# Assignment 5 - White Box Testing Decision Coverage Method for Withdraw

#### 1) Code Sections (in *Backend.py*)

```
# function input transaction summary file and master account file
# if transaction summary valid then return valid master account file data (data type: dictionary)
# if transaction summary not valid then immediately stop
def get_new_master_accounts(self, file1, file2):
   master_list = self.read_master_accounts(file1)
    trans_list = self.read_transaction(file2)
    for i in trans_list:
        if i[0] == "DEP":
            if i[1] in master_list:
                balance = int(master_list[i[1]][0]) + int(i[2])
                master_list[i[1]][0] = str(balance)
                print("Error! " + str(i[1]) + " not in the master account file!")
                return False
    1 elif i[0] == "WDR":
            if i[1] in master_list:
    2
                balance = int(master_list[i[1]][0]) - int(i[2])
                if balance < 0:
                    print("Error! " + str(i[1]) + " have a negative balance!")
                    return False
                else:
                    master_list[i[1]][0] = str(balance)
                print("Error! " + str(i[1]) + " not in the master account file!")
                return False
    for i in master_list:
        line = str(i) + " " + master_list[i][0] + " " + master_list[i][1] + "\n"
     4 if len(line) > 47: # check each line have more than 47 characters or not
            print("Error! Each line is at most 47 characters!")
    return master_list # return a valid master dictionary
```

#### 2.1) Analysis of Test Cases

In the code for *withdraw*, there are branches (if statements) to complement the constraints, such as non-negative balance, valid account number and character limit for each line in *MasterAccountsFile.txt*. Decision coverage method covers both sides (true and false) of each decision, which is appropriate to test the constraints and requirements. The completion criterion is to design a test case for each side of each decision. In *Backend.py*, there are 4 decisions for withdraw section.

#### 2.2) Design of the Test Cases:

For all cases, the input files are *TransactionSummaryFile.txt* and *MasterAccountsFile.txt*.

Notes for the table:

```
-i in trans_list: the element in trans_list, which is the input reading from TransactionSummaryFile.txt
e.g. ['WDR', '1234567', '100000', '0000000', 'theo']
-i[0]: the transaction code, e.g. "NEW"
-i[1]: the account number, e.g. "1234567"
-i[2]: the withdraw amount, e.g. "200000"
```

Decision	i[0] input	i[1] input	i[2] input	Test	i[0]	i[1]	i[2]
1: true	"WDR"	account	withdraw	T1	"WDR"	account number	withdraw amount
		number	amount				
1: false	transaction	account	withdraw	T2	transaction code	account number	withdraw amount
	code other	number	amount		other than		
	than "WDR"				"WDR"		
2: true	"WDR"	account	withdraw				
		number	amount				
2: false	"WDR"	account	withdraw	T3	"WDR"	account number	withdraw amount
		number	amount				
3: true	"WDR"	account	withdraw	T4	"WDR"	account number	withdraw amount
		number	amount				
3: false	"WDR"	account	withdraw				
		number	amount				

Notes for the table and code:

- -i in master\_list element in trans\_list reading from *MasterAccountsFile.txt*-line: a string of combination of a key and its value, e.g. "2233445 100000 Kelly"

Decision	i in trans_list	line input	Test	i	line
	input				
4: true	a list of one line transatcion items separated by space	string of one line transaction	T5	a list of one line transatcion items separated by space	string of one line transaction
4: false	a list of one line transatcion items separated by space	string of one line transaction	Т6	a list of one line transatcion items separated by space	string of one line transaction

## 3) Creation of Test Inputs for Test Cases

		in	put based on	code	input from	file		expected of	output
Test	Purpose	i[0] input	i[1]	i[2]	input from	input from	expected	expected	expected terminal
			input	input	TransactionSum	MasterAc	MasterAc	ValidAccoun	output
					maryFile.txt	countsFile	countsFil	tsList.txt	
						.txt	e.txt		
T1	withdraw	"WDR"	"1234567"	"100000"	'WDR 1234567	"1234567	"1234567	"1234567\n"	"New Master
	100000				100000 0000000	200000	100000	"000000"	Accounts File
	cents from a				Theo\n', 'EOS	Theo\n"	Theo\n"		created
	valid				0000000 000				successfully!"
	account				0000000 ***'				"New Valid
	successfully								Accounts File
									created
									successfully!"
T2	do	"DEP"	"1234567"	"200000"	'DEP 1234567	"1234567	"1234567	"1234567\n"	"New Master
	transactions				100000 0000000	200000	300000	"0000000"	Accounts File

	other than withdraw on a valid account successfully (deposit 100000 cents succesfully)				Theo\n', 'EOS 0000000 000 0000000 ***'	Theo\n"	Theo\n"		created successfully!" "New Valid Accounts File created successfully!"
Т3	error when a valid account is not in the MasterAccountsFile.txt	"WDR"	"1122334"	"100000"	100000 0000000	200000	"1234567 200000 Theo\n"	"0000000"	'Error! 1122334 not in the master account file!'
Т4	error when withdraw over the balance( withdraw 300000 cents from an account has only 100000 cents)	"WDR"	"1234567"	"300000"	300000 0000000	100000	"1234567 100000 Theo\n"	"1234567\n" "0000000"	"Error! 1234567 have a negative balance!"

		in	put based	on code	input fr	om file	expec	ted output
Test	Purpose	i	line	input from	input from	expected	expected	expected
				TransactionSu	MasterAccou	MasterAccou	ValidAccounts	Terminate output
				mmaryFile.txt	ntsFile.txt	ntsFile.txt	List.txt	_
T5	Error when one	['WDR',	"223344	'WDR 2233445	"2233445	"2233445	"2233445\n"	"Error! Each line
	line in	'2233445',	5	100000	12345678901	12345678901	"0000000"	is at most 47
	"TransactionSumm	'100000',	1234567	0000000	23456789012	23456789012		characters!"
	aryFile.txt "	'0000000',	8901234	KellyKellyKell	3	3		
	exceeds 47	'KellyKell	5678901	yKe\n', 'EOS	KellyKellyKe	KellyKellyKe		
	characters	yKellyKe'	23	000 000000	llyKe\n"	llyKe\n"		
	(the balance has 23	]	KellyKel	0000000 ***'				
	digits, name has 17		lyKellyK					
	characters with		e"					
	code, account							
	number, spaces							
	and "\n"space,							
	then 50 characters							
	in total)							
Т6	withdraw	['WDR',	"123456	'WDR 2233445	"2233445	"2233445	"2233445\n"	"New Master
	successfully when	'1234567',	7 100000	100000	12345678901	12345678901	"0000000"	Accounts File
	each line is within	'100000',	theo"	0000000	23456789	23356789		created
	47 characters(e.g.	'0000000',		KellyKellyKell	KellyKellyKe	KellyKellyKe		successfully!"
	withdraw 100000	'KellyKell		yKe\n', 'EOS	llyKe\n"	llyKe\n"		"New Valid

cents)	yKellyKe'	0000000 000		Accounts File
		0000000 ***'		created
				successfully!"

## 4.1) Test Report - Resulting of the testing

Decision Coverage Method for testing withdraw

Note: if terminal outputs "New Master Accounts File created successfully!" and "New Valid Accounts File created successfully!", then it means the files are generated successfully

		input		output			
Test	Purpose	input from	input from	updated	updated	Terminate output	
		"TransactionSum	"MasterAccount	"MasterAccountsF	"ValidAccoun		
		maryFile.txt"	sFile.txt"	ile.txt"	tsList.txt"		
T1	withdraw 100000 cents	'WDR 1234567	"1234567	"1234567 100000	"1234567\n"	"New Master Accounts	
	from a valid account	100000 0000000	200000 Theo\n"	Theo\n"	"0000000"	File created	
	successfully	Theo\n', 'EOS				successfully!"	
		000 000000				"New Valid Accounts	
		0000000 ***'				File created	
						successfully!"	
T2	do transactions other	'DEP 1234567	"1234567	"1234567 300000	"1234567\n"	"New Master Accounts	
	than withdraw on a	100000 0000000	200000 Theo\n"	Theo\n"	"0000000"	File created	
	valid account	Theo\n', 'EOS				successfully!"	
	successfully(Deposit	000 000000				"New Valid Accounts	
	100000 cents	0000000 ***'				File created	
	succesfully)					successfully!"	
Т3	error when the account	'WDR 1122334	"1234567	"1234567 200000	"1234567\n"	'Error! 1122334 not in	
	is not in the	100000 0000000	200000 Theo\n"	Theo\n"	"0000000"	the master account file!'	
	"MasterAccountsFile.t	Theo\n', 'EOS					
		000 000000					
		0000000 ***'					
T4	error when withdraw	'WDR 1234567	"1234567	"1234567 100000	"1234567\n"	"Error! 1234567 have a	
	( 0	300000 0000000	100000 Theo\n"	Theo\n"	"0000000"	negative balance!"	
	withdraw 300000 cents	•					
		000 000000					
	,	0000000 ***'					
T5	error when one line in	'WDR 2233445	"2233445	"2233445	"2233445\n"	"Error! Each line is at	
	"TransactionSummary	100000 0000000		123456789012345	"0000000"	most 47 characters!"	
		5 5		67890123			
				KellyKellyKellyK			
	(e.g. the balance has 23		Ke∖n"	e\n"			
	0	0000000 ***'					
	characters with code,						
	account number,						
	spaces and "\n"space,						
	then 50 characters in						
	total)						
Т6	withdraw successfully	'WDR 2233445	"2233445	"2233445	"2233445\n"	"New Master Accounts	
	when each line is	100000 0000000		123456789012335		File created	
	within 47	KellyKellyKelly	456789	6789		successfully!"	

characters(e.g.	Ke\n', 'EOS	KellyKellyKelly	KellyKellyKellyK	"New Valid Accounts
withdraw 100000	000 000000	Ke\n"	e∖n"	File created
cents)	0000000 ***'			successfully!"

# 4.2) Failure Report

Test name	Purpose (what it was testing)	Nature of Failure (How the output is wrong)	Error in code	How to Fix
T4	error when withdraw over the balance(e.g. withdraw 300000 cents from an account has only 100000cents)	I Master Accounts rue ext	In Backend.py, there are two missing spaces when print error message for withdraw	In Backend.py, add space to the printing statement in the get_new_master_accounts()