

## Transaction Manager Testcases

classname: TransactionManagerController method: <code>public boolean OpenAccount(ActionEvent event) //Handles account creation</code>			
Test Case #	Requirement	Test description Data	Expected result/output
1	Successfully open up checking account	First name: Arun Last name: Felix DOB: 1/3/2003 Account type: Checking Initial: 1	true
2	Successfully open up College Checking w/ NB	First name: Thomas Last name: Buech DOB: 1/3/2002 Account type: College Checking Initial Deposit: 13 Campus: New Brunswick	true
3	Successfully open up College Checkings account w/ NW	First Name: FirstNW Last Name: LastNW DOB: 1/4/2003 Account type: College Checking Initial Deposit: 13 Campus: Newark	true
4	Successfully opens up College checking account w/ camden	First Name: FirstC Last Name: LastC DOB: 1/4/2003 Account type: College Checking Intial Deposit: 13 Campus: Camden	true
5	Successfully Opens saving account with loyal	First Name: FirstSL Last Name: LastSL DOB: 2/3/2001 Account type: Savings Initial Deposit: 12 Loyal: yes	true

6	Successfully Opens saving account with loyal	First Name: FirstSU Last Name: LastSU DOB: 2/3/2001 Account type: Savings Initial Deposit: 12 Loyal: no	true
7	Successfully Opens Money Market Account	First Name: FirstMM Last Name: LastMM DOB: 2/1/2000 Account type: Money Market Initial deposit: 2000	true
8	Inputs future date	First Name: capierre Last Name: Charme DOB: 11/5/2035 Account type: Checking Initial Deposit: 2000	false
9	Client attempts to input user under 16 years old	First Name: Michael Last Name: Adelson DOB: 11/5/2013 Account type: Checking Initial Deposit: 2000	false
10	Client attempts to input negative balance	First Name: Kal Last Name: Pandit DOB: 1/2/2003 Account type: Checking Initial Deposit: -3	false
11	Client inputs invalid balance	First Name: Ryoji Last Name: Kaji DOB: 1/1/1992 Account type: Checking Initial deposit: 83ba	false
12	Client inputs invalid Date	First Name: INVALID Last Name: DDD DOB: 1/sdre Account type:	false

		Checking Initial deposit: 1000	
13	Client doesn't select Account type	First Name: Hengyi Last Name: Wang DOB: 1/2/1999 Account Type: Initial deposit: 10000	false
14	Client doesn't add First name	First name: Last name: NF DOB: 1/2/1998 Account Type: Checking Initial Deposit: 1000	false
15	Client doesn't add Last name	First name: NL Last name: DOB: 1/2/1994 Account Type: Checking Initial Deposit: 1000	false
16	Client doesn't add deposit	First Name:ND Last Name: ND DOB: 1/2/1995 Account type: Checking Initial Deposit:	false
17	Client doesn't select account type	First Name: FN Last Name: noaccount DOB: 1/3/1983 Account type: Initial deposit: 20	false
18			

Class: TransactionManagerController

Method: `private void openCollegeCheckingAccount(Profile profile, double balance, Date date)`

1	The user attempting to register for an	First name: * Last name: *	Show "Age cannot be more than 24"
---	---	-------------------------------	--------------------------------------

	account at an age of 24 or greater	DOB: 01/03/1964 Account type: College Checking Initial deposit 2000	for College Checking"
--	------------------------------------	---	-----------------------

Class: AccountDatabase Method: <code>public boolean open(Account account)</code>			
id	requirement	input	output
1	Must not open duplicate accounts	First Name: A Last Name: A DOB: 1/2/2001 Account type: Checking Initial deposit: 1 <b>Insert twice</b>	Will add 1st acct to database, but will not add 2nd acct, and will return false.

Classname: Transactionmanagercontroller Method: <code>private void openMoneyMarketAccount(Profile profile, double balance)</code>			
id	requirements	input	output
1	User attempts to open Money market account with less than 2000 dollars	First Name: A Last Name: A DOB: 1/2/2001 Account type: Money Market Initial deposit: 1999	It will not make account, will print: "Balance cannot be below 2000!"

Classname: Transactionmanagercontroller Method: <code>private boolean deposit(ActionEvent event)</code>			
id	requirements	input	output
1	User tries to deposit without selecting the account type.	First Name: A Last Name: A DOB: 1/2/2001	Prints Please Select an Account type!" and returns false

		Account type: deposit: 1999	
2	User inputs invalid date or no date	First name: Arun Last name: Felix DOB:s/324/21 Account type: Checking amount	It will print "Please enter a valid date!" and return false
3	User attempts to input no amount or invalid	First name: Arun Last name: Felix DOB:s/324/21 Account type: Checking amount: aoeu	Prints "Please enter a valid amount to deposit" and returns false
4	User attempts to deposit to account that doesn't exist,	First name: Arun Last name: Felix DOB: 1/23/3 Account type: Checking amount: 192832	Prints "Account does not exist!" and returns false.
5	User successfully deposits money to account	First name: Arun Last Name: Felix DOB: 1/3/2003 Account type: Checking Amount: 1242 -we assume this account was already added	Prints "Deposit successful" and returns true.

Classname: TransactionManagerController Method: <code>private void withdraw(ActionEvent event)</code>			
id	requirement	input	output
1	User doesn't input account type	First name: Thomas Last name: buech Acct type: Amount: 5000	Print "Please Select an Account type!" and return false.
2	User puts in invalid	First: Arun	Prints "Please enter a

	amount	Last: Felix DOB: 1/3/2003 Acct type: Checking Amount: asd	valid account to withdraw!" and returns false.
3	User inputs invalid date	First: Arun Last: Felix DOB: 1/3/2asd Acct type: Checking Amount: 123	Prints "Please enter a valid date!" and returns false.
4	User inputs account info that doesn't exist	First: Capierre Last: Charme DOB: 1/4/2003 Acct type: checking Amount: 4	Prints "Account does not exist!" and
5	User successfully withdraws money from account	First: Arun Last: Felix DOB: 1/3/2003 Acct type: checking Amount:50  -Assuming the account was created and there's enough money in the account.	Withdraws successfully and returns true.
6	User attempts to withdraw money from account with inadequate funds	First: Arun Last: Felix DOB: 1/3/2003 Acct type: Checking amount : 100 -assuming account exists and there's inadequate funds	Prints "Cannot Withdraw - insufficient funds in the account: "

Classname: TransactionManagerController Method: <code>void CloseAccount(ActionEvent event)</code>			
id	requirements	input	output
1	Account type is not selected	First Name: Mannan Last Name: Shukla DOB: 1/4/2004	Will print out "Please Select an Account type!"

		Acct type:	
2	User inputs invalid date or no date	First Name: Thomas Last Name: Buch DOB: as34r Acct type: checking	Print "Please enter a valid date! "
3	User successfully closes account	First name: Arun Last name: Felix DOB: 1/3/2003 Acct type: checking	Print "Account Closed Successfully!"