IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 2, Ver. III (Mar-Apr. 2016), PP 27-30
www.iosrjournals.org

Personal Financial Assistant

Kuntal Biswas

Abstract: Managing our finances has always been an important part of our life. While it is easy to let situation define our expenses, it becomes exceedingly difficult to keep track of all the spending, and is not very appealing to dedicate any certain amount of time to remember, calculate, or define a limit to it. It would be much easier if someone could do the same for us, efficiently. Our project proposes an idea of a wallet, that stores and analyzes a user's usage pattern, provides a system for loan and savings. It also provides a general prediction for the user's desired month's expenditure based on previous usage data. A provision for grouping and categorization of spending has also been suggested, based on real world usage. Monthly usage stats can be stored, and reviewed later by the user in graphical form, since it is easy to comprehend at a glance. The following introduction section discusses the importance, as well as the applicable scenarios for our project. It describes the difficulties and their possible solutions. The next section of methodology presents a simple block diagram of the working model of our project. It discusses the mathematical formula used, their justification and how the final simplified expression was realized. It also briefly discusses the algorithms used in various sections. The following results section presents some screenshots of the prototype program in execution, a simple interface, and a graph made using some test values. The final section of conclusion and future work discusses the current shortcomings of this project, and their possible resolution in future. A few solutions have been suggested as well, which although, are beyond the scope of this project, but can be implemented by collaboration of other resources.

I. Introduction

Having a fast paced lifestyle is both a boon as well as a bane. On one hand, it provides a sense of achievement, a satisfaction of moving at the pace of the world. On the other, it deprives us of the simple amenity called time. Not everyone has the patience, or even willingness to do all redundant jobs, instead of dedicating that duration to a more productive area. Almost the same, is the problem of inefficiency with personal finances. Not everyone is willing to maintain their own journal, detailing each little spending, marking and planning where to save or cut costs. It takes a lot of time, constant vigilance, and a willingness to stick to a set budget, to be able to maintain a healthy financial system. A greater problem arises when no matter how great a month of planned spending is done, we have to do it again for the next month.

While a more traditional system of keeping logs, either digital or hard copies, is an option, offloading the same job to someone else, would lift off a certain amount of burden over some people who are struggling to maintain their budget. This has numerous other benefits as well. First, we have to do almost no work or calculations here, everything is recorded and processed by our assistant. Second, we do not have to repeat it every other month, our assistant takes care of it intelligently in the background. Third, every month has a different pattern of spending, which we do not have to care about. For example, we buy gifts in holiday season, which isn't repeated every month. These exceptions as a month, might pose a problem when we try estimating our expenditure. Here, this won't be a problem.

In this project, we propose an idea of a personal wallet which, maintains records of our expenditure, analyses our usage patterns, and forms a prediction of our future expenditure, based on our previous usage. Records are maintained in a sqlite databse, with provision for multiple users. Separate tables have been assigned for every month, with divisions based on real world sections like, food, rent etc. These tables will also serve as the basis of graphs, that the user might request, as a review of his/her previous usage. The project has been designed in the Python programming language. Graphical user interface has been designed using Tkinter package. Matplotlib has been used to draw the graphs from tabular data.

In the following sections, we discuss the working theory, results and discussions over the processes used, and the mathematical formulae used for various algorithms that have been utilized in this project.

II. Methodology

In this paper, our main aim was to propose an idea of a personal wallet.

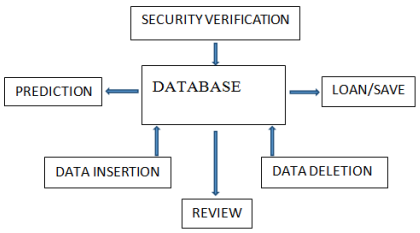

Fig. 1. Block Diagram representing the working modules

Security Verification: The security of each user is very important. So, every user needs a valid user ID and a password to log into his/her account. A separate table is maintained for the storage of user credentials. During login, a comparison is done between the user entered values and the stored credentials. Only when both user ID as well as password are found to be an exact match, the user is allowed to make any modification to his account.

Database: There are three main tables in this database. One for the storage of user credentials. One for keeping track of monthly salary. One for storing the details like counter variables, savings data and loan counters. Each added month creates a new table, with the name being a combination of user ID, month name and the year. The table contains two columns, first column contains the division names like 'Food', 'Rent' etc., and second column contains its values. Each added month has a corresponding entry in the salary table, and the respective changes are made to the data stored in the details table.

Prediction: Prediction of expected expenditure, is a very important aspect of this paper. We have assumed that the expenditure is dependent on the month. For example, months with holidays and birthdays are bound to have greater spending rate than other, regular months. Assuming that a person has 10 friends, and buys gifts for all 10 of them. For this very same month, next year, 10 presents will be bought. Only difference might be the inflation rate, which will essentially be reflected in the salary itself. Another important effect on expenditure could be an addition of a family member, which will not be a one month thing, but a regular, fixed increase every month. This is easily calculated by taking the average difference between the spending rate of last 24 months and the last 12 months, and including this average change during the calculation of expected expenditure for the current month, this present year.

$$\text{Current month Expenditure} = \frac{(\text{Average change} \times \text{current month salary})}{100} + \frac{(\text{previous month expenditure} \times \text{current month salary})}{\text{Previous month salary}}$$

The mathematical formula used is:

Fig. 2. Calculation of current month expenditure

Which is a reduced form of the expression:

$$\frac{(Sal\_p - Exp\_p) \times 100}{Sal\_p} \_\_ \frac{(Sal\_c - Exp\_c) \times 100}{Sal\_c} = \text{Average Change}$$

Fig. 3. Calculation of average change

Where Sal_p = previous year current month salary,
Sal_c = current year current month salary,
Exp_p = previous year current month expenditure,

Exp_c = current year current month expenditure.

Justification: Assuming that the user has at least 12 previous months of stored data, the average change in percentage of expenditure of the previous month, between previous year and current year is calculated by the formula in figure 3, since all the variables on the right hand side are known. Now, this average change is a value

that can be represented on the number line. Since this variable represents the increase/decrease in percentage of spending, it must also be reflected on the next month expenditure. It should also be noted, that this value is free from the expenditure difference between months of a single year. It only shows the difference of usage due to inflation, addition of a family member, or any permanent cause.

If the user has 24 months of data, the average of the average change can be taken as a more accurate value for average change. This ensures the inclusion of sudden increase or decrease in expenditure, in any particular month. This is more accurate than a direct comparison with last year's corresponding months.

This average change is used once again, using the formula in figure 3, only difference being, the current month is set to the month whose expected expenditure is to be found. Three variables from the left hand side, other than the current year current month expenditure (Exp_c) is known, and the right hand side is known. This gives a fairly accurate estimation of the total expected expenditure. The formula in figure 2 gives the very same result, and is a reduced form of the expression in figure 3. It is used for a simplification purpose, so that the debugging process could be easier.

Data insertion/deletion: It is done by simple sqlite commands through python. User is presented with a GUI, with specific, simple instructions for entry of data. Any commit button pressed, would commit the data to the database, for permanent storage.

Loan/Save: Two counter variables are maintained, one for the storage of average monthly deduction, and another for reverse counting the months till which the amount is to be deducted. Assuming that the user has loaned an amount from a different source, a provision has been made to enable a proper deduction of monthly installments from monthly savings, automatically without user intervention.

The user sets a loan amount, and the duration in which the loan amount is to be repaid. Monthly installments can be easily calculated by dividing the total loan amount by the duration. This installment amount is stored in a counter variable in the details table. The duration is also stored in the details table. For every added month, a check is made if the duration counter variable is zero or not.If the duration counter is zero, no extra deductions are performed on the monthly savings. If duration counter is not zero, the monthly installment counter value is deducted from the monthly savings amount, and the rest is added to total savings counter. Then, the duration counter is deducted by one, and this change is reflected in the details table.

Similarly, if the user requests to save an amount in a particular duration, two counters are maintained. One for the save amount, another for the duration. For every added month, a check is made if the duration counter variable is zero or not.If the duration counter is zero, no extra deductions are performed on the monthly salary. If duration counter is not zero, the monthly installment counter value is deducted from the monthly salary amount, and the rest is used in calculating a prediction that will allow that particular amount to be saved. Then, the duration counter is deducted by one, and this change is reflected in the details table.

Graph: The user may review his/her data in the form of a graph. The x-axis shows the selected months, preferably the last months in decreasing order. The y-axis shows either savings, or expenditure, depending on the request of the user. Graph has been selected as the method for review, since it is easier to comprehend at a single glance, rather than going through numbers in a tabular form.

III. Results

The prototype program is designed using python. Each section is defined in a different window, to avoid confusion. Buttons are clearly defined with their purpose. The results are displayed in the terminal, while the graph, designed using matplotlib package, is displayed in a separate window. Figures 4, 5 and 6 show the main windows with their respective features. Figure 4 shows the login window, with options to create new user, and for existing user to enter their credentials. Figure 5 shows the main window, where addition of month, prediction, loan and graph features can be used by the user. Clicking on each button takes the user to its respective window. Figure 6 shows the graph of 11 months. The green line shows savings, and the blue line shows expenditure, against the months marked at the x-axis.

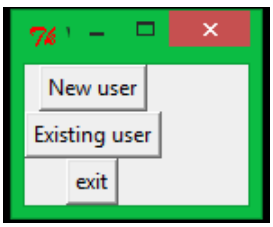

Fig. 4. Login window

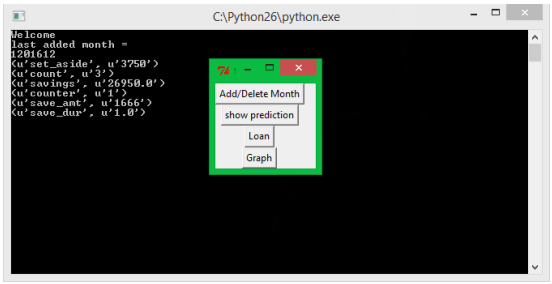

Fig. 5. Main window after login

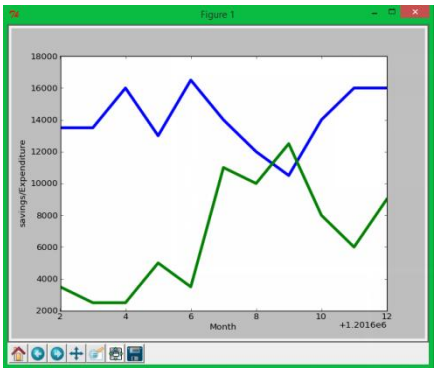

Fig. 6. Graph. Green = savings / Blue = expenditure

Conclusion and future work:

User convenience is always the priority when a new product is designed. Aim of this paper is the same, to introduce an idea of a convenient system for maintaining one's finances without having to go through lengthy data, and calculations. The more this is used, the more the database grows, allowing a fairly accurate rate of prediction.

Since this analyses the user's data, as well as stores it for future reference, a stronger security system could be implemented, to protect the user's privacy. Multi user support calls for better security measures, since the same hardware will be available to multiple users, and to protect individual data from other users. Also, if the user desires, collection of data into the database could be automated, by tracking usage of credit/debit cards, across multiple devices.