Alex Smith

Professor Rivas

Software Development 1

4/4/2017

Project 2 Milestone – Household Budget Manager

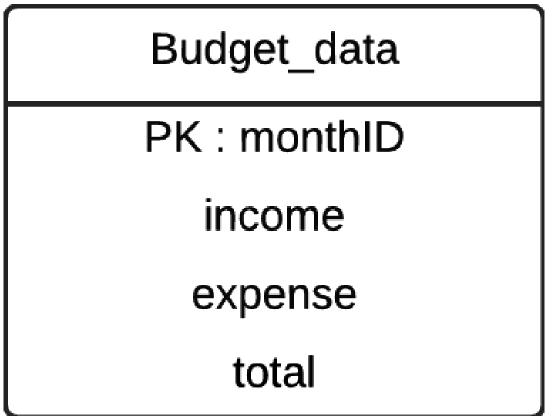
My project is a household budget manager that will be able to save the user's data about their income and expenses to calculate how much they are overbudget or underbudget. It will incorporate a database on a server so the data does not need to be inputted by the user each time. This milestone report will state what progress I have made so far, what processes still need to be accomplished in order to fulfill the project proposal, and a brief description of the system and of the UML diagram.

Work on the budget management program began shortly after the submission of the proposal. After researching multiple languages and methods used for creating databases, PostgreSQL was chosen for this project to create the database and manipulate the data in tables. Initially, a simple diagram (displayed below) detailing the table name, primary key, and attributes was created to serve as a guide. The primary key is the monthID, which determines which month the income data and expense data is for. The other columns in the table include income, expense, and total, which determines if the user is overbudget or underbudget by subtracting the expenses value from the income value and using an if statement to determine if the value is positive or negative. In addition to the table, a class UML diagram has been created detailing all of the methods and their purpose. Most of the planning and database operations have been completed for this project.

In order to fulfill the project proposal, more work on the Java code is necessary. The methods detailed in the UML class diagram have not been coded but have been researched on how they will be implemented. The server and database need to be connected to the Java file so methods can alter and manipulate the data in the table. Once the table has been connected and the methods have been programmed according to the UML class diagram, the project will be able to fulfill the requirements in the proposal.

The attached UML class diagram describes what methods the budget management system can execute and has a brief description of each data field and method. There are seven total data fields in the class. The monthID is an integer that determines which month the user is inputting the data for and can be changed in order to store multiple months' worth of data. The income and expense are doubles that store values input by the user and can be inserted into the table in the database. The total determines if the user is overbudget or underbudget depending if the income is greater than the expense or not and the exact amount is determined through subtraction of the expense value from the income. The change value is a double that stores the result of comparing the total from multiple rows in order to show trends over time. The last two data values are month1 and month2 which are used in the getChange method. Including the default constructor, there are 7 total methods. The insertIncome and insertExpense methods prompt the user to list the respective data, which is then summed up and put in the respective column. The currentIncome and currentExpense methods print the value of the income and expense variables respectively. The getBudget method updates the value of the total variable and prints it. The last method, getChange, requires two total values in different rows and compares them to using greater than and less than operators and subtraction to print either an increase or a decrease in total funds available at the end of the month based upon the data in the table. This method will also compare the income and expenses for each month using the same operators. These data fields and methods in the class diagram will be programmed in Java and will fulfill the project proposal.

This project is introducing me to the applications databases and how to create them. In addition to the basic features of a money tracking program, this program will also be able to show trends between multiple months and detect a decrease or increase in the user's income or expenses. This project has practical uses as a budget system for a household.



Budget_data

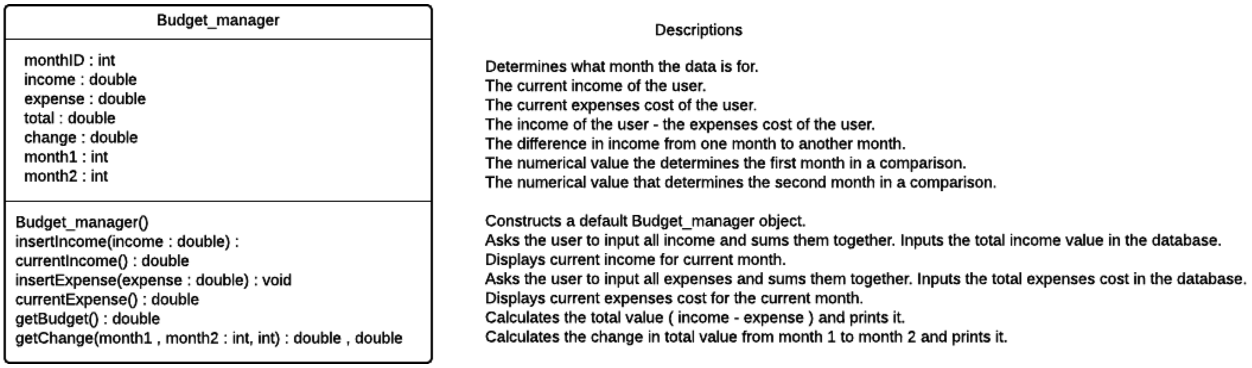
PK : monthID

income

expense

total

Database Diagram



Budget_manager

monthID : int
income : double
expense : double
total : double
change : double
month1 : int
month2 : int

Budget_manager()
insertIncome(income : double) :
currentIncome() : double
insertExpense(expense : double) : void
currentExpense() : double
getBudget() : double
getChange(month1 , month2 : int, int) : double , double

Descriptions

Determines what month the data is for.
The current income of the user.
The current expenses cost of the user.
The income of the user - the expenses cost of the user.
The difference in income from one month to another month.
The numerical value the determines the first month in a comparison.
The numerical value that determines the second month in a comparison.

Constructs a default Budget_manager object.
Asks the user to input all income and sums them together. Inputs the total income value in the database.
Displays current income for current month.
Asks the user to input all expenses and sums them together. Inputs the total expenses cost in the database.
Displays current expenses cost for the current month.
Calculates the total value (income - expense) and prints it.
Calculates the change in total value from month 1 to month 2 and prints it.

UML Class Diagram with descriptions

Bibliography

1. LIANG, Y. DANIEL. "Objects and Classses." INTRODUCTION TO JAVA PROGRAMMING. Vol. 10. S.l.: PRENTICE HALL, 2015. 322-64. Print.