CECS 443 Team 6

Budget Manager

URL: N/A

Github: https://github.com/miching/BudgetManegerCECS443

Project Proposal

1. **Project Title and Authors**
   a. Members: Javier, Michael, Thomas, Raphael, Dean
2. **Preface**:
   a. Readership: The software is meant for people who need help budgeting. The end user will be the customer. The software will be used externally.
   b. Version: 2.0
   c. Rationale: This document will provide user requirements, system requirements, and system models for the Budget Manager project.

3. **Introduction**: The goal of this proposed project is to help with the issue of budgeting for others.The problem is that many people struggle with budgeting their money due to the fact that they have no way to visualize it. We hope to solve this problem by clearly showing the client the spending that they due on a weekly, monthly or yearly basis. We will achieve this by using visuals. As well as providing an easy way for the client to get notifications on their spending and categorize the spending. When finished we will have a fully functional website that will help with the issue of budgeting with people.

4. **Architectural Change**:
    a. During the implementation of our project we decided that our architecture was going to be MVC (Model-View-Controller). We chose this initially because it seemed to be the best architecture intuitively. However, prior to learning about the different architectural models, we had already implemented ninety-percent of the project. Ultimately, the architecture for our project ended up being a simple frontend/backend architecture.
    b. We initially began to code our project without an architecture in mind until we were almost done with our implementation. We then decided that the Model View Controller was the best fit for our project. After our presentation we figured out that we had a simple frontend/backend architecture.
5. **Detailed Design Change** :
    a. The overall design of our system did not drastically change throughout implementation. However, as a result of changing some of the requirements, the system was altered slightly. Specifically, the system no longer had to accept user input for the different categories. Instead, the design featured hard-coded categories that would cover a vast majority of use cases.
    b. Moreover, since our notification requirement changed to notifying the user when they reached eighty-percent of the budget, the system was slightly altered. Instead of having to prompt the user for their notification threshold, the threshold was hard-coded to be eighty-percent.

    c. The rationale behind these changes mostly has to do with our time being limited. We figured that our previous system design would take too much time to implement. However, we still tried to implement a similar design that would not require as much time to code.

6. **Requirement Change:**
    - One requirement we changed was creating categories, to which we instead had pre-set categories the user could choose from.
        a) The overall design of the system slightly changed since it no longer had to prompt the user for different categories.
        b) Yes, this requirement has been implemented.
    - Another requirement we changed was the ability to set a threshold for a notification after passing the threshold in the budget. We instead just notified the user whenever the user has reached the 80% threshold of their budget.
        a) This did change our design.
        b) This changed our design because it altered the way a user interacted with our system, in particular our database. Instead of saving a threshold notification of the budget to the database, it now just manually calculates when entries are submitted to see if it has passed 80% of the budget.
        c) Yes, this requirement has been implemented.