

Class name: TransactionManagerController

Method signature: private void handleOpenButton(ActionEvent event) {}
// Open an account with user input

Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method returns a string stating an error if the account to be opened does not have all the required fields (first name, last name, date of birth, account type)	<ul style="list-style-type: none">• Open an account without all the required fields filled in• Input data: First name of Jane, date of birth of 03/12/2001, savings account	"Error: Fill in all required fields (First name, last name) and input a valid date of birth"
2	This method returns a string if the account to be opened is already in the account database.	<ul style="list-style-type: none">• Open a checking account and open another checking account with the same input• Input data: First name of Jane, last name of Doe, date of birth of 03/12/2001, checking account	"Jane Doe 3/12/2001(C) is already in the database."
3	This method returns a string if the account to be opened does not have a valid date of birth	<ul style="list-style-type: none">• Open a savings account with an invalid date of birth• Input data: First name of Jane, last name of Doe, date of birth of 29/10/2001, savings account	"Error: Fill in all required fields (First name, last name) and input a valid date of birth"
4	This method returns a string if the account is opened successfully	<ul style="list-style-type: none">• Open a savings account with given input• Input data: First name of John, last name of Young, date of birth of 02/10/2000, savings account	"John Young 2/10/2000(S) opened."
5	This method returns a string stating an error if the account to be opened is a College Checking account but did not have a campus selected	<ul style="list-style-type: none">• Open a college checking account without selecting a campus• Input data: First name of James, last name of Brooks, date of birth of 05/22/2002, college checking account	"Error: Please select a campus for College Checking account."
6	This method returns a string if the account to be opened is a College Checking account but the user is under 16 years old	<ul style="list-style-type: none">• Open a college checking account where the user's age is below 16• Input data: First name of James, last name of Brooks, date of birth of 05/22/2011, college checking account, Newark campus	"DOB invalid: user under 16."

7	This method returns a string if the account to be opened is a College Checking account but the user is over 24 years old	<ul style="list-style-type: none"> Open a college checking account where the user's age is over 24 Input data: First name of James, last name of Brooks, date of birth of 05/22/1993, college checking account, Newark campus 	"DOB invalid: user over 24."
8	This method returns a string if the college checking account is opened successfully	<ul style="list-style-type: none"> Open a college checking account Input data: First name of James, last name of Brooks, date of birth of 05/22/2003, college checking account, Newark campus 	"James Brooks 5/22/2003(C C) opened."

Class name: TransactionManagerController Method signature: private void handleCloseButton(ActionEvent event) {} // Close an account with user input			
Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method returns a string stating an error if the account to be closed does not have all the required fields (first name, last name, date of birth, account type)	<ul style="list-style-type: none"> Close an account without all the required fields filled in Input data: First name of John, date of birth of 05/11/2001, savings account 	"Error: Fill in all required fields (First name, last name, and date of birth)"
2	This method returns a string if the account to be closed is not in the account database.	<ul style="list-style-type: none"> Close a checking account with the given input Input data: First name of John, last name of Doe, date of birth of 05/11/2001, checking account 	"John Doe 5/11/2001(C) is not in the database."
3	This method returns a string if the account to be closed does not have a valid date of birth	<ul style="list-style-type: none"> Close a savings account with an invalid date of birth Input data: First name of John, last name of Doe, date of birth of 29/11/2001, savings account 	""
4	This method returns a string if the account is closed successfully	<ul style="list-style-type: none"> Open a savings account with the given input and close the savings account with the same input Input data: First name of John, last name of Young, date of birth of 02/10/2000, savings account 	"John Young 2/10/2000(S) has been closed."

Class name: TransactionManagerController Method signature: private void handleCloseButton(ActionEvent event) {} // Close an account with user input			
Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method returns a string stating an error if the account to be closed does not have all the required fields (first name, last name, date of birth, account type)	<ul style="list-style-type: none"> Close an account without all the required fields filled in Input data: First name of John, date of birth of 05/11/2001, savings account 	"Error: Fill in all required fields (First name, last name, and date of birth)"
2	This method returns a string if the account to be closed is not in the account database.	<ul style="list-style-type: none"> Close a checking account with the given input Input data: First name of John, last name of Doe, date of birth of 05/11/2001, checking account 	"John Doe 5/11/2001(C) is not in the database."
3	This method returns a string if the account to be closed does not have a valid date of birth	<ul style="list-style-type: none"> Close a savings account with an invalid date of birth Input data: First name of John, last name of Doe, date of birth of 29/11/2001, savings account 	""
4	This method returns a string if the account is closed successfully	<ul style="list-style-type: none"> Open a savings account with the given input and close the savings account with the same input Input data: First name of John, last name of Young, date of birth of 02/10/2000, savings account 	"John Young 2/10/2000(S) has been closed."

Class name: TransactionManagerController Method signature: private void handleDepositWithdrawButton(ActionEvent event){} // Deposits/Withdraw money into/from an account			
Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method returns a string if user is trying to deposit/withdraw from an account that does not exist	<ul style="list-style-type: none"> Deposit/withdraw into/from an account that is not in the account database Input data: First name of John, last name of Young, date of birth of 02/10/2000, savings account 	"John Young 2/10/2000(S) is not in the database."

2	This method returns a string if user has successfully deposited	<ul style="list-style-type: none"> ● Open an account and deposit into the same account that is in the account database ● Input data: First name of John, last name of Young, date of birth of 02/10/2000, savings account, amount 200 	"John Young 2/10/2000(S) Deposit - balance updated."
3	This method returns a string if user is trying to withdraw a larger amount than what's in the account	<ul style="list-style-type: none"> ● Open an account, deposit into that same account an amount and withdraw from the same account an amount that is larger than the amount in the account ● Input data: First name of John, last name of Young, date of birth of 02/10/2000, savings account, deposited amount of 200, withdrawn amount of 300 	"John Young 2/10/2000(S) Withdraw - insufficient fund."
4	This method returns a string if user has successfully withdrawn	<ul style="list-style-type: none"> ● Open an account, deposit into that same account an amount and withdraw from the same account an amount that is smaller or equals to than the amount in the account ● Input data: First name of John, last name of Young, date of birth of 02/10/2000, savings account, deposited amount of 200, withdrawn amount of 150 	"John Young 2/10/2000(S) Withdraw - balance updated."

Class name: TransactionManagerController Method signature: private void handleDisplayButton(ActionEvent event){ // Prints all accounts in database, sorted by account types			
Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method returns a string if the user is trying to print the accounts when the account database is empty	<ul style="list-style-type: none"> ● Accounts are not loaded into database and the button is selected to print all accounts in database, sorted by account types 	"Account Database is empty!"
2	This method returns a string followed by all the all the accounts in database and a string	<ul style="list-style-type: none"> ● Accounts are loaded into database successfully and the button is selected to print all accounts in database, sorted by account types 	"*Accounts sorted by account type and profile." followed by all accounts in database sorted by account

	indicating the end of list if the accounts are in the database successfully		type and a string “** end of list.”
--	---	--	-------------------------------------

Class name: TransactionManagerController Method signature: private void handleLoadAcc(ActionEvent event){} // Loads from a selected file into accounts database			
Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method returns a string if the accounts are loaded successfully	<ul style="list-style-type: none"> Load from a file that has valid data 	“Accounts loaded.”
2	This method returns a string stating that a file is not found	<ul style="list-style-type: none"> Load from a file that has invalid data 	“File not found”
3	This method returns a string if the file that is loaded is empty	<ul style="list-style-type: none"> Load from a file that has empty/zero data 	“Account Database is empty!”
4	This method returns a string if the file has missing data for loading account	<ul style="list-style-type: none"> Load from a file that has missing data 	“Missing data for loading an account”

Class name: TransactionManagerController Method signature: private void handleDisplayFeesInterestButton(ActionEvent event){} // Prints all accounts in database sorted by account types with calculated fees and monthly interests			
Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method returns a string if user is trying to display fees and interest when the account database is empty	<ul style="list-style-type: none"> Accounts are not loaded into database and the button is selected to print all accounts in database with calculated fees and monthly interest 	“Account Database is empty!”

2	This method returns a string followed by all the all the accounts in database with the calculated fee and monthly interest and a string indicating the end of list if the accounts are in the database successfully	<ul style="list-style-type: none"> Accounts are loaded into database successfully and the button is selected to print all accounts in database with calculated fees and monthly interest 	“*list of accounts with fee and monthly interest” followed by all accounts in database sorted by account type with the calculated fees and monthly interests and a string “* end of list.”
---	---	---	--

Class name: TransactionManagerController

Method signature: private void handleUpdateAccountsButton(ActionEvent event){}
// Prints all accounts in database sorted by account types after applying fees and monthly interests to accounts

Test Case #	Requirement	Test description and Input Data	Expected result/output
1	This method returns a string if user is trying to display applied fees and interest when the account database is empty	<ul style="list-style-type: none"> Accounts are not loaded into database and the button is selected to print all accounts in database with applied fees and monthly interest 	“Account Database is empty!”
2	This method returns a string followed by all the all the accounts in database with the applied fee and monthly interest and a string indicating the end of list if the accounts are in the database successfully	<ul style="list-style-type: none"> Accounts are loaded into database successfully and the button is selected to print all accounts in database with the applied fees and monthly interest 	“*list of accounts with fee and monthly interests applied.” followed by all accounts in database sorted by account type with the applied fees and monthly interests and a string “* end of list.”