

Back Office Unit Testing for Withdraw Transactions

Team Name: Runtime Terror

Team Member

Anna Chulukov, Faranak Sharifi, Marwan El Khodary, Ryan Fernandes

Transaction.

conditon		accountNum			
number	masterAccountDict input	input	moneyHave	amount	Test
1	{'1234567': ['10', 'my account'], '1122334': ['1000', 'sdads']}	1234567	0	10	Test-1
1	{'1234567': ['10', 'my account'], '1122334': ['1000', 'sdads']}	1234567	-1	11	Test-2
1	{'1234567': ['10', 'my account'], '1122334': ['1000', 'sdads']}	1234567	1	9	Test-3

Code Snippet from backend.py of withdraw method

```
102 def withdraw(masterAccountDict, accountNum, amount)
103     # see how much money they have
104     moneyHave = masterAccountDict[accountNum][0]
105     # subtract it by the amount they want to withd
106     moneyHave = int(moneyHave) - int(amount)
107     if moneyHave < 0:
108         print("Do not have enough funds!")
109         return masterAccountDict
110     # store new amount of money back into the dict
111     masterAccountDict[accountNum][0] = str(moneyHa
112     return masterAccountDict
```

For the withdraw transaction, we followed the decision coverage method of white box testing. We did not do condition or loop coverage since our code did not require it. The completion criteria was to create test cases that would cover the three possible decisions for our if statement in that function. Test 1 checks if the user withdraws ten dollars, and they only have ten dollars, then the amount left in the account will be zero, and thus no output message is given since everything is valid. On the other hand, test 2 checks if the user withdraws more

Output

None

"Do not have enough funds!"

None

```
t) :  
  
draw  
  
tionary  
ave)
```