Test Specification for isValid() Method of the Date Class

Class name: Date			
Method signature: public boolean isValid(){} // check if a given date is a valid calendar date			
Test Case #	Requirement	Test Description and Input Date	Expected Result / Output
1	The number of days in February for a non- leap year shall be 28. The method shall return false if the date given has 29 days for a non-leap year.	• Create an instance of Date with non-leap year, month=2, and day>28 • Test Date: "02/29/2011"	false
2	The number of days in February for a leap year shall be 29. The method shall return true if the date given has 29 days for a leap year.	• Create an instance of Date with leap year, month=2, and day=29 • Test Date: "02/29/2016"	true
3	The number of months in a year is 12. The method shall return false if the year given is greater than 12.	• Create an instance of Date with a month > 12 • Test Date: "13/5/2023"	false
4	The number of months in a year is 12. The method shall return false if the year given is less than 1.	• Create an instance of Date with a month < 1 • Test Date: "0/5/2023"	false
5	The number of days in 31-day months should not exceed 31. The method shall return false if the days given are greater than 31.	• Create an instance of Date with a 31-day month with a day > 31 • Test Date: "10/32/2023"	false
6	The number of days in 30-day months should not exceed 30. The method shall return false	• Create an instance of Date with a 30-day month with a day > 30 • Test Date: "11/31/2023"	false

	if the days given are greater than 30.		
7	The number of days in a month is at least 1. The method shall return false if the days given are less than 1.	● Create an instance of Date with a day < 1 ● Test Date: "10/0/2023"	false
8	The method shall return true if the year, month, and day given exist.	● Create an instance of Date with a valid year, month, and day ● Test Date: "1/10/2024"	true

Test Specification for the Open Button [on Open/Close tab]

Class name: TransactionManagerController **Method signature:** openButtonClicked(){} //opens an account using the information entered by the user			
Test Case #	Requirement	Test Description and Input Date	Expected Result / Output
1	First name is required to open an account. A missing information pop-up shall appear if the first name is not provided.	- First Name: - Last Name: Doe - Date of Birth: 11/01/2000 - Checking - Balance: 5000	Pop-Up : Missing Information Error Message
2	Last name is required to open an account. A missing information pop-up shall appear if the last name is not provided.	- First Name: John - Last Name: - Date of Birth: 11/01/2000 - Checking - Balance: 5000	Pop-Up : Missing Information Error Message
3	Date of birth is required to open an account. A missing information pop-up shall appear if the date of birth is not provided.	- First Name: John - Last Name: Doe - Date of Birth: - Checking - Balance: 5000	Pop-Up : Missing Information Error Message
4	Account type is required to open an account. A missing information pop-up shall appear if the account type is not provided.	- First Name: John - Last Name: Doe - Date of Birth: 11/01/2000 - Balance: 5000	Pop-Up : Missing Information Error Message
5	Balance is required to open an account. A missing information pop-up shall appear if the balance is not provided.	- First Name: John - Last Name: Doe - Date of Birth: 11/01/2000 - Checking - Balance:	Pop-Up : Missing Information Error Message
6	Accounts cannot be opened for users under 16.	- First Name: John - Last Name: Doe - Date of Birth: 11/01/2015	Pop-Up : Invalid Date of Birth: 11/01/2015. Must be over 16.

		● Checking ● Balance: 5000	
7	College Checking cannot be opened for users over 24.	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● College Checking ● Balance: 5000	Pop-Up : Invalid Date of Birth: 11/01/2015. Must be under 24.
8	The date must be a valid date in order to open an account. If the date is not valid, then an invalid dob error message would appear.	● First Name: John ● Last Name: Doe ● Date of Birth: 13/01/1993 ● Checking ● Balance: 5000	Pop-Up : Invalid Date of Birth: 2/29/2011 not a valid calendar date!
9	Initial deposit cannot be zero. If the initial deposit is zero, then a invalid information pop-up must appear	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● Checking ● Balance: 0	Pop-Up : Invalid input. Balance cannot be 0 or negative.
10	Initial deposit cannot be negative. If the initial deposit is negative, then a invalid information pop-up must appear	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● Checking ● Balance: -5000	Pop-Up : Invalid input. Balance cannot be 0 or negative.
11	A campus needs to be selected in order to open a College Checking. If the user does not select a campus, then a missing information pop-up must appear	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● College Checking ● Balance: 5000	Pop-Up : Invalid input: Missing Campus Input
12	All the information is input correctly in their respective field. If the user has input all the information, then the account will be opened and a message would be displayed.	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● Checking ● Balance: 1000	John Doe 11/01/1993 (C) opened.
13	An already opened account cannot be opened again. If an account is	● First Name: John ● Last Name: Doe	John Doe 11/01/1993 (C) is

	already open, then a message would appear saying that account is already in the database	- Date of Birth: 11/01/1993 - Checking - Balance: 1000	already in the database.

Test Specification for the Close Button [on Open/Close tab]

Class name: TransactionManagerController
Method signature: closeButtonClicked(){} //closes the account entered if it exists in the database

Test Case #	Requirement	Test Description and Input Date	Expected Result / Output
1	First name is required to close an account. A missing information pop-up shall appear if the first name is not provided.	- First Name: - Last Name: Doe - Date of Birth: 11/01/2000 - Checking - Balance: 5000	Pop-Up : Missing Information Error Message
2	Last name is required to close an account. A missing information pop-up shall appear if the last name is not provided.	- First Name: John - Last Name: - Date of Birth: 11/01/2000 - Checking - Balance: 5000	Pop-Up : Missing Information Error Message
3	Date of birth is required to close an account. A missing information pop-up shall appear if the date of birth is not provided.	- First Name: John - Last Name: Doe - Date of Birth: - Checking - Balance: 5000	Pop-Up : Missing Information Error Message
4	Account type is required to close an account. A missing information pop-up shall appear if the account type is not provided.	- First Name: John - Last Name: Doe - Date of Birth: 11/01/2000 - Balance: 5000	Pop-Up : Missing Information Error Message
5	A campus needs to be selected in order to close a College Checking. If the user does not select a campus, then a missing information pop-up must appear	- First Name: John - Last Name: Doe - Date of Birth: 11/01/1993 - College Checking - Balance: 5000	Pop-Up : Invalid input: Missing Campus Input
6	Account needs to exist in order for you to close it. If the account does not exist in the database, then a	- First Name: John - Last Name: Doe - Date of Birth: 11/01/1993	John Doe 11/01/1993 (C) is not in the database

	message would be displayed stating that.	● Checking ● Balance: 5000	
7	If the account exists, then the manager would close the account and display an account closed message.	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● Checking ● Balance:	John Doe 11/01/1993 C closed.

Test Specification for the Deposit Button [on Deposit/Withdraw tab]

Class name: TransactionManagerController
Method signature: depositButtonClicked(){} //deposits the amount in the entered account

Test Case #	Requirement	Test Description and Input Date	Expected Result / Output
1	First name is required to deposit into an account. A missing information pop-up shall appear if the first name is not provided.	- First Name: - Last Name: Doe - Date of Birth: 11/01/2000 - Checking - Amount: 5000	Pop-Up : Missing Information Error Message
2	Last name is required to deposit into an account. A missing information pop-up shall appear if the last name is not provided.	- First Name: John - Last Name: - Date of Birth: 11/01/2000 - Checking - Amount: 5000	Pop-Up : Missing Information Error Message
3	Date of birth is required to deposit into an account. A missing information pop-up shall appear if the date of birth is not provided.	- First Name: John - Last Name: Doe - Date of Birth: - Checking - Amount: 5000	Pop-Up : Missing Information Error Message
4	Account type is required to deposit into an account. A missing information pop-up shall appear if the account type is not provided.	- First Name: John - Last Name: Doe - Date of Birth: 11/01/2000 - Amount: 5000	Pop-Up : Missing Information Error Message
5	A number format error occurs when the user tries to input non-numbers into the amount box. An invalid amount error pop-up will appear.	- First Name: John - Last Name: Doe - Date of Birth: 11/01/1993 - College Checking - Balance: Deer	Pop-Up : Invalid input: Not a valid amount
6	Deposit cannot be negative. If the deposit is negative or 0, then a invalid information pop-up must appear	- First Name: John - Last Name: Doe - Date of Birth: 11/01/1993 - Checking	Pop-Up : Invalid input: amount cannot be 0 or negative.

		● Balance: -50	
7	Deposit cannot be zero. If the deposit is negative or 0, then a invalid information pop-up must appear	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● Checking ● Balance: 0	Pop-Up : Invalid input: amount cannot be 0 or negative.
8	Money would be deposited into the account if it is a valid existing account.	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● Checking ● Balance: 100	John Doe 11/01/1993 (C) Deposit - balance updated.

Test Specification for the Withdraw Button [on Deposit/Withdraw tab]

Class name: TransactionManagerController
Method signature: withdrawButtonClicked(){} //withdraws the amount from the entered account

Test Case #	Requirement	Test Description and Input Date	Expected Result / Output
1	First name is required to withdraw into an account. A missing information pop-up shall appear if the first name is not provided.	● First Name: ● Last Name: Doe ● Date of Birth: 11/01/2000 ● Checking ● Amount: 5000	Pop-Up : Missing Information Error Message
2	Last name is required to withdraw into an account. A missing information pop-up shall appear if the last name is not provided.	● First Name: John ● Last Name: ● Date of Birth: 11/01/2000 ● Checking ● Amount: 5000	Pop-Up : Missing Information Error Message
3	Date of birth is required to withdraw into an account. A missing information pop-up shall appear if the date of birth is not provided.	● First Name: John ● Last Name: Doe ● Date of Birth: ● Checking ● Amount: 5000	Pop-Up : Missing Information Error Message
4	Account type is required to withdraw into an account. A missing information pop-up shall appear if the account type is not provided.	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/2000 ● Amount: 5000	Pop-Up : Missing Information Error Message
5	A number format error occurs when the user tries to input non-numbers into the amount box. An invalid amount error pop-up will appear.	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● College Checking ● Balance: Deer	Pop-Up : Invalid input: Not a valid amount
6	Withdraw cannot be negative. If the withdraw is negative or 0, then	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993	Pop-Up : Invalid input: amount cannot be 0 or negative.

	a invalid information pop-up must appear	● Checking ● Balance: -50	
7	Withdraw cannot be zero. If the withdraw is negative or 0, then a invalid information pop-up must appear	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● Checking ● Balance: 0	Pop-Up : Invalid input: amount cannot be 0 or negative.
8	There needs to be enough money in the account in order to withdraw the given amount. If there are insufficient funds, then a message stating such should be displayed.	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● Checking ● Balance: 100000000	John Doe 11/01/1993 (C) Withdraw - insufficient fund.
9	If there are sufficient funds in a valid account, then the amount can be withdrawn	● First Name: John ● Last Name: Doe ● Date of Birth: 11/01/1993 ● Checking ● Balance: 10	John Doe 11/01/1993 (C) Withdraw - balance updated.

Test Specification for the Print All Accounts Button [on Account Database tab]

Class name: TransactionManagerController			
Method signature: printAllButtonClicked(){} //prints all the accounts that are currently in the database sorted			
Test Case #	Requirement	Test Description and Input Date	Expected Result / Output
1	If the database is empty, then it would print a message stating such.	● Click "Print All Accounts" button with an empty database	Account Database is empty!
2	If the database base contains any number of accounts, then it would print all the accounts with their corresponding information sorted by account type and name.	● Click "Print All Accounts" button with a non-empty database	*Accounts sorted by account type and profile. College Checking::Jane Doe 10/1/2000::Balance $999.99::NEW_BRUNSWICK Savings::Jane Doe 10/1/1995::Balance $1000.00 Savings::John Doe 2/19/1990::Balance $200.00 * end of list.

Test Specification for the Print Interest and Fees Button [on Account Database tab]

Class name: TransactionManagerController			
Method signature: printInterestFeesButtonClicked(){} //prints all the accounts in the database with their corresponding interests and fees sorted by type and name			
Test Case #	Requirement	Test Description and Input Date	Expected Result / Output
1	If the database is empty, then it would print a message stating such.	● Click "Print Interest and Fees" button with an empty database	Account Database is empty!
2	If the database base contains any number of accounts, then it would print all the accounts with their corresponding interests and fees sorted by account type and name	● Click "Print Interest and Fees" button with a non-empty database	*list of accounts with fee and monthly interest College Checking::Jane Doe 10/1/2000::Balance $999.99::NEW_BRUNSWICK::fee $0.00::monthly interest $0.83 Savings::Jane Doe 10/1/1995::Balance $1000.00::fee $0.00::monthly interest $3.33 Savings::John Doe 2/19/1990::Balance $200.00::fee $25.00::monthly interest $0.67

Test Specification for the Load Accounts from a File [on Account Database tab]

Class name: TransactionManagerController
Method signature: loadAccountsButtonClicked(){} //Load accounts from a .txt file and starts to read it using BufferReader

Test Case #	Requirement	Test Description and Input Date	Expected Result / Output
1	If a correct txt file is selected and has the correctly formatted inputs to the program, such as that from the given bankAccounts.txt, then the program continues regularly as expected (adding, closing, depositing, withdrawing, etc.)	● Click "Load Accounts from a file" button and choose a correctly formatted input txt file, such as the given bankAccounts.txt	John Doe 2/19/2000 (CC) opened. Jane Doe 10/1/2000 (CC) opened. Jane Doe 10/1/1995 (S) opened. Jane Doe 10/1/1995 (MM) opened.
2	If the file selected is not found, then it outputs a message and the program would continue to run.	● Click "Load Accounts from a file" button and choose a file that	FILE NOT FOUND.
3	If no file is selected (close out of file selection), the text area displays "NO FILE SELECTED"	● Click "Load Accounts from a file" button and close out of the file selection window (no file selected)	NO FILE SELECTED
4	If a file of a different type than requested (non-txt type) is selected, the text area displays "ERROR: TRY RELOADING THE FILE OR ENTERING DATA MANUALLY."	● Click "Load Accounts from a file" button and choose	ERROR: TRY RELOADING THE FILE OR ENTERING DATA MANUALLY.