Boundary Value Testing

deposittoAcc
Inputs:

- String “username”
- String “accountName”
- Double “amount”

Username

- Must be a valid username in the system
- Since this input can’t have a min/max value system, for the purpose of testing with this method the username will be treated as either valid or invalid (T/F)

Account

- Valid account type that exists
- Since this input can’t have a min/max value system, for the purpose of testing with this method the accountType will be treated as either valid or invalid (T/F)

Amount:

- 0 < amount <= 10,000
- Min: 1
- Min- = 0
- Min+ = 2
- Nom = 5,000
- Max: = 10,000
- Max- = 9,999
- Max+ = 10,001

Test Cases

Test Case	username	accountName	Amount	Expected Result
1	Valid	Valid	0	Reject Deposit
2	Valid	Valid	1	Accept deposit
3	Valid	Valid	2	Accept deposit
4	Valid	Valid	5000	Accept Deposit
5	Valid	Valid	9,999	Accept Deposit
6	Valid	Valid	10,000	Accept Deposit
7	Valid	Valid	10,001	Reject Deposit

withdrawFromAcc
Inputs

- String username
- String accountType
- Double withdrawAmount

Username

- Must be a valid username in the system
- Since this input can’t have a min/max value system, for the purpose of testing with this method the username will be treated as either valid or invalid (T/F)

accountType

- Valid account type that exists
- Since this input can’t have a min/max value system, for the purpose of testing with this method the accountType will be treated as either valid or invalid (T/F)

WithdrawAmount

- 0 < x <= 5,000
- Min- = 0
- Min = 1
- Min+ = 2
- Nom = 2500
- Max- = 4999
- Max = 5000
- Max+ = 5001

Test Cases

Test Case	username	accountType	withdrawAmount	Expected Result
1	valid	valid	0	Reject withdrawal
2	valid	valid	1	Accept Withdrawal
3	valid	valid	2	Accept Withdrawal
4	valid	valid	2500	Accept Withdrawal
5	valid	valid	4999	Accept

				Withdrawal
6	valid	valid	5000	Accept Withdrawal
7	valid	valid	5001	Reject withdrawal

transferFunds
Inputs:

- String username
- String fromAccountType
- String toAccountType
- Double Amount

Username

- Must be a valid username in the system
- Since this input can’t have a min/max value system, for the purpose of testing the username will be treated as either valid or invalid

fromAccountType

- Valid account type that exists
- Since this input can’t have a min/max value system, for the purpose of testing the fromAccountType will be treated as either valid or invalid

toAccountType

- Valid account type that exists
- Since this input can’t have a min/max value system, for the purpose of testing the toAccountType will be treated as either valid or invalid

Amount

- 0 < x <= 7500
- Min- = 0
- Min = 1
- Min+ = 2
- Nom = 3750
- Max- = 7499
- Max = 7500
- Max+ = 7501

Test Cases

Test Case	username	fromAccount Type	toAccountTyp e	amount	Expected Result

1	valid	valid	valid	0	Reject Transfer
2	valid	valid	valid	1	Accept Transfer
3	valid	valid	valid	2	Accept Transfer
4	valid	valid	valid	3750	Accept Transfer
5	valid	valid	valid	7499	Accept Transfer
6	valid	valid	valid	7500	Accept Transfer
7	valid	valid	valid	7501	Reject Transfer

Equivalence Based Testing (Weak Robust)

depositToAcc
Inputs

- String username
- String accountName
- Double amount

Username (u)
V1: u: { u = “appLogin” || “testLogin” }
NV1: u: {u != “appLogin” || “testLogin” }

accountType (at)
V2: at: { at == “chequing” || “savings” }
NV2: at: {at != “chequing” || “savings” }

amount (a)
V3: a: { 0 < a <= 10000}
NV3: a: {a <= 0}
NV4: a: {10000 < a}
NV5: a: {a == “abc”} (any string)

Test Case	username	accountName	amount	Expected Result
1	testLogin	savings	5000	Successful

				Deposit
2	guest	chequing	5000	Reject Deposit
3	testLogin	investment	5000	Reject Deposit
4	testLogin	savings	-20	Reject Deposit
5	testLogin	savings	12000	Reject Deposit
6	testLogin	savings	abc	Reject Deposit

withdrawFromAcc

Inputs

- String username
- String accountType
- Double withdrawAmount

Username (u)

V1: u: { u = “appLogin” || “testLogin” }

NV1: u: {u != “appLogin” || “testLogin” }

accountType (at)

V2: at: { at == “chequing” || “savings” }

NV2: at: {at != “chequing” || “savings” }

withdrawAmount (wa)

V3: wa: { 0 < wa <= 5000}

NV3: wa: {wa <= 0}

NV4: wa: {5000 < wa}

NV5: wa: {wa == “abc”} (any string)

Test Case

Test Case	username	accountType	withdrawAmount	Expected Result
1	appLogin	savings	3000	Successful withdrawal

2	guest	savings	3000	Rejected
3	appLogin	investment	3000	Rejected
4	appLogin	savings	-20	Rejected
5	appLogin	savings	7000	Rejected
6	appLogin	savings	abc	Rejected

TransferFunds
Inputs

- String username
- String fromAccountType
- String toAccountType
- Double amount

Username (u)
V1: u: { u = “appLogin” || “testLogin” }
NV1: u: {u != “appLogin” || “testLogin” }

fromAccountType (at)
V2: at: { at == “chequing” || “savings” }
NV2: at: {at != “chequing” || “savings” }
NV3: at: {at != toAccountType }

toAccountType (tat)
V3: tat: { tat == “chequing” || “savings” }
NV4: tat: { tat != “chequing” || “savings” }
NV5: tat: { tat != fromAccountType }

Amount (aa)
V4: a: { 0 < a <= 7500}
NV6: a: {a <= 0}
NV7: a: {7500 < wa}
NV8: a: {a == “abc”} (any string)

Test Case

Test Case	username	fromAccount Type	toAccountTyp e	withdrawAt	Expected Result
1	appLogin	savings	chequing	3000	Successful transfer

2	guest	savings	chequing	3000	Rejected
3	appLogin	investment	chequing	3000	Rejected
4	appLogin	savings	investment	3000	Rejected
5	appLogin	savings	savings	3000	Rejected
6	appLogin	savings	chequing	-20	Rejected
7	appLogin	savings	chequing	8000	Rejected
8	appLogin	savings	chequing	abc	Rejected