cashflow Design

# Purpose

The reasons I keep a budget are:

* To ensure that 10% is given to Charity
* To save money for big purchases or goals
* Maintain an appropriate emergency fund
* Monitor account transactions for fraud
* To implement the 10% rule

# Cashflow Ideas and Concepts

Future:

* Maybe in the future I only need a few envelopes (funds): Charity, Bills, Goals (fund for each Goal), Spending
  + Categories: Bank, Spending, Charity, Goals
  + Funds: Bills, Spending, Charity, [fund for each goal]

# System Concepts

* Make an incremental change and just stop recording a couple of the biggest categories
* For example, stop distinguishing Spending from Living, and make a separate Goal category for gifts and big spending. So I’d be putting day-to-day spending in with Living and keeping more occasional spending separate.
* Gifts is already a fund, add some % distribution to it for now

Process

* Record all transactions except the Living category.
* When I balance my accounts, calculate the difference between the account balances and the budget balances
* Enter a transaction for Living equal to the difference calculated.

Future:

* Create tools to read in exported files from Chase, Widget, and AmEx and compare those transactions to the ones entered by hand, using the account, amount, and date (+/- 3 days) information. Line up transactions from bank with transactions entered, and identify any differences.

# Requirements

## Definitions

**Funds** – These are like the envelopes. Money is spent out of Funds. Each Fund is in a Category, and money spent from that Fund is deducted from the corresponding Category. For example, the Grocery Fund is in the Living category. When groceries are bought, the money is deducted from Living.

**Categories** – These are groups of funds, or groups of envelopes. Distributions are paid to Categories, not to Funds. For example, Living is a Category, and it includes Funds Grocery, Gasoline.

**Goals –** Goals are like temporary envelopes. They may receive distributions, so in that way they are similar to Categories. Is there a need to group Goals into categories as well, or associate them with a category? Should money for a goal come from distribution, or should it be transferred out of a category? Answer: create a Category called “Goals,” and pay Distribution to that. Specific goals will be handled like Funds, and their Category will be set to “Goals. So Goals is really a Category, and it contains a special type of Fund. What makes it special? It has an Active property, and it has a Target. Could I just do that for all Funds?

## Functions

* Enter transaction data
  + Allow user to insert new transactions above anywhere in the list
  + Mobile-friendly option
  + Three types:
    - Standard transaction (income or expense)
    - Transfer between Funds
    - Transfer between Accounts
    - Transaction split between multiple funds.
      * Saved as multiple transactions with the same sort order. Anytime sort order changes, all transactions have to move together.
      * Displayed differently: one date and description, but multiple amounts and funds
* Archive transaction data
  + When transactions are archived, save the balance of each account and fund
  + Allow user to manually enter the balance of an account to compare to the calculated balance.
* Allow user to change funds and categories
  + Set goals
  + View goals
  + Goals are stored in a flexible format
  + Goals can be completed and archived.
  + Funds can have target balances
  + Funds and categories are essentially the same thing. They are both envelopes that are intended to be used for a while. The both have optional target balances.
  + Goals also have a target amount, and they have an Active property that can be set to TRUE (Active) or FALSE (inactive)
  + Inactive goals are not loaded or sent to the client in most cases. The user has the option to view “Completed Goals,” in which case only the inactive goals are sent.
  + Goals have a “Date Completed” property that is populated with the current date when the Active property is set to False. This can be shown to the user when Completed Goals are displayed.
* Set Emergency fund (FUTURE)
  + Outcome drives the target for the fund called Emergency Fund
* Allow user to set Distribution amounts and effective date.
  + Income can be distributed to Categories or active Goals.
  + User can change distribution percentages through the web interface
  + User can see current and historical distribution amounts; so it might be best not to store those as properties of the Funds or Goals.
    - Option 1: Each Fund/Category and each Goal has a property called Distribution.
      * 1a: This property is a structure with two columns: effective date, and percentage
      * 1b: This property has only the value from the most recent Distribution. Historical values are kept somewhere else.
    - Option 2: Create an array of structures called Distribution. Each structure in the array has a field for effective date, and for each Fund/Category and each active Goal. When a new goal is added, part of the process includes adding it to the most recent Distribution structure. When a goal is set to Inactive, nothing happens immediately, but that Goal is not available as an option the next time Distributions are set. This means each structure in the Distribution array may have different fields.
    - Selection: Option 2 because it is most similar to the Excel implementation and it is cleanest because it keeps all of the information in one place that is easy to reference.

## Data to Store

Structures and properties

* List of Funds and Goals
  + Active
  + Balance – Funds don’t have balances, Categories do. But then does that mean each Goal needs to be a category?
  + Category
    - for all goals, Category=Goals
    - many funds will be in a category of the same name. For example, Home Improvement is both a fund and a category; it is the only fund in that category.
* Distributions
  + Current and historical distribution percentages
* Account Names and Balances

# Design

## Data Structures

Non-bullet items are variable names for the structures. Bullets underneath are properties of the structure.

Funds

* Bills
  + Active = TRUE
  + Balance =
  + Category = Bank
  + Target = [ ]
* Loans
  + Active = TRUE
  + Balance =
  + Category = Bank
  + Target = [ ]
* Bank Other
  + Active = TRUE
  + Balance =
  + Category = Bank
  + Target = [ ]
* Grocery
  + Active = TRUE
  + Balance =
  + Category = Living
  + Target = [ ]
* Dining
  + Active = TRUE
  + Balance =
  + Category = Living
  + Target = [ ]
* Gasoline
  + Active = TRUE
  + Balance =
  + Category = Living
  + Target = [ ]
* Spending Other
  + Active = TRUE
  + Balance =
  + Category = Spending
  + Target = [ ]
* Charity
  + Active = TRUE
  + Balance =
  + Category = Charity
  + Target = [ ]
* Marann / TRUE / / Marann / [ ]
* Ten Percent / TRUE / / Ten Percent / [ ]
* Emergency / TRUE / / Emergency / [ ]
* Gifts / TRUE / / Gifts / [ ]
* TEAM / TRUE / / TEAM / [ ]
* Wedding / TRUE / / Wedding / ($200x5 + shower gifts)
* etc… (see Names tab of Cashflow7.xlsx)

Distribution

* [0]
  + DateEffective = 9/13/2010
  + Bank = 0.63
  + Spending = 0.27
  + Charity = 0.10
* [19]
  + Date Effective = 3/15/2017
  + Bank = 0.40
  + Spending = 0.06
  + Charity = 0.10
  + etc… see Distribution tab of Cashflow7.xlsx

Accounts (keys = name)

* Cash
  + balance
  + dateBalanced = 4/14/2017
* WidgetSavings
  + balance
  + dateBalanced = 4/14/2017
* etc (see Cashflow tab of Cashflow7.xlsx, or G2:G8 of Names tab)

### Object Classes

The above data structures are implemented as an array of objects.

|  |  |
| --- | --- |
| Object | Properties |
| fund | active  balance  category  target |
| distribution | *all elements of categories\_list* are declared here |
| account | balance  date balanced |

The variable *Funds* is an array of objects of class ‘fund.’

The variable *Distribution* is an array of objects of class ‘distribution.’

The variable *Accounts* is an array of objects of class ‘account.’

## Data Storage

### Names

Store names of all accounts, funds, categories, and goals

|  |  |  |
| --- | --- | --- |
| accounts\_categories | | |
| Name | Type | Other Names |
| Cash | Account |  |
| Widget | Account | GE Checking |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| funds\_goals | | | | |
| Name | Type | Active | Category | Goal |
| Grocery | Fund | TRUE | Living | 0 |
|  |  |  |  |  |

## Scripts

### databaseConnect.php

Short php file to connect to the database and create a mysqli object.

### Other scripts

UI with form to add a transaction

Mobile-friendly UI for adding a transaction

PHP to perform the commit to the database

* Includes taking action based on Sort Order input (input may be Before item X)

Display all recently added transactions (not archived).

* Include an “Add Before” button at the end of each row to add a transaction with that sort number
* Include new balance of accounts and/or funds affected (may be 2 accounts, 2 funds, or one of each)

Display selected account or fund (like balance3 today)

* Includes updated balance of the account or fund on each row

Display all archived transactions that meet certain search criteria (Fund, Account, or Date range)

Archive transactions

* includes updating balances in cash\_balances

### names.php

Reads accounts\_categories table and funds\_goals table. Creates the following arrays:

* accounts\_list
* categories\_list
* funds\_list

### classes.php

Declare classes *fund*, *distribution*, and *account* according to [Object Classes](#_Object_Classes).

### dashboard4.html

Display all Account and Fund balances

* Last three entries in cash\_balances, and balance after all recent transactions

# Next Steps

1. (done) Create funds\_goals table in database (done), add all funds and goals currently in use
2. (done) Should I change array keys to the names for Accounts, Funds? Instead of Accounts[1], Accounts[Cash], or Accounts[Widget\_Checking].
   1. Yes, do that.
3. (done) What is the plan for Categories? There is no Category class right now. Review spec above and decide if I need to create one in names.php and pass it to read\_cash\_balances\_idx(). Otherwise I’m only updating the funds, which is stuff like Grocery, Bills, etc.
   1. I do need a Categories class.
   2. Added it to names.php and read\_cash\_balances\_idx().
4. Complete dashboard4.php
   1. Rename all documentation from .html to .php
   2. (done) Continue moving the “/\*” down the page and troubleshooting as I go.
   3. (done) Test and troubleshoot include for classes.php (done)
   4. test and troubleshoot names.php
   5. Read last three entries from cash\_balances
   6. Display last three entries from cash\_balances
5. Finish classes.php (distribution class)
6. Create function for processing transactions that can be called from dashboard4.php and from the function that adds things to the database.

# Appendix

From Todoist:

1) Adjust distributions  
- Emergency 2%  
- Adjust bank based on bills  
- Consider combining Spending and Living; maybe reopen Gifts as a separate category? Has there been any benefit to keeping Living expenses separate from Spending Money? I guess the benefit is that I could set aside my own spending money apart from Groceries and Dining. Consider rearranging and reallocating.  
- Create goal for car  
  
2) Create Goals as temporary categories   
3) Decide what to stop recording. Maybe Spending and Living? Maybe stop recording Bank and assume it will follow the budget each month?  
4) Specify way to do balancing/accounting. Use account balances to make assumptions about spending in unrecorded categories.