Track My Dollar V2.0 - BudgetOnTheGo[BOTGo]

Ashok Kumar Selvam
aselvam@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Rithik Jain
rjain25@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Sri Athithya Kruth Babu
sbabu@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Subramanian Venkataraman
svenka25@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

Zunaid Hasan Sorathiya
zhsorath@ncsu.edu
North Carolina State University
Raleigh, North Carolina, USA

ABSTRACT

For the next iteration of the projects, we have chosen to extend and work further on the MyDollarBot project. MyDollarBot is a Telegram Bot that enables users to track their expenditures across various common household categories, in effect creating a money management and tracking system for those aiming to be financially more savvy. The previous implemented the following features: adding, editing, deleting and viewing expenses. We have added the following features, significantly improving on this project: Overall budgeting, Category-wise budgeting, Complete modularization of the code base for further extensibility, Addition of test cases, Depiction of expenses using graphs and an estimator feature for the next month's expenses.

KEYWORDS

Telegram, Good Repository, Readability and Reusability, New Features, Extensiblity

1 INTRODUCTION

Telegram is a widespread messaging service with interfaces for access available through desktop application, mobile application on a variety of platforms and also a web UI. As such, users using Telegram can access it through almost any device with an internet connection and web surfing capabilities.

In light of this, we saw the potential of this project for extension. We also noted that while the previous team had done great work, there were several features worth adding that could be of great service to users of this Telegram Bot. The original work included an add feature to add new expenses, an edit feature to edit already added expenses, a delete feature to remove expenses and a display feature to display all expenses made.

In this project, we saw the potential for extension and development in multiple avenues: we noted that there was no way to keep track of a budget, say, if a user was trying to keep their expenditures below a certain threshold. There also was the question of whether this would be applied over all of their expenses, or just a few categories. We have decided to and implemented both of these as features in our project.

Next, we also saw that while the display feature gave an accounting of all expenses in the Telegram Bot's Database, it was bland and did not allow for easy and convenient understanding of where the most money was spent, that is to say, in which category. To solve this problem, we have implemented a display feature to display the user expenses in the form of pie charts and graphs, We strongly believe that this will help users better understand their own expenses.

In addition to the above, we also decided that an estimator feature to help users look at their projected spending for the coming week or month would add value, and thus we have implemented the same. We strongly believe this will help users plan their spending well based on a history of their spending.

We also noticed two major issues with the codebase, which we have recitfied completely - these were a distinct lack of test cases, and a complete deficit in modularization of the code. The entirety of the code was in a single file written in Python, and this did not make for easy modularization or understanding of the different features in the codebase. We made it part of our agenda to sort these out too, as we went ahead.

2 BUDGETING - OVERALL

In the initial version of MyDollarBot built by the previous team, we noticed that while expense tracking was the primary feature - we had addition, deletion and editing of expenses, there was no way for a user to track their expenses, which would definitely be a crucial part of any great money management application.

So, in this vein, we proceeded to add a general budgeting feature which would enable users to add a budget, remove a budget and edit a budget. The budget would work across different expense categories, and all expenses added in the MyDollarBot app for the month would be subtracted from this budget amount set by the user.

To explain further, a user can add a budget for the current month. Every expense they make would be subtracted from the budget every time they add an expense, and they would be informed of how they have remaining in their budget for the month. In case they would like to remove the budget due to a sudden financial windfall, they may do so through the budget removal feature.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

In case they feel the need to add more money to their budget, for example if they find themselves in a surplus, they can do this easily through the budget editing feature. In addition to this, there is also a budget view feature, where they can see the amount they have set for themselves.

3 BUDGETING - CATEGORY-WISE

While a user may see benefit in having a complete budget covering all categories of their expenditures, it might not be a requirement for them to budget their expenses in all categories of everyday expenses. They may be individuals who prefer to have a fixed amount dedicated to say, transport, or food for a month,

For this use case and user story, we came up with the category-wise budgeting feature. The category-wise budgeting features allow a user to set individual budgets for any or all categories in the MyDollarBot Telegram App. We currently offer the following categories: Food, Groceries, Utilities, Transport, Shopping, Miscellaneous.

A user can add individual budgets for one or some or all of these, and from then on track their monthly expenditure for those particular categories. We believe that this can assist people with lowering their expenses in certain categories, and can be a probable contributor to reduction of frivolous expenditures.

A user can also edit these budgets, individually or altogether to suit any changes in their plans for the month. As with the overall budgeting feature, every time a user adds an expense to a category, the expense gets deducted from that particular category's budget.

4 EXPENSE VISUALIZATION

The initial release of MyDollarBot featured a simple yet effective display to view all of a user's expenses in the current day or month, However, this does not inform the user of the overall trends of their spending, nor does it provide a big-picture view of it.

Perceiving this need, we have implemented a data visualization feature for users to view their spending over a month in form of bar charts and pie charts in the Telegram UI. This feature looks at the user's spending in the current month, and based on the data across different categories, generates a pie chart or bar graph.

This allows the user to see where how their spending is distributed in various categories, which in turn should allow for better planning and organization, leading to improvements in money management.

5 EDIT FEATURE IMPROVEMENT

The edit feature in the first iteration of MyDollarBot, involved the user typing in the date of the expense they wished to edit in a the following format: "12 September 2021". This naturally led to several possibilities for making errors, typos and formatting errors being the most obvious and identifiable.

We felt it would be better not to allow such a scope for error as it hindered user experience in MyDollarBot, and also introduced corner cases in terms of error handling which were tedious to handle.

To this end, we opted to re-write the edit feature, introducing an easier method for users to do this. The users are presented with a list of all expenses, and they can choose the ones they would like to edit.

6 REFACTORING FOR MODULARITY

In our work extending MyDollarBot, we initially noticed that the entirety of the codebase comprised only a single Python file. This, to us, seemed to be lacking, both from an easy-of-readability perspective, and also a extensibility perspective.

It would be very confusing, and also would make the file far too long if we continued as-is. So, we opted to first completely modularize the codebase, dividing the codebase into the pertinent sections for each of the already implemented features from the previous iteration before we proceeded with our own new features to improve the project.

We believe that this has and will contribute to significant ease of developer work in this repository, in addition to separation of concerns at a feature level, both of which are key for the long-term survivability of this work.

7 TESTING

Testing is a vital part of any software project, both for the continued smooth functioning of the software product, as well from a sanity-check perspective. The first iteration available on Github featured only a single test case, and as such has no code coverage to speak of.

Recognizing the importance of and the imminent requirement to fix the project in this area, we elected to proceed with writing and adding unit tests for all of the already existing features. Through this we have achieved a code coverage of 80 percent in the codebase, including our new features.

We believe this strongly adds to the overall robustness of the project, and makes it very easy for further extension and further work to happen on the same codebase.