|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case#** | **Purpose of the Test Case** | **Input Data** | **Expected Output** |
| 1 | Test the GUI for mismatches in inputs for open accounts button, and tests open accounts for functionality and repeated adds.  **• Case 1:** press open button with empty text boxes  **• Case 2:** press open button with invalid first name filled and rest empty  • **Case 3:** press open button with valid first name filled, and rest empty  **• Case 4**: press open button with valid first name filled, an invalid second name, and rest empty.  **• Case 5:** press open button with valid first name filled, and a valid last name, and rest empty.  **• Case 6:** press open button with valid first name filled, valid last name filled, invalid number format date, and rest empty.  **• Case 7:** press open button with valid first name filled, valid last name filled, invalid date, and rest empty.  **• Case 8**: press open button with valid first name filled, valid last name filled, valid date, and rest empty.  **• Case 9:** press open button with valid first name filled, valid last name filled, valid date, and invalid double.  **• Case 10:** press open button with valid first name filled, valid last name filled, valid date filled, valid double filled, and rest unclicked (radio buttons).  **• Case 11**: press open button with valid first name filled, valid last name filled, valid date filled, valid double filled, checking clicked, and direct deposit unclicked.  **• Case 12:** press open button with valid first name filled, valid last name filled, valid date filled, valid double filled, checking clicked and direct deposit clicked.  **• Case 13**: press open button with valid first name filled, valid last name filled, valid date filled, valid double filled, saving clicked and loyal customer not clicked.  **• Case 14:** press open button with valid first name filled, valid last name filled, valid date filled, valid double filled, Saving clicked, and loyal customer clicked.  **• Case 15**: press open button with valid first name filled, valid last name filled, valid date filled, valid double filled, and money market clicked  **• Case 16:** press open button with valid first name filled, valid last name filled, valid date filled, valid double filled, checking clicked, and direct deposit unclicked.  **• Case 17:** press open button with valid first name filled, valid last name filled, valid date filled, valid double filled, Saving clicked, and loyal customer  **• Case 18**: press open button with valid first name filled, valid last name filled, valid date filled, valid double filled, and money market clicked | **• Case 1:** no inputs  **• Case 2:** first Name: Julian!1  **• Case 3**: first Name: Julian  **• Case 4:** first Name: Julian, last name: Romero!2  **• Case 5:** first Name: Julian, last name: Romero  **• Case 6:** first Name: Julian, last name: Romero, Date: 12!/21/2003  **• Case 7:** first Name: Julian, last name: Romero, Date: 2/29/2021  **• Case 8:** first Name: Julian, last name: Romero, Date: 2/28/2021  **• Case 9:** first Name: Julian, last name: Romero, Date: 2/28/2021, 21!  **• Case 10:** first Name: Julian, last name: Romero, Date: 2/28/2021, 21, radio buttons not pressed  **• Case 11:** first Name: Julian, last name: Romero, Date: 2/28/2021, 21, checking clicked, and direct deposit not clicked  **• Case 12:** first Name: Jack, last name: Romero, Date: 2/28/2021, 21, checking clicked, and direct deposit clicked  **• Case 13:** first Name: Julian, last name: Romero, Date: 2/28/2021, 21, Saving clicked, and loyal customer not clicked  **• Case 14**: first Name: Jack, last name: Romero, Date: 2/28/2021, 21, Saving clicked, and loyal customer clicked  **• Case 15**: first Name: Julian, last name: Romero, Date: 2/28/2021, 21, money market clicked  **• Case 16:** first Name: Julian, last name: Romero, Date: 2/28/2021, 21, checking clicked, and direct deposit not clicked  **• Case 18**: first Name: Julian, last name: Romero, Date 2/28/2021, 21, money market clicked | **• Case 1:** “First name must be entered!”  **• Case 2:** “First name cannot have numbers or symbols!”  **• Case 3**: “last name must be entered!”  **• Case 4:** “Last name cannot have numbers or symbols!”  **• Case 5:** “Date must be entered and cannot have letters or symbols!”  **• Case 6:** “Date must be entered and cannot have letters or symbols!”  **• Case 7:** “Not a valid Date!”  **• Case 8:** “Amount must be entered and be a double!”  **• Case 9:** “Amount must be entered and be a double!”  **• Case 10:** “An account type must be selected!”  **• Case 11:** “Checking Account, Julian Romero, has been added.”  **• Case 12**: “Checking Account, Jack Romero, has been added.”  **• Case 13:** “Savings Account, Julian Romero, has been added.”  **• Case 14:** “Savings Account, Jack Romero, has been added.”  **• Case 15:** “MoneyMarket Account, Julian Romero, has been added.”  **• Case 16:** “Checking Account has not been added: Julian Romero.”  **• Case 17:** “Savings Account has not been added: Jack Romero.”  **• Case 18:** “Money Market Account has not been added: Julian Romero.” |
| 2 | Test the GUI for mismatches in inputs for close account button, and tests close accounts for functionality and repeated adds.  **• Case 1:** press close button with empty text boxes  **• Case 2:** press close button with invalid first name filled and rest empty  • **Case 3:** press close button with valid first name filled, and rest empty  **• Case 4**: press close button with valid first name filled, an invalid second name, and rest empty.  **• Case 5:** press close button with valid first name filled, and a valid last name, and rest empty.  **• Case 6:** press close button with valid first name filled, and a valid last name, and checking clicked.  **• Case 7:** press close button with valid first name filled, and a valid last name, and saving clicked.  **• Case 8:** press close button with valid first name filled, and a valid last name, and money market clicked.  **• Case 9:** press close button with valid first name filled, and a valid last name, and checking clicked.  **• Case 10:** press close button with valid first name filled, and a valid last name, and saving clicked.  **• Case 11:** press close button with valid first name filled, and a valid last name, and money market clicked. | **• Case 1:** no inputs  **• Case 2:** first Name: Julian!1  **• Case 3**: first Name: Julian  **• Case 4:** first Name: Julian, last name: Romero!2  **• Case 5:** first Name: Julian, last name: Romero  **• Case 6:** first Name: Julian, last name: Romero, checking clicked  **• Case 7:** first Name: Julian, last name: Romero, saving clicked  **• Case 8:** first Name: Julian, last name: Romero, money market clicked  **• Case 9:** first Name: Julian, last name: Romero, checking clicked  **• Case 10:** first Name: Julian, last name: Romero, saving clicked  **• Case 11:** first Name: Julian, last name: Romero, money market clicked | **• Case 1:** “First name must be entered!”  **• Case 2:** “First name cannot have numbers or symbols!”  **• Case 3**: “last name must be entered!”  **• Case 4:** “Last name cannot have numbers or symbols!”  **• Case 5:** “An account type must be selected!”  **• Case 6:** “Checking Account, Julian Romero, has been closed.”  **• Case 7:** “Saving Account, Julian Romero, has been closed.”  **• Case 8:** “Money Market account, Julian Romero, has been closed.”  **• Case 9:** “Checking Account has not been closed: Julian Romero.”  **• Case 10:** “Savings Account has not been closed: Julian Romero.”  **• Case 11:** “Money market account Account has not been closed: Julian Romero.” |
| 3 | Tests the import method  **• Case 1:** import a file not in the format specified for importing  **• Case 2:** import a file in the format specified for importing but one input for the accounts has a wrong input format  **• Case 3:** import a file in the correct format specified for importing  **• Case 4:** import a file in the correct format specified for importing | **• Case 1:** wrong formated text file  **• Case 2:** database.txt but first account has “!” in first name  • **Case 3:** database.txt  • **Case 4:** database.txt | **• Case 1:**  “wrong input size format”  **• Case 2:**  “wrong input name format”  **• Case 3:**  “Import completed”  **• Case 4:**  “Warning some accounts already in database and have not been added”  “Import completed “ |
| 4 | Tests the Export method  **• Case 1:** Export a file to the desktop when database is empty  **• Case 2:** Export a file to the desktop when database is not empty | **• Case 1:** Empty database  **• Case 2:** import database.txt as starting database | **• Case 1:** “Export Failed database empty”  **• Case 2:** “Export Successful” |
| 5 | Tests the Withdraw method  **• Case 1:** Check if withdraw works with no inputted data  **• Case 2:** Check if withdraw works with a missing first name  **• Case 3:** Check if withdraw works with a missing last name  **• Case 4:** Check if withdraw works with a missing amount  **• Case 5:** Check if withdraw works with a missing account type  **• Case 6:** Check if withdraw works with an account that does not exist  **• Case 7:** Check if withdraw works with a checking account that exists with enough money  **• Case 8:** Check if withdraw works with a savings account that exists and if withdraw confuses account types  **• Case 9:** Check if withdraw works with a negative amount  **• Case 10:** Check if withdraw works with a checking account that exists with not enough money | Note: DEFAULT will be first name = “a”, last name = “b”, amount = 10, account type = Checking  **• Case 1:** no input  **• Case 2:** DEFAULT, but first name = “”  **• Case 3:** DEFAULT, but last name = “”  **• Case 4:** DEFAULT, but amount = “”  **• Case 5:** DEFAULT, but account type = null  **• Case 6:** DEFAULT  **• Case 7:** DEFAULT, but after creating a checking account with the DEFAULT data and 100 money  **• Case 8:** DEFAULT, but after creating a checking and savings account with the DEFAULT data  **• Case 9:** DEFAULT, but amount = -10  **• Case 10:** DEFAULT, but after creating a checking account with the DEFAULT data and 0 money | **• Case 1:** invalid names  **• Case 2:** invalid names  **• Case 3:** invalid names  **• Case 4:** invalid amount  **• Case 5:** invalid account type  **• Case 6:** account not found  **• Case 7:** success  **• Case 8:** success  **• Case 9:** invalid amount  **• Case 10:** not enough balance |
| 6 | Tests the Deposit method  **• Case 1:** Check if deposit works with no inputted data  **• Case 2:** Check if deposit works with a missing first name  **• Case 3:** Check if deposit works with a missing last name  **• Case 4:** Check if deposit works with a missing amount  **• Case 5:** Check if deposit works with a missing account type  **• Case 6:** Check if deposit works with an account that does not exist  **• Case 7:** Check if deposit works with a checking account that exists  **• Case 8:** Check if deposit works with a savings account that exists and if deposit confuses account types  **• Case 9:** Check if deposit works with a negative amount | Note: DEFAULT will be first name = “a”, last name = “b”, amount = 10, account type = Checking  **• Case 1:** no input  **• Case 2:** DEFAULT, but first name = “”  **• Case 3:** DEFAULT, but last name = “”  **• Case 4:** DEFAULT, but amount = “”  **• Case 5:** DEFAULT, but account type = null  **• Case 6:** DEFAULT  **• Case 7:** DEFAULT, but after creating a checking account with the DEFAULT data and 100 money  **• Case 8:** DEFAULT, but after creating a checking and savings account with the DEFAULT data  **• Case 9:** DEFAULT, but amount = -10 | **• Case 1:** invalid names  **• Case 2:** invalid names  **• Case 3:** invalid names  **• Case 4:** invalid amount  **• Case 5:** invalid account type  **• Case 6:** account not found  **• Case 7:** success  **• Case 8:** success  **• Case 9:** invalid amount |
| 7 | Tests the Print Account method  **• Case 1:** Check on empty database  **• Case 2:** Check on 1 account database and check direct deposit  **• Case 3:** Check on database of 6 and check if not having direct deposit displays, and check if having loyal displays, and check if not having loyal displays, and check if withdrawals displays | Note: Accounts will be inputted with the name “fC lC” with C representing the Nth letter of the alphabet of the Nth account and date 1/2/2000 and balance N \* 100  **• Case 1:** no input  **• Case 2:** create a checking account with direct deposit  **• Case 3:** in addition to Case 2, create a checking account without direct deposit, create a savings without loyal, create a savings with loyal, create a checking without direct deposit, withdraw 10 from the money market account | **• Case 1:** database is empty  **• Case 2:** the data for the one account  **• Case 3:** the data for all the classes including the account type, the 100 or 90 balance (for the one money market), the number of withdrawals for the money market, the special attribute on checking and savings accounts |
| 7 | Tests the Print by Dates method  **• Case 1:** Check if date of Dates is sorted  **• Case 2:** Check if month of Dates is sorted  **• Case 3:** Check if year of Dates is sorted  **• Case 4:** Check if name of accounts is sorted when the date is the same | Note: Accounts will be inputted as a checking with the name “C C” with C representing the Nth letter of the alphabet of the Nth account and balance N \* 100  **• Case 1:** create accounts with date 1/1/2000, date 1/3/2000, and date 1/2/2000  **• Case 2:** create accounts with date 1/1/2000, date 3/1/2000, and date 2/1/2000  **• Case 3:** create accounts with date 1/1/2000, date 1/1/2002, and date 1/1/2001  **• Case 4:** create accounts with date 1/1/2000, date 1/1/2000, and date 1/1/2000 | The result for all cases is expected to be the first account made, the third account made, then the second account made |
| 8 | Tests the Print by Name method  **• Case 1:** Check if first name of names is sorted  **• Case 2:** Check if last name of names is sorted | Note: Accounts will be inputted as a checking with the name “C C” with C representing the Nth letter of the alphabet of the Nth account and balance N \* 100  **• Case 1:** create accounts with names: “a a”, “c a”, and “b a”  **• Case 2:** create accounts with names: “a a”, “a c”, and “a b” | The result for all cases is expected to be the first account made, the third account made, then the second account made |