

Class name: **TransactionManagerController**

Method signature: **private void OpenAccount(String firstName, String lastName, Date dateOfBirth, String accountType, String campus, boolean isLoyal, Double initialDeposit)**

Test Case #	Requirement	Test description and Input data	Expected Result
1	The method shouldn't open an account if initial deposit = 0	Input valid profile information. Make initial deposit 0.	"Initial deposit cannot be 0 or negative."
2	The method shouldn't open an account if initial deposit < 0	Input valid profile information. Make initial deposit a negative number	"Initial deposit cannot be 0 or negative."
3	The method shouldn't open an account if no initial deposit is entered	Input valid profile information. Leave the initial deposit text box blank.	"Please enter a valid amount."
4	The method shouldn't open an account if first name is empty	Input last name, valid dob, account type, and valid deposit. Leave first name empty	"Please enter a first name"
5	The method shouldn't open an account if last name is empty	Input first name, valid dob, account type, and valid deposit. Leave last name empty	"Please enter a last name"
6	The method shouldn't open an account if dob is empty	Input first name, last name, account type, and valid deposit. Leave dob empty	"Please enter a date of birth."
7	The method shouldn't open an account if account holder is under 16	Input valid profile information. Input date of birth that is under 16	"DOB invalid:" + dateOfBirth + " under 16."

8	The method shouldn't open a College Checking account if account holder is not under 24.	Input valid profile information. Input date of birth that is not under 24.	"DOB invalid: " + dateOfBirth + " over 24."
9	The method shouldn't open a College Checking account if account holder already has a Checking account.	Add Checking account to database and with the same profile, attempt to add a College Checking account.	firstName + lastName + dateOfBirth + "(CC) is already in the database"
10	The method shouldn't open a Checking account if account holder already has a College Checking account.	Add College Checking account to database and with the same profile, attempt to add a Checking account	firstName + lastName + dateOfBirth + "(C) is already in the database"
11	The method shouldn't open a Money Market account if initial deposit is < 2000	Input valid profile information. Select Money Market account. Input an initial deposit that is less than 2000	"Minimum of \$2000 to open a Money Market account."

Class name: **TransactionManagerController**

Method signature: **private void CloseAccount(String firstName, String lastName, Date dateOfBirth, String accountType)**

Test Case #	Requirement	Test description and Input data	Expected Result
1	The method can't close an account that doesn't exist in the database	Make sure the database is empty, select "close", input valid profile information and account type.	firstName + lastName + dateOfBirth + accountType + is not in the database"
2	The method shouldn't	Input last name, valid	"Please enter a first

	close an account if first name is empty	dob, account type, and valid deposit. Leave first name empty	name”
3	The method shouldn’t close an account if last name is empty	Input first name, valid dob, account type, and valid deposit. Leave last name empty	“Please enter a last name”
4	The method shouldn’t close an account if dob is empty.	Input first name, last name, account type, and valid deposit. Leave dob empty	“Please enter a date of birth.”

Class name: **TransactionManagerController**  
Method signature: **private void DepositToAccount(String firstName, String lastName, Date dateOfBirth, double amount, String accountType)**

Test case #	Requirement	Test description and Input data	Expected result/output
1	The method shouldn’t deposit to an account if first name is empty	Input last name, valid dob, account type, and valid deposit. Leave first name empty	“Please enter a first name”
2	The method shouldn’t deposit to an account if last name is empty	Input first name, valid dob, account type, and valid deposit. Leave last name empty	“Please enter a last name”
3	The method shouldn’t deposit to an account if dob is empty.	Input first name, last name, account type, and valid deposit.	“Please enter a date of birth.”

		Leave dob empty	
4	The method shouldn't deposit to an account that doesn't exist	Make sure the database is empty, select "deposit", input valid profile information, account type, and amount to deposit.	firstName + lastName + dateOfBirth + accountType + is not in the database"
5	The method shouldn't deposit an amount that is 0	Open any account. Attempt to deposit to that account, enter 0 as the deposit amount	"Deposit - amount cannot be 0 or negative."
6	The method shouldn't deposit an amount that is negative	Open any account. Attempt to deposit to that account, enter a negative number as the deposit amount	"Deposit - amount cannot be 0 or negative."
7	The method cannot deposit an amount if no amount is entered	Open any account. Attempt to deposit to that account, leave the deposit text box blank	"Please enter a valid amount."

Class name: **TransactionManagerController**

Method signature: **private void WithdrawFromAccount(String firstName, String lastName, Date dateOfBirth, double amount, String accountType)**

Test case #	Requirement	Test description and Input data	Expected result/output
1	The method shouldn't withdraw from an account if first name is empty	Input last name, valid dob, account type, and valid withdraw amount. Leave first name empty	"Please enter a first name"

2	The method shouldn't withdraw from an account if last name is empty	Input first name, valid dob, account type, and valid withdraw amount. Leave last name empty	"Please enter a last name"
3	The method shouldn't withdraw from an account if dob is empty	Input first name, last name, account type, and valid deposit. Leave dob empty	"Please enter a date of birth."
4	The method shouldn't withdraw from an account that doesn't exist	Make sure the database is empty, select "withdraw", input valid profile information, account type, and amount to withdraw.	firstName + lastName + dateOfBirth + accountType + is not in the database"
5	The method shouldn't withdraw from an amount that is 0	Open any account. Attempt to withdraw from that account, enter 0 as the withdraw amount	"Withdraw - amount cannot be 0 or negative."
6	The method shouldn't withdraw from an amount that is negative	Open any account. Attempt to withdraw from that account, enter a negative number as the withdraw amount	"Withdraw - amount cannot be 0 or negative."
7	The method cannot withdraw from an amount if no amount is entered	Open any account. Attempt to withdraw from that account, leave the withdraw text box blank	"Please enter a valid amount."

Class name: **Date**

Method signature: **public boolean isValid()**

Test case #	Requirement	Test description and	Expected Result
-------------	-------------	----------------------	-----------------

		Input data	
1	The method should return true if a valid date is given	Create an instance of Date with valid day and month and year  Test date: "01/14/1999"	true
2	The method should return false for any date given with a year before 1900	Create an instance of Date with valid day and month but with a year < 1900.  Test date: "11/21/800"	false
3	The method should return false for any date given with an invalid month	Create an instance of Date with valid day and year but invalid month.  Test date: "13/10/2002"	false
4	The method should return false for any date given with an invalid day	Create an instance of Date with valid month and year but invalid day.  Test date: "10/33/2000"	false
5	The method should return false for any invalid leap year date	Create an instance of Date with valid month and year but invalid day  Test date: "02/29/1999"	false