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 Data	Expected result/output
1	The method shall return false for any date with a year before 1900.	• Create an instance of Date with valid day and month but with a year < 1900. • test data: "2/29/1899"	false
2	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 the month = 2, day > 28, and the year is a non-leap year • test data: "2/29/2011"	false
3	Valid month shall be >= 1. The method shall return false for a month value < 1.	• Create an instance of Date with the month < 1 • test data: "0/29/2012"	false
4	The day of a date should be >= 1 and less than another constant depending on the month and if the year is a leap year.	• Create an instance of Date with the day < 1 • test data: "11/0/2012"	false
5	Valid month shall be <= 12. The method shall return false for a month value > 12.	• Create an instance of Date with the month > 12 • test data: "13/29/2012"	false
6	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 the month = 2, day = 29, and the year is a leap year • test data: "2/29/2012"	true
7	Number of days in January shall be between 1 and 31 inclusive.	• Create an instance of Date with the month 1, day [1,31], valid year such as 2012 • test data: "1/3/2012"	true

class name: AccountManager
method signature: public boolean close() {} //returns true if found and closed account, otherwise

false.			
Test Case #	**Requirement**	**Test description and Input Data**	**Expected result/output**
1	The method shall return true if the event is found in AccountDatabase and is removed.	• Create two same instances of Account objects. Pass into AccountDatabase and close the same account. • test data: AccountDatabase(accounts[], 1) close(the same account)	true
2	The method shall return false if the event is not found in AccountDatabase and subsequently not removed.	• Create two different instances of Account objects. Pass into AccountDatabase and try to close the different Account. • test data: AccountDatabase(accounts[], 1) close(different Account)	false
3	GUI should not allow empty initial deposit.	• Enter in all other parameters to create an account on the GUI but don't enter in initial deposit. • test data: first name, last name, D.O.B, Account Type, Initial Deposit EMPTY	Not possible
4	Should not accept initial deposit of value <= 0	• Enter in all other parameters to create an account on the GUI but enter value <= 0 as initial deposit. • test data: first name, last name, D.O.B, Account Type, Initial Deposit <= 0	Initial deposit cannot be 0.
5	GUI should not allow empty/missing parameters	• Enter in all other parameters to create an account on the GUI but don't enter one parameter such as first name. • test data: last name, D.O.B, Account Type, Initial Deposit = 1000	Not possible

6	Should print the correct list of accounts ordered by account type and profile when selecting “Account Type Order” GUI button.	• Create multiple accounts using the GUI and enter valid parameters for each. • test data: Multiple accounts entered through GUI, should print as expected.	Should print the correct order.
7	Cannot withdraw if account has insufficient balance	• Create account using the GUI and enter valid parameters, then enter same parameters but put greater value in Transaction amount parameter on GUI and click withdraw • test data: first name, last name, D.O.B, Account Type, Initial Deposit 1000; first name, last name, D.O.B, Account Type, Transaction Amount 1001	Withdraw - insufficient fund.
8	Should not be able to type invalid/unnecessary characters into GUI input boxes. In this case should, not be able to enter anything other than alphanumeric characters.	• Enter non-alphanumeric characters into the input boxes • test data: |}”:<>?<>?*&^%$#	Not possible
9	Should not be able to enter numbers into first and last name GUI fields	• Enter numbers into first or last name GUI fields • test data: first name: 89713	Not possible
10	Should not be able to enter alphabetical characters into Initial Deposit or Transaction Amount fields.	• Enter alphabetical characters into the Transaction Amount field. • test data: qwsdaefsrafesaet	Not possible
11	Typing in display boxes should not affect anything else in GUI or in process. GUI should remain in sane state.	• Enter random characters into any display box in the GUI. • test data: aes8fya9y2hkajds|})(*&^*!%~	Not effect