DOCUMENTATION: EXPENSE TRACKER

Form 1 variables:

- user: (user input) stores username
- status: (returned) status either to welcome new user or welcome back old user
- NEW_EXPENSE_TEXT: (constant variable) show text "New Expense"
- reportText: (constant variable) show text "Report"
- editRecurringChargeText: (constant variable) show text "Edit Recurring Charge"

NewExpense variables:

- user: user's name, stores the username passed in from Form 1
- storeNames: (returned from MySQL) list of names and ID of stores in the database
- categoryNames: (returned from MySQL) list of categories and ID of the expense in the database
- recurStatus: (set based on user's choice of YES and NO) true if user select yes and false if user select no
- numPerson: (user's input) number of people the expense is shared with
- currentDate: (user's input) date of the expense
- sharedNamesString: (user's input) names of people inputted by user in string format
- numComas: number of comas in the sharedNamesString has
- totalRecurMonths: number of months the recurring charge defined by user
- numberOfMonths: number of months the recurring charge defined by user, subtracted by 1 after each charge is added to the expense table in MySQL

Monthly and MonthlyReport variables:

- username: stores the username passed in from Form 1
- startRange: (user's input) date, start of the range of the report
- endRange: (user's input) date, end of the range of the report
- monthExpenseData: (data table returned from MySQL) contains the rows and columns for the expense from MySQL
- monthlyExpenses: (data table) stores the DataTable passed in from Monthly form
- startDate: start of the range, passed in from Monthly form
- endDate: end of the range, passed in from Monthly form
- sumAmount: (returned from MySQL) total amount for the expenses within range
- sumCat: (data table returned from MySQL) contains total amount of each expense category
- sumSharedAmount: (returned from MySQL) total of the amount of shared expenses
- sumOtherShared: (returned from MySQL) total amount of expenses others shared with you
- sumPerson: (data table returned from MySQL) contains total amount of each person that shared expenses with you

EditCharge variables:

- username: stores username passed in from Form 1
- stores: (data table returned from MySQL) list of store names and ID in the database
- categories: (data table returned from MySQL) list of categories and ID in the database
- sharedStatus: (user's input) if the expense is a shared expense

- storeID: ID of store
- catID: ID of the expense category

CommitChange variables:

- expenseDate: date of expense passed in from EditCharge form
- expenseAmount: amount of expense passed in from EditCharge form
- expenseSharedStatus: status whether the expense is shared or not passed in from EditCharge form
- storeID: store ID passed in from EditCharge form
- catID: category ID passed in from EditCharge form
- newExpenseAmount: new amount inputted by user
- status: returned status whether the amount was changed or the expense cannot be found
- changeDateStatus: returned status whether the date was changed or the expense cannot be found

ExpenseTrackerCode variables:

- storeNames: (returned from MySQL) list of names and ID of stores in the database
- categoryNames: (returned from MySQL) list of categories and ID of the expense in the database
- numCatId: (default = 0) stores expense category id
- numStoreId: (default = 0) stores store id
- listOfNames: list of names of people whom the expense is shared with
- numSharedPercent: calculated percent of the expense is shared
- namesOfPeople: temporary stores the name of people from string
- expenseAmount: amount of the expense
- expenseDate: date of the expense
- expenseUser: user inserting the expense
- expenseSharedStatus: (true or false) if the expense is a shared expense
- expenseNumShared: total number of people in the shared expense including the user themselves
- expenseNumSharedPercent: percent of the amount each person pays in the shared expense
- expenserecurCharge: (true or false) if the expense is a recurring charge
- expenseSharedPersonName: list of names of people in the shared expense
- expenseAmountShared: amount each person pays in the shared expense
- Value: total amount returned if the expenses exist
- statusExist: status if the expenses exist, YES for exist
- dataTable: (in GetSumByCat) contains rows and columns from the expense table with the summation by expense category, returned to Monthly
- totalShared: total amount of the shared expenses
- value: (in SumOtherShared) total amount of expenses others shared with user
- newTable: (in GetSumByPerson) total amount by person of the expenses are shared with
- expenseDate: (in CheckRowExist) inquired expense date
- expenseAmount: (in CheckRowExist) inquired expense amount
- expenseShared: (in CheckRowExist) inquired shared expense status

- username: (in CheckRowExist) username of the user
- status: (in CheckRowExist) the datatable returned if the expense exist
- statusExist: (in ChangeRecurringAmount) status returned from MySQL – either that the expense doesn’t exist or amount is changed
- statusExist: (in ChangeRecurringDate) status returned from MySQL – either that the expense doesn’t exist or date is changed
- statusExist: (in DeleteRecurringCharge) status returned from MySQL – either that the expense doesn’t exist or the expense was deleted

FEATURES:

FORMS explanation:

Form 1: Allows user to input their username follow by picking their choice of what transaction they would like to perform

1. usernameInput_TextChanged(): Changing user's input to string

2. usernameInput_KeyDown(): User press enter after entering their username, Show the choices for user to choose

3. submitButton_Click(): User click submit button after entering their username, Show the choices for user to choose

4. newWelcome_Click(): Show welcome message custom to user's name

5. hideUsernameInsertBox(): Hide username input textbox and label

6. userChoicesBox(): Show user the choices for different transaction

7. enterChoice_Click(): After user submit choice by clicking button: shows different form dependent of user's choice

8. closeButton_Click(): Closes the form if the user press the close button

NewExpense: Allows user to input new expense

9. StoreNameLabel(): Function to call ExpenseTracker class to obtain all the store names in database, use to fill ComboBox for store's name

10. storeName_Validating(): Validates that the user pick or input a store name

11. CategoryLabel(): Function to call ExpenseTracker class to obtain all the store names in database, use to fill ComboBox for expense category

12. expenseCats_Validating(): Validates if user pick or input expense category

13. amount_Validating():Validates that user input amount for the expense

14. recurYes_CheckedChanged(): Shows the Textbox for user to input the amount of recurring months if user choose YES for recurring charge

15. recurMonths_Validating(): Validates if user input number of months for the recurring charge

16. sharedYes_CheckedChange(): If the user select YES, shows the textboxes for number of people and names of people the charge is shared with

17. numShared_Validating(): Validates that the user input number of people the expense is shared with

18. sharedNames_Validating(): Validates that the user input and have the correct number of name(s) of people for the shared expense

19. submit_Click_1(): Expense is saved to the database after the user click the submit button

Monthly and MonthlyReport: Allows user to view the report for a particular range specified by user

20. closeButton_Click(): Window closes if user click the CLOSE button

21. submitButton_Click(): Shows the report of the expense within the specified range when user click SUBMIT button

22. constructor MonthlyReport(): Prints report of the expense in the specified range

23. closeButton_Click(): (in MonthlyReport form) closes the window

EditCharge: Allows user to edit the amount and/or date or of a recurring charge or delete a particular recurring charge

24. StoreNameLabel(): Calls function in ExpenseTracker class to obtain all store names and add to ComboBox

25. CategoryLabel(): Calls function in ExpenseTracker class to obtain all expense categories and add to ComboBox

26. change_Click(): Shows the form that allows the user to edit the amount and date of the recurring charge after the Change button is click

27. delete_Click(): Delete the particular recurring expense after user clicks delete

CommitChange:

28. constructor CommitChange(): set data source of gridview to show the selected recurring charge

29. changeAmount_Click(): When Change Amount is clicked, calls function in ExpenseTracker class to edit the amount of the recurring charge

30. changeDate_Click(): When Change Date is clicked, calls function in ExpenseTracker class to change the date of the recurring charge

31. close_Click(): CLOSE button clicked: closes window

ExpenseTracker Class:

32. InsertUsername(): Check if username exist and insert new username to username table, returns status: welcome new user or welcome back

33. CheckStoreName(): Get all store names and its id from stores table in MySQL, put in a list as a tuple, returns that list

34. CheckExpenseCategory(): Get all expense category and its id from expense_categories table in MySQL, put in a list as a tuple, returns that list

35. SetCatId(): Obtain category id based on category name, checked against list of tuple created above, if not found, add the new category to expense_categories table in MySQL

36. SetStoreId(): Obtain store id based on store name, checked against list of tuple created above, if not found, add the new store name to stores table in MySQL

37. calcAmountPerPerson(): Calculate amount each person owes in a shared expense

38. sharedNamesIntoList(): Convert a string of names of people in the shared expense into a list

39. sharedPerson(): Inserting shared expense with names of people the expense is shared with into shared_person table in MySQL

40. InsertExpense(): Insert the new expense to the expense table in MySQL

41. GetExpenses(): Returns expenses in between the start and end date

42. SumMonthTotal(): Sums the total amount for the expenses in between the input dates

43. GetSumByCat(): Getting total amount by expense category

44. SumSharedAmount(): Calculating the total shared amount - amount of the expense is shared with others

45. SumOtherShared(): Calculating the amount of the expenses others shared with user

46. GetSumByPerson(): Calculating the total shared amount by person of the expenses are shared with

47. CheckRowExist(): Connect to MySQL to check if the input expense exist and returns the expense if it exists

48. ChangeRecurringAmount(): Connect to MySQL to change the amount of a particular recurring charge

49. ChangeRecurringDate(): Connect to MySQL to change the date of a particular recurring charge

50. DeleteRecurringCharge(): Connect to MySQL to delete particular recurring charge

MySQL tables:

Expense: primary key: ID (auto-generated) main expense table, stores all the expenses inputted by user

#	**Field**	**Table**	**Type**
1	ID	expense	INT
2	STORE_ID	expense	INT
3	Amount	expense	FLOAT
4	Shared_expense	expense	BOOL
5	Shared_number	expense	INT
6	Percent_shared	expense	FLOAT
7	Category_id	expense	INT
8	Recurring_charge	expense	BOOL
9	Shared_amount	expense	FLOAT
10	Date	expense	DATE
11	USER_ID	expense	VARCHAR(500)

Store: primary key: Store_id (auto-generated) stores name of the stores and their id

#	Field	Table	Type
1	Store_name	store	VARCHAR(500)
2	Store_id	store	INT

Expense_categories: primary key: Category_id (auto-generated) stores the expense categories and their id

#	Field	Table	Type
1	Expense_cat	expense_categories	VARCHAR(500)
2	Category_id	expense_categories	INT

Shared_person: primary key: ID (auto-generated) stores the name

#	Field	Table	Type
1	ID	shared_person	INT
2	Name	shared_person	VARCHAR(500)
3	expense_date	shared_person	DATE
4	amount	shared_person	FLOAT
5	store_id	shared_person	INT

Username: stores all usernames

#	Field	Table	Type
1	user_name	username	VARCHAR(500)