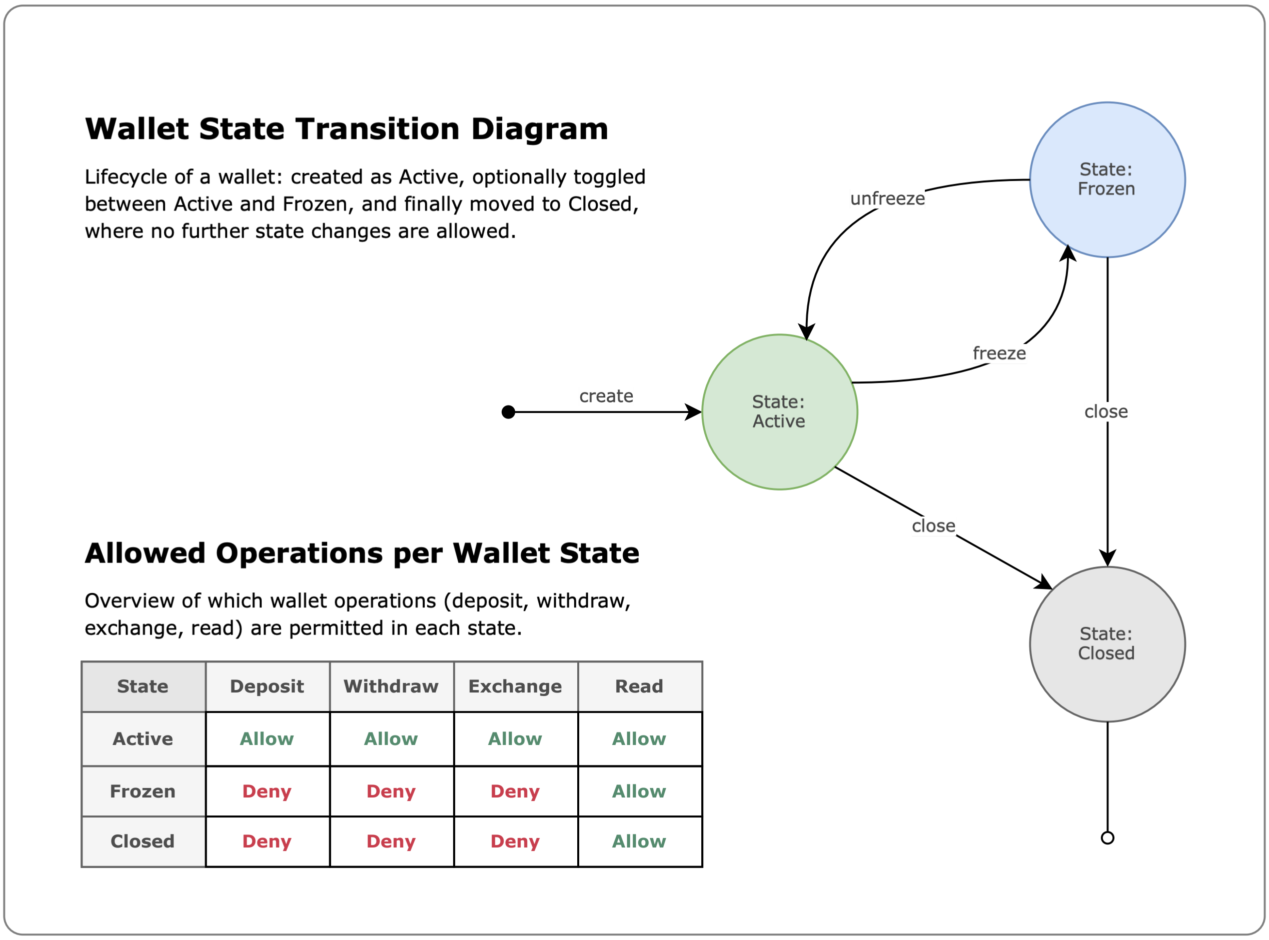
Exchange moves money between two wallets in possibly different currencies, using an external exchange rate.

	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**
Conditions							
Source wallet exists & ACTIVE	Y	N	Y	Y	Y	Y	Y
Target wallet exists & ACTIVE	Y	-	N	Y	Y	Y	Y
Amount valid (> 0 and numeric)	Y	-	-	N	Y	Y	Y
Amount ≤ source balance	Y	-	-	-	N	Y	Y
Currencies supported	Y	-	-	-	-	N	Y
Exchange rate available	Y	-	-	-	-	-	N
Actions							
Apply transaction (update both wallets)	X						
Record successful transaction	X						
Record failed transaction (no balance change)		X	X	X	X	X	X
Return failure response (external problem)							X

State Transition Diagram – Wallet lifecycle

State transition testing is applied to the wallet lifecycle, where behaviour depends on the current wallet status and the events applied to it.

The SRS defines three statuses for a wallet: *Active*, *Frozen*, and *Closed*. Wallets are created in the Active state, can be toggled between *Active* and *Frozen*, and can finally be moved to *Closed*, where no further state changes are allowed.



Wallet State Transition Diagram

Lifecycle of a wallet: created as Active, optionally toggled between Active and Frozen, and finally moved to Closed, where no further state changes are allowed.

Allowed Operations per Wallet State

Overview of which wallet operations (deposit, withdraw, exchange, read) are permitted in each state.

State	Deposit	Withdraw	Exchange	Read
Active	Allow	Allow	Allow	Allow
Frozen	Deny	Deny	Deny	Allow
Closed	Deny	Deny	Deny	Allow

From the diagram we can derive several useful coverage criteria:

- State coverage – at least one test reaches each state
  - e.g. create → Active, create + freeze → Frozen, create + close → Closed.
- Transition coverage – at least one test exercises each allowed transition
  - Active → Frozen, Frozen → Active, Active → Closed, Frozen → Closed.
- Invalid transition coverage – tests that try transitions not in the diagram
  - e.g. freeze (Closed), unfreeze (Closed), close (Closed) again.

The state diagram is thus used as a test basis: it defines which sequences of events must be supported, which must be rejected, and how that interacts with wallet transaction behaviour in the rest of the test suite.