|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Purpose of Test Case | Input Data | Expected Output |
| 1 | Leave all text fields empty and check buttons unchecked and click Add Account or close Account buttons to see if the GUI catches the errors | First Name:  Last Name:  DateOpen:  Balance:  No radio buttons selected | Warning window pop ups:   * First Name cannot be left null * Last Name cannot be left null * Date cannot be left null * Non-Numeric data entered * Please select the type of account you want to open. |
| 2 | Click Add account with all valid text fields and radio buttons selected | Case 1: Account not in database  First Name: Jane  Last Name: Doe  Checking radio button: selected  Case 2: Account already in database  First Name: Jane  Last Name: Doe  Checking radio button: selected | Case 1:  Account added to the database  Case 2:  Account is already in database |
| 3 | Imports .txt file with 6 different accounts to database testing add() and grow() and equals() and the import functionality | Import database.txt ...  S, Jane, Doe, 500.0,11/1/2019, false  S, John, Doe, 500.0, 1/1/2010, true  C, Jane, Doe, 500.0, 2/29/2008, false  C, Richard, Scanlan, 1500.0, 7/31/2014, false  C, Jason, Brown, 300.12, 6/28/2015, false  S, Mary, Johnson, 2000.88, 7/31/2014, true | Account opened and added to the database.  Account opened and added to the database.  Account opened and added to the database.  Account opened and added to the database.  Account opened and added to the database.  Account opened and added to the database. |
| 4 | Close Account button clicked with all proper text fields filled out. Testing remove() | Case 1: Account found  First Name: Jane  Last Name: Doe  Checking radio button: selected  Case 2: account not found  First Name: Jane  Last Name: Smith  Checking radio button: selected | Case 1:  Account closed and removed from the database.  Case 2:  Account does not exist. |
| 5 | Click deposit or withdraw buttons with invalid/empty text fields to see if GUI catches errors. | First Name:  Last Name:  Amount: | Warning window pop ups:   * Non-Numeric data has been entered * First name cannot be left null * Last Name cannot be left null. |
| 6 | Click deposit button w valid text fields. Testing deposit() and find() methods | Case 1: Account found  First Name: Richard  Last Name: Scanlan  Amount: 300.45  Checking: selected  Case 2: Account not found  First Name: Richard  Last Name: Scanlan  Amount: 300.45  Checking: selected | Case 1:  $300.45 deposited to account  Case 2:  Account does not exist. |
| 7 | Export account database into a .txt file | Exports a .txt file of account entries | Saves a .txt file in your computer. The data is the string returned by printAccounts method which lists all the accounts in the bank database. |
| 8 | Click withdraw button w valid text fields. Testing withdrawal() method. | Case 1: Insufficient funds  First name: Jane  Last name: Doe  Checkings button selected  Amount: 1000.00  Case 2: Withdrawal successful  First name: Jane  Last name: Doe  Checkings button selected  Amount: 100.00  Case 3: Account Does not exist  First name: John  Last name: Doe  Checkings button selected  Amount: 100.00 | Case 1:  Insufficient funds.  Case 2:  100.00 withdrawn from account.  Case 3:  Account does not exist. |
| 9 | Printing accounts in database testing all print methods and sort methods. | Case 1: printByDateOpen  Print menu -> print by date open  Case 2: PrintByLastName  Print menu -> print by last name  Case 3: printAccounts  Print Menu - > print Accounts in database | Case 1:  --Printing statements by date opened--  \*Checking\*Jane Doe\* $500.00\*2/29/2008  -interest: $ 0.02  -fee: $ 25.00  -new balance: $ 475.02  \*Savings\*John Doe\* $500.00\*1/1/2010\*special Savings account\*  -interest: $ 0.15  -fee: $ 0.00  -new balance: $ 500.15  --end of printing--  Case 2:  --Printing statements by last name--  \*Checking\*Jerry Anderson\* $976.44\*9/23/2020  -interest: $ 0.04  -fee: $ 25.00  -new balance: $ 951.48  \*Money Market\*Charlie Brown\* $588.66\*6/2/2015\*6 withdrawals\*  -interest: $ 0.32  -fee: $ 12.00  -new balance: $ 576.97  --end of printing--  Case 3:  -Listing accounts in the database--  \*Checking\*Jerry Anderson\* $926.52\*9/23/2020  \*Money Market\*Charlie Brown\* $565.29\*6/2/2015\*6 withdrawals\*  --end of printing-- |
| 10 | Testing date picker | Case 1: Select the date from date picker: valid  Case 2: Try to enter wrong date text in the date picker: invalid | Case 1: Date will be selected because it is valid.  Case 2: Any invalid date, such as any text instead of a date, will be considered null.  NOTE: But if you select a date once, and then manually clear the text in the date picker, it is not considered null. Because the text gets cleared but the date you selected still stays selected. |