CSCI-4830 Web App 2

**Budgeting Tool**

Software Requirements Specification

**Members:** Alec Bokowski, Christian Woodle, Gabriel Perez, James Triplett, Mark Turpen, Thomas Cordwin

March 7th, 2019

**Introduction**

Our product or web application is a budgeting tool that will help people manage their finances. Our tool will be easy to use and efficient so that users are able to quickly gather all the information they need to track their finances. With an organized and clean interface, users should find that our web application requires little to no learning to make the integration of this web application into a user’s budgeting process as seamless as possible.

The web application will require a few things from the user. To view their budgeting portfolio, the web application will need to authenticate users through a username and password combination. If the user does not have an account with us, they will be prompted to create an account. New users will have to input input their income as well as their expenses into our system. From the gathered information, our web application’s mathematical algorithm will generate a budget summary for the user to maintain. Users that are already in our system will be taken to their budget dashboard where they can view their current budget or make edits to their budget reflecting any changes to their income or expenses. Additional features will include taking in a savings goal, calculating the estimated time frame to reach that goal, and giving the user the option to have preset income/expense combinations for future use.

The requirements in detail are outlined below

**Requirements Specification**

**Functional:**

1. Users will be prompted to login when they enter the app
   1. Users will enter their username
   2. Users will enter their password
   3. Username and password will be authenticated to ensure they are a user
2. New users will click “Sign Up” and they will be taken to an account creation screen
   1. Users will enter their desired username
      1. Username will be checked to ensure it is unique
   2. Users will enter their password
   3. Both username and password will be saved in the database
3. New users will have to input their data into an “Income” and “Expenses” tab
   1. Users can click each tab to switch between them
   2. Income Tab will allow the user to input their amount of income from various sources, and how often that income occurs.
      1. Description, Value, Frequency, and Comment Column will be present
         1. There will always be an equal number of value and frequency columns
         2. By default there will be 3 income rows
      2. Description Column
         1. Text box will allow user to enter a description of the row
      3. Value Column
         1. Text box will allow user to enter a numerical value
         2. Only accept positive numerical values
      4. Frequency Column
         1. Text box will allow user to only enter a positive numerical value
         2. Dropdown box next to text box allows user to specify period. Possible values are:
            1. Daily
            2. Weekly
            3. Monthly
            4. Yearly

EX: A user gets paid “value”=1000 for their job “frequency”=2 times a “dropbox value”=”month”

* + 1. Comment Column
       1. Text box will allow users to enter a comment about the value and frequency
    2. Plus icon will create another row with a description,value, frequency, and comment column
    3. “X” icon will be present to the right of each “Comment” value which will allow user to delete a row
       1. When clicked, will ask the user “Are you sure you want to delete this row?” with the options “Yes” or “Cancel”
          1. Yes will delete the row whose “X” icon was clicked
          2. Cancel will cancel the delete query, and return to the previous screen.
    4. “Next” button will be displayed below the last row which will take the user to the “Expenses” tab
  1. Expense Tab will allow the user to input their amount of expenses for various things, and how often the expense occurs.
     1. Description, Value, Frequency, and Comment Column will be present
        1. There will always be an equal number of value and frequency columns
        2. By default there will be 3 expense rows
     2. Description Column
        1. Text box will allow user to enter a description of the row
     3. Value column
        1. Text box will allow user to enter a numerical value
        2. Only accept positive numerical values
     4. Frequency Column
        1. Text box will allow user to only enter a positive numerical value
        2. Dropdown box next to text box allows user to specify period. Possible values are:
           1. Daily
           2. Weekly
           3. Monthly
           4. Yearly

EX: A user buys “value”=20 of gas “frequency”=2 times a “dropbox value”=”month”

* + 1. Comment Column
       1. Text box will allow users to enter a comment about the value and frequency
    2. Plus icon will create another row with a description,value, frequency, and comment column
    3. “X” icon will be present to the right of each “Comment” value which will allow user to delete a row
       1. When clicked, will ask the user “Are you sure you want to delete this row?” with the options “Yes” or “Cancel”
          1. Yes will delete the row whose “X” icon was clicked
          2. Cancel will cancel the delete query, and return to the previous screen.
    4. “Previous” button will be displayed to the left below the last row
       1. Button will take the user back to the “Income” tab
    5. “Generate Budget” button will be displayed below the last row which
       1. Will generate the budget for the user
       2. Button will be grayed out unless at least one row is filled out in both “Income” or “Expenses” tab
    6. Clicking “Generate Budget” button will generate a “Budget” page based on the input data for the user

1. Previous users will automatically be taken to their “Budget” page
   1. The budget page will display a well organized summary of the user’s incomes and expenses.
   2. The budget page will show how much the user will save/lose per “time” chosen based on their information in the income and expense tabs.
      1. There will be a drop down box for “time” allowing the user to switch between monthly and yearly
   3. “Edit budget” button will appear in the top right bar while will allow users to edit information in their budget
      1. Upon clicking, user will be taken to “Income” and “Expenses” tab where they can change any values
         1. Standard validation will apply to all edits
      2. Clicking “Generate Budget” button will generate a new budget based on the new input data
2. “Account” button will appear in the top right corner of the web application after the budget has been generated
   1. “Change Username” will be an button
      1. Clicking the button will take the user to a change username screen
      2. User will be asked to enter their current username and their new username
         1. New username will be checked to ensure it is unique
      3. Upon Clicking “Change Username,” user will be presented with “Yes” or “Cancel” buttons
         1. After clicking “Yes”, user will be asked if they want to change their password as well
         2. Clicking “Cancel” will cancel the process
      4. User will be asked if they want to change their password as well
   2. “Change Password” will be a button
      1. Clicking the button will take the user to a change password screen
      2. User will be asked to enter their current password and their new password
      3. Upon clicking “Change Password,” user will be presented with “Yes” or “Cancel” buttons
      4. Clicking “Yes” will change their password
         1. After clicking “Yes”, user will be asked if they want to change their username as well
         2. Clicking “Cancel” will cancel the process

**Non-Functional:**

1. Front page must be easy to understand and use
2. The login page must be secure
   1. The login page must utilize a secure connection
   2. Passwords must not be stored as plaintext and should be encrypted
3. For efficiency, the system must recognize if a user is already logged in so the next time the user visits the web application, the user does not have to log in again
4. The web application and all of the data it holds must be secure
   1. Malicious users must not have unauthorized access to another user’s account
   2. The financial information of users must not be manipulatable by unauthorized users
   3. The financial information of users must be readily available upon the user’s request
5. Mathematical and financial calculations must be accurate
   1. Algorithms must be tamper-proof to ensure that monetary values are reported accurately.
6. The application should continue to function no matter the input given
   1. The types of exceptions that should be caught but does not stop execution include but are not limited to:
      1. Improper number formatting, i.e. “10.00..23” or “99.123”
         1. Positive numbers only.
            1. Expenses will be deducted as a positive number
         2. Numbers can have no more than 2 decimal places as per the U.S. currency system.
      2. Negative numbers
      3. Numeric fields having alphabetic characters
      4. Empty fields
7. Calculation page of the generated budget must present the calculated information neatly
8. The web application will be hosted on Heroku
   1. Heroku will enable the front and back-end services of the web application
      1. The front-end of the web application will be developed in HTML with support from JavaScript and CSS.
      2. The back-end of the web application will be composed of various Java Servlets. When end-users submit requests to the server, a Java Servlet will handle the request and respond with the appropriate HTML page
   2. PostgreSQL will be used in conjunction with Heroku to store a database that consists of users and income/expense sheets. For more details, refer to the Software Design Document.

System Evolution

After we have created our app and implemented the base features, we hope to receive feedback from our users. We want to hear what they think about the user interface and if it is intuitive. We want our users to enjoy our app, and the visual aspect is extremely important. If users do not like the placement or color scheme, then we will need to work to change this. Over time we will continue to update how the website looks so that it looks modern.

Once we have established the core features and released our app, we will look into adding additional functionality. We will listen to customer feedback and see what additions or changes they would like to make to the system. There are also a few features that we wish to implement in the future ourselves. We want to add a feature that tracks the users stocks that they may have purchased. They will be able to select their current stocks and the price that they bought them at. Our app will dynamically track the current price to see if they have made or lost money over time.

Additionally, we will add a feature that allows users to upload an Excel sheet or CSV file and we will generate a budget using this information. We will also create an option to export their budget to an Excel or CSV spreadsheet for user convenience.