Amy Wang, Junhao Shen Project 3 Test Document

|  |  |  |  |
| --- | --- | --- | --- |
| Test # | Purpose | Input Data | Expected output |
| 1 | Test all possible empty / errors to see if program can catch them and not crash  Open and Close Account Tab:   * Case 1: all empty, returns error statement * Case 2/3: balance left empty and balance not a number, returns error statement * Case 4/5: date left empty and date not an integer, returns error statement * Case 6: invalid date entered, returns error statement * Case 7: type of account left empty, returns error statement | * Case 1:   All fields left empty   * Case 2/3:   First name = John  Last name = Doe  balance left empty and used “abc” as input for balance   * Case 4/5:   First name = John  Last name = Doe  Balance = 100  date left empty and used “m”, “d”, “y” as input for date   * Case 6:   First name = John  Last name = Doe  Balance = 100  Date = 21/21/2121   * Case 7:   First name = John  Last name = Doe  Balance = 100  Date = 12/12/2020 | * Case 1:   Please enter a first name and last name!   * Case 2/3:   Given balance is not properly formatted!   * Case 4/5:   Given date is not properly formatted!   * Case 6:   21/21/2121 is not a valid date!   * Case 7:   Please select account type! |
| 2 | Test empty / errors to see if program can catch them and not crash  Withdraw / Deposit Tab:   * Case 8: all empty, returns error statement * Case 9/10: amount left empty and amount not a number, returns error statement * Case 11: type of account left empty, returns error statement | * Case 8:   All fields left empty   * Case 9/10:   First name = John  Last name = Doe  amount left empty and used “abc” as input for amount   * Case 11:   First name = John  Last name = Doe  Balance = 100  Date = 12/12/2020 | Case 8:  Please enter a first name and last name!   * Case 9/10:   Given amount is not properly formatted!   * Case 11:   Please select account type! |
| 3 | Test the open account button / method to see if successfully created account   * Case 1: Create a checking account, should not return error statement * Case 2: Try to create an account already in the database, should return error statement | * Case 1:   First name = John  Last name = Doe  Balance = 100  Date = 12/12/2020  Select checking   * Case 2:   First name = John  Last name = Doe  Balance = 100  Date = 12/12/2020  Select checking | * Case 1:   Account opened and added to the database.   * Case 2:   Account is already in the database. |
| 4 | Test the close account button / method to see if successfully closed account   * Case 1: Remove account that isn’t in database, should return error statement * Case 2: Remove account from the database, should not return error statement | * Case 1:   First name = A  Last name = B  Select checking   * Case 2:   First name = Bob  Last name = Bobby  Select savings | * Case 1:   Account does not exist.   * Case 2:   Account closed and removed from the database. |
| 5 | Test the clear button  Should clear fields and reset to default state | Enter random input in text fields, and select radio button  Click clear button | Fields and radio buttons are empty, direct deposit and loyal customer checkboxes are disabled |
| 6 | The the deposit button / method to see if successfully deposits money into an account   * Case 1: Deposit into account that isn’t in database, should return error statement * Case 2: Deposit into account from the database, should not return error statement | * Case 1:   Select checking  First name = A  Last name = B  Amount = 100   * Case 2:   Select checking  First name = John  Last name = Doe  Amount = 12 | * Case 1:   Account does not exist.   * Case 2:   $12.00 deposited to account. |
| 7 | The the withdraw button / method to see if successfully withdraws money from an account   * Case 1: Withdraw from account that isn’t in database, should return error statement * Case 2: Withdraw from account from the database, no error statement * Case 3: Withdraw more than the balance of existing account, should return error statement | * Case 1:   Select checking  First name = A  Last name = B  Amount = 100   * Case 2:   Select checking  First name = John  Last name = Doe  Amount = 10   * Case 3:   Select checking  First name = John  Last name = Doe  Amount = 10000 | * Case 1:   Account does not exist.   * Case 2:   $10.00 withdrawn from account.   * Case 3:   Insufficient funds. |
| 8 | Test the printing options to see if they work as intended   * Case 1: Database is empty, should state that for all three * Case 2: database not empty, print account statements * Case 3: database not empty, print by date opened * Case 4: database not empty, print by last name | * Case 1:   Empty database   * Case 2   Database contains 3 accounts, fees and interest should not be calculated   * Case 3/4   Database contains 3 accounts, fees and interest should be properly calculated, new balance should also be calculated | * Case 1:   Database is empty.   * Case 2:   (proper output is displayed in textArea - not included in test document)   * Case 3/4:   (proper output is displayed in textArea - not included in test document) |
| 9 | Test the import button / method to see if successfully imported the file and added accounts to the existing database.  Tested find to ignore already existing accounts and see if program can ignore it and not crash | Accounts in database,  Import: database.txt  (imported database has account already in database) | Successfully imported database, duplicates were ignored.  (when selected print account statements, received database with imported accounts) |
| 10 | Test the export button / method to see if successfully exported database and wrote account info to text file | Many accounts in database,  Export:  accountDatabase | Successfully exported database.  (under same parent folder that contains src folder, exportedDatabase.txt is created with proper format) |