Joshua Atienza (NetID: jra165), Kyle Lee (NetID: kl781)

Test Document for MainController Class

Test Case #	Purpose for test case	Input Data	Expected Output
0	Test open account functionality ● Case 1: Open account with insufficient information ○ Case 1a: Missing first name ○ Case 1b: Missing last name ○ Case 1c: Missing date ○ Case 1d: Missing balance ○ Case 1e: Missing type of account ● Case 2: Open account with sufficient, valid information ○ Case 2a: Checking i. Direct Deposit = True ii. Direct Deposit = False ○ Case 2b: Savings i. Loyal = True ii. Loyal = False ○ Case 2c: Money Market ○ Case 2d: Account already exists ● Case 3: Open account with sufficient information, invalid date ● Case 4: Open account with sufficient information, negative balance	*Case #1a* First Name: Last Name: Atienza Date: 9/26/1999 Balance: 500 AccountType: Checking *Case #1b* First Name: Josh Last Name: Date: 9/26/1999 Balance: 500 AccountType: Checking *Case #1c* First Name: Josh Last Name: Atienza Date: Balance: 500 AccountType: Checking *Case #1d* First Name: Josh Last Name: Atienza Date: 9/26/1999 Balance: AccountType: Checking *Case #1e* First Name: Josh Last Name: Atienza Date: 9/26/1999 Balance: 500 AccountType: *Case #2ai* First Name: Josh Last Name: Atienza Date: 9/26/1999 Balance: 500 AccountType: Checking Direct Deposit: True	*Case #1a* "Please enter a full name." *Case #1b* "Please enter a full name." *Case #1c* "Not enough information provided, account could not be created." *Case #1d* "Number format exception, enter a valid starting balance." *Case #1e* "No account type chosen." *Case #2ai* "Account opened and added to the database."

		Case #2aii First Name: Kyle Last Name: Lee Date: 2/26/1999 Balance: 500 AccountType: Checking Direct Deposit: False *Case #2bi* First Name: JP Last Name: Edralin Date: 4/11/1999 Balance: 500 AccountType: Savings Loyal: True *Case #2bii* First Name: Ken Last Name: Chang Date: 7/11/1998 Balance: 500 AccountType: Savings Loyal: False *Case #2c* First Name: Jasmine Last Name: Shen Date: 9/16/1999 Balance: 500 AccountType: Money Market *Case #2d (Josh Atienza Checking exists already)* First Name: Josh Last Name: Atienza Date: 9/26/1999 Balance: 500 AccountType: Checking Direct Deposit: True *Case #3* First Name: Josh Last Name: Atienza Date: 13/26/1999 Balance: 500 AccountType: Checking Direct Deposit: True	*Case #2aii* “Account opened and added to the database.” *Case #2bi* “Account opened and added to the database.” *Case #2bii* “Account opened and added to the database.” *Case #2c* “Account opened and added to the database.” *Case #2d* “Account is already in the database.” *Case #3* “13/26/1999 is not a valid date!”

		Case #4 First Name: Josh Last Name: Atienza Date: 9/26/1999 Balance: -1000 AccountType: Checking Direct Deposit: True	*Case #4* “No negative numbers, please enter a valid number.”
1	Test close account functionality ● Case 1: Close account with insufficient information (missing first/last name) ● Case 2: Close account with sufficient information (first name, last name) ○ Case 2a: Account exists ○ Case 2b: Account doesn’t exist	*Case #1* First Name: Susan Last Name: *Case #2a* (In database) First Name: Susan Last Name: Anthony *Case #2b* (Not in database) First Name: Lavar Last Name: Ball	*Case #1* “Account does not exist.” *Case #2a* “Account closed and removed from the database.” *Case #2b* “Account does not exist.”

2	Test deposit functionality ● Case 1: Deposit with insufficient information ○ Case 1a: Missing first name ○ Case 1b: Missing last name ○ Case 1c: Missing amount ○ Case 1d: Missing type of account ● Case 2: Deposit with sufficient, valid information ● Case 3: Deposit with sufficient information, invalid amount	*Case #1a* First Name: Last Name: James Amount: 45.00 AccountType: Checking *Case #1b* First Name: LeBron Last Name: Amount: 45.00 AccountType: Checking *Case #1c* First Name: LeBron Last Name: James Amount: AccountType: Checking *Case #1d* First Name: LeBron Last Name: James Amount: 45.00 AccountType: Case #2 (Lebron James in database) First Name: LeBron Last Name: James Amount: 45.00 AccountType: Checking Case #3 First Name: LeBron Last Name: James Amount: -45.00 AccountType: Checking	*Case #1a* “Account does not exist.” *Case #1b* “Account does not exist.” *Case #1c* “Number format exception, enter a valid amount to deposit.” *Case #1d* “Select a valid account type to deposit into.” *Case #2* “45.00 deposited to account.” *Case #3* “No negative numbers, please enter a valid number.”
3	Test withdraw functionality ● Case 1: Withdraw with insufficient information ○ Case 1a: Missing first name ○ Case 1b: Missing last name ○ Case 1c: Missing amount ○ Case 1d: Missing type of account ● Case 2: Withdraw with sufficient, valid information	*Case #1a* First Name: Last Name: Lee Amount: 20.00 AccountType: Checking *Case #1b* First Name: Kyle Last Name: Amount: 20.00 AccountType: Checking *Case #1c*	*Case #1a* “Account does not exist.” *Case #1b* “Account does not exist.” *Case #1c*

	○ Case 2a: Enough funds ○ Case 2b: Not enough funds ● Case 3: Withdraw with sufficient information, invalid amount	First Name: Kyle Last Name: Lee Amount: AccountType: Checking *Case #1d* First Name: Kyle Last Name: Lee Amount: 20.00 AccountType: Case #2a First Name: Kyle Last Name: Lee Amount: 35.00 AccountType: Checking (Account Balance Before: 50.00) Case #2b First Name: Kyle Last Name: Lee Amount: 35.00 AccountType: Checking (Account Balance Before: 20.00) Case #3 First Name: Kyle Last Name: Lee Amount: -35.00 (Account Balance Before: 20.00)	“Number format exception, enter a valid amount to withdraw.” *Case #1d* “Select a valid account type to withdraw from.” Case #2a “35.00 withdrawn from account.” Case #2b “Insufficient funds.” Case #3 “No negative numbers, please enter a valid number.”
4	Test print functionality ● Case 1: Print empty database ● Case 2: Print regular ● Case 3: Print account by date ● Case 4: Print account by last name	*Case #1* Database: *empty* *Case #2* Database: **refer to added accounts in “test open functionality”**	*Case #1* Database is empty. *Case #2* “--Listing accounts in the database-- *Checking*Josh Atienza*$500.00*9/26/1999*direct deposit account* *Checking*Kyle Lee*$500.00*2/26/1999 *Savings*JP Edralin*$500.00*4/11/1999*special Savings account* *Savings*Ken Chang*

		        Case #3 Database: **refer to added accounts in “test open functionality”**	$500.00*7/11/1998 *Money Market*Jasmine Shen* $500.00*9/16/1998*0 withdrawals* --end of listing-- *Case #3:* --Printing statements by date opened-- *Savings*Ken Chang* $500.00*7/11/1998 -interest: $ 0.10 -fee: $ 0.00 -new balance: $ 500.10 *Money Market*Jasmine Shen* $500.00*9/16/1998*0 withdrawals* -interest: $ 0.26 -fee: $ 12.00 -new balance: $ 488.27 *Checking*Kyle Lee* $500.00*2/26/1999 -interest: $ 0.02 -fee: $ 25.00 -new balance: $ 475.02 *Savings*JP Edralin* $500.00*4/11/1999*special Savings account* -interest: $ 0.15 -fee: $ 0.00 -new balance: $ 500.15 *Checking*Josh

		Case #4 Database: **refer to added accounts in “test open functionality”**	Atienza* $500.00*9/26/1999*direct deposit account* -interest: $ 0.02 -fee: $ 0.00 -new balance: $ 500.02 --end of listing-- *Case #4* --Printing statements by last name-- *Checking*Josh Atienza* $500.02*9/26/1999*direct deposit account* -interest: $ 0.02 -fee: $ 0.00 -new balance: $ 500.04 *Savings*Ken Chang* $500.10*7/11/1998 -interest: $ 0.10 -fee: $ 0.00 -new balance: $ 500.21 *Savings*JP Edralin* $500.15*4/11/1999*special Savings account* -interest: $ 0.15 -fee: $ 0.00 -new balance: $ 500.29 *Checking*Kyle Lee* $475.02*2/26/1999 -interest: $ 0.02 -fee: $ 25.00 -new balance: $ 450.04 *Money Market*Jasmine

			Shen* $488.27*9/16/1998*0 withdrawals* -interest: $ 0.26 -fee: $ 12.00 -new balance: $ 476.54 --end of printing--
5	Test import functionality ● Case 1: Import text file ○ Case 1a: All valid tokens in text file ○ Case 1b: InputMismatchException (wrong data type in input) ○ Case 1c: NoSuchElementException (insufficient data provided) ● Case 2: Click import, but don't import anything (NullPointerException)	*Case #1a* ActionEvent event is initiated, followed by the selection of a valid text file Text File: S,Josh,Atienza,500,1/1/2019,true *Case #1b* ActionEvent event is initiated, followed by the selection of a valid text file + line has wrong data type in input Text File: S,Josh,Atienza,500,1/1/2019,8 *Case #1c* ActionEvent event is initiated, followed by the selection of a valid text file + line has insufficient data provided Text File: S,Josh,Atienza *Case #2* ActionEvent event is initiated, followed by cancel of import file	*Case #1a* "File imported." *Case #1b* "Invalid input data type, account not created." *Case #1c* "No such element exception, not enough information provided." *Case #2* "File not selected, unable to import transactions to file."
6	Test export functionality ● Case 1: Export text file ● Case 2: Click export, but don't export anything	*Case #1* ActionEvent event is initiated on export, followed by the selection of a valid export file name and an associating directory	*Case #1* AccountDatabase is written to a text file named by the user. and TextArea displays "File

		Case #2 ActionEvent event is initiated on export, followed by cancel of export file	successfully exported.” *Case #2* “File not selected, unable to export transactions to file.”
7	Test clear functionality ● Case 1: Few to all populated elements on Open/Close tab	*Case #1* ActionEvent event is initiated on clear	*Case #1* (all data fields are cleared)