Adam Ring

Money Warp Documentation

Figure 1 ("Mobile Has Thrust the Purchase Funnel into Warp Speed")

Executive Summary

Money Warp is a software project to balance, keep track of, and Warp all your banking accounts. Once you have entered your reoccurring bills and income, you will be able to Warp forward in time and see how your account balances look at any date in the future. With this feature you will be able to forecast your savings accounts and see if you have enough money to pay all your bills.

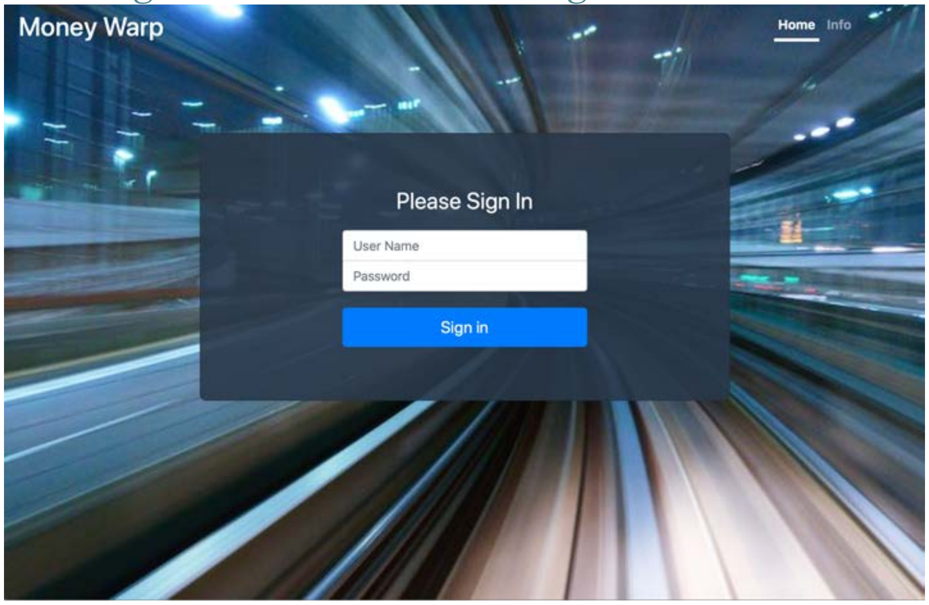
There are many other account balancing products available, however none of these have a solid forecasting model. The best alternate software with a forecasting feature, has a limit of only three months in the future. Money Warp will fill that void and allow unlimited forecasting.

Forecasting will be Money Warp's crown and jewel, and to support that Money Warp needs a foundation to work from. This foundation will incorporate the following: multiple account registers to enter deposits and expenditures; a monthly account balancing function; scheduling of deposits and expenditures; and reoccurring deposits and expenditures.

Money Warp's audience is anyone that wants to balance their accounts and preform Money Warp forecasting. The design and programming of Money Warp is tailored for ease of use; so many different people can easily take advantage of Money Warps advanced features.

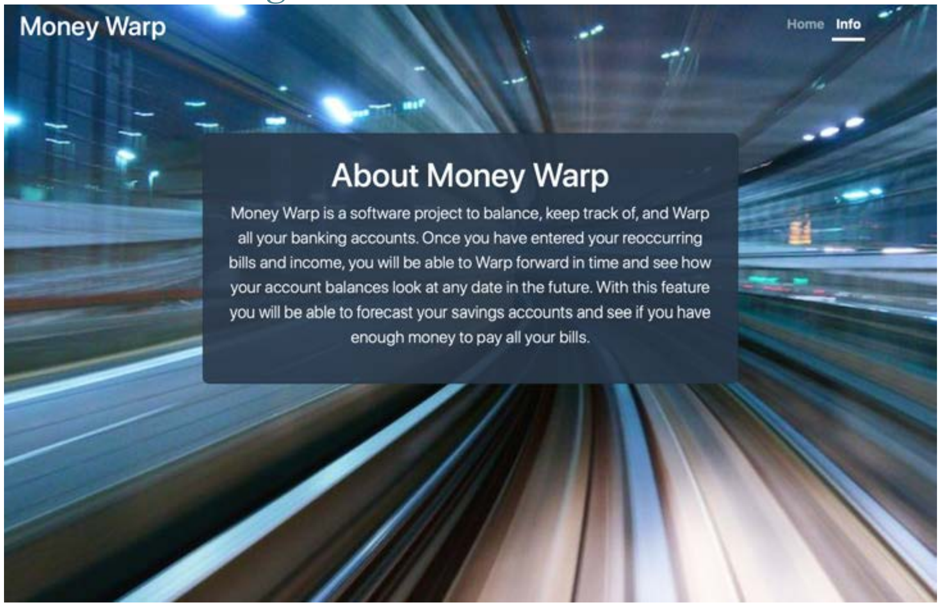
Usage Scenarios

Login | How a registered user would log in



This is the login screen. For phase I the system is a closed system and you would need to be granted access in order to use the system. With your credentials, you would enter your user name in the user name field and your password in the password field and then click on “Sign In”. If the provided user name and password are valid you will be granted access into the system.

Info | Informational Page

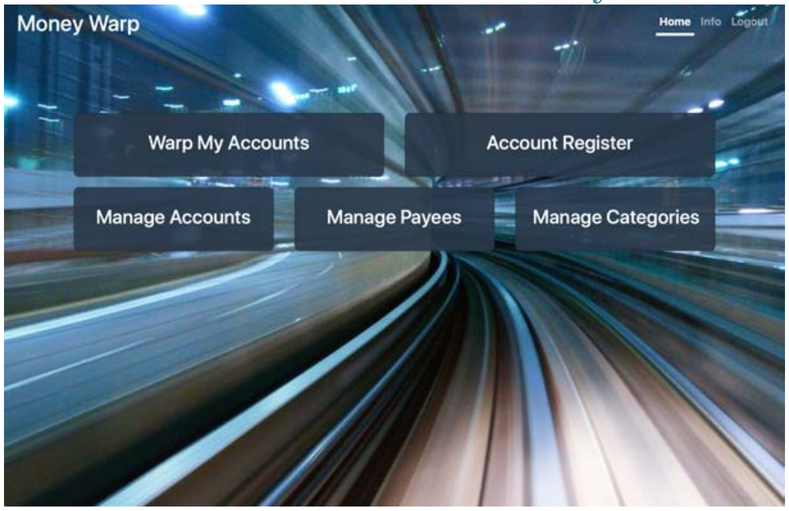


About Money Warp

Money Warp is a software project to balance, keep track of, and Warp all your banking accounts. Once you have entered your reoccurring bills and income, you will be able to Warp forward in time and see how your account balances look at any date in the future. With this feature you will be able to forecast your savings accounts and see if you have enough money to pay all your bills.

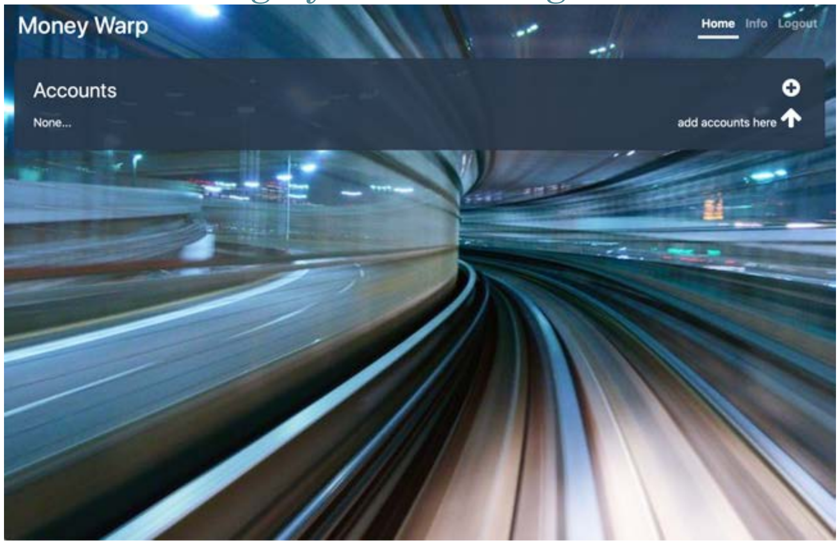
This is an informational page, giving a brief description of its purpose.

Home | Systems Home screen/Main Menu system

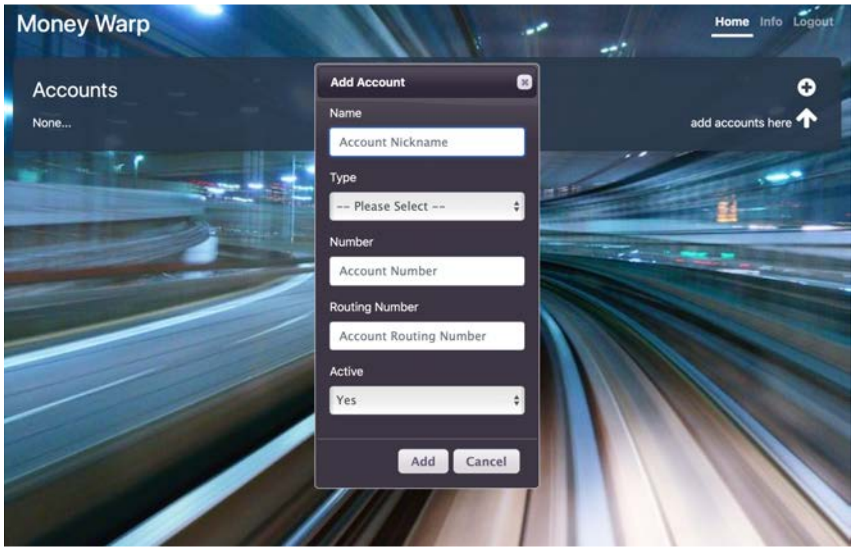


This is the “Home” page that you will see once logged in. From here you can jump to the five functions of system. To access any of the five functions, simply click on the function you would like to access. You can access this page from anywhere in the system by clicking on “Home” in the top right menu list.

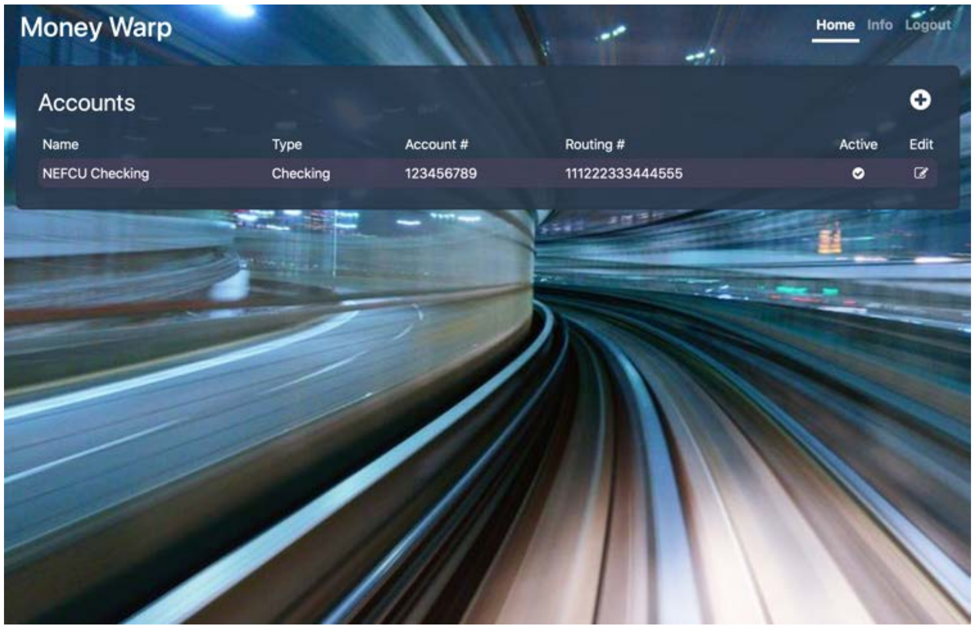
Accounts | Used to manage your banking accounts.



From the home screen click on "Manage Accounts" to access the accounts page. The accounts page is used to add or edit bank accounts. By default, there are no accounts. You can click on the "+" in the right corner to add a new account.



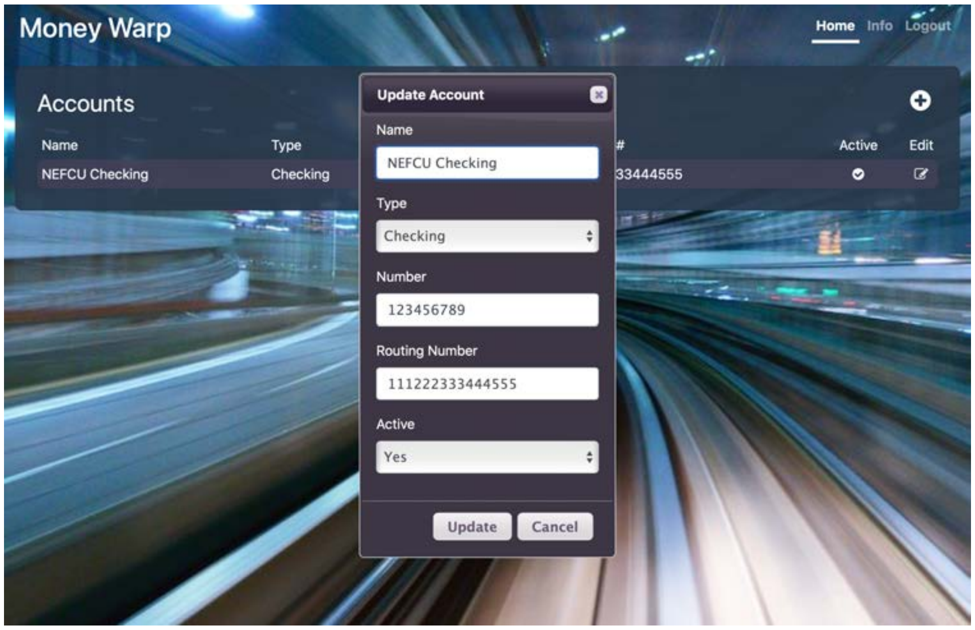
Once you click on the "+" you will be presented with a form to add a new account. Filling out the form and clicking "Add" at the bottom will add a new account to the system. If any of the fields contain invalid data a warning will be displayed and you would have to fix any issues before you are allowed to proceed.



Accounts

Name	Type	Account #	Routing #	Active	Edit
NEFCU Checking	Checking	123456789	111222333444555	✓	

If there are no errors, the newly added account will be displayed.



Update Account

Name: NEFCU Checking

Type: Checking

Number: 123456789

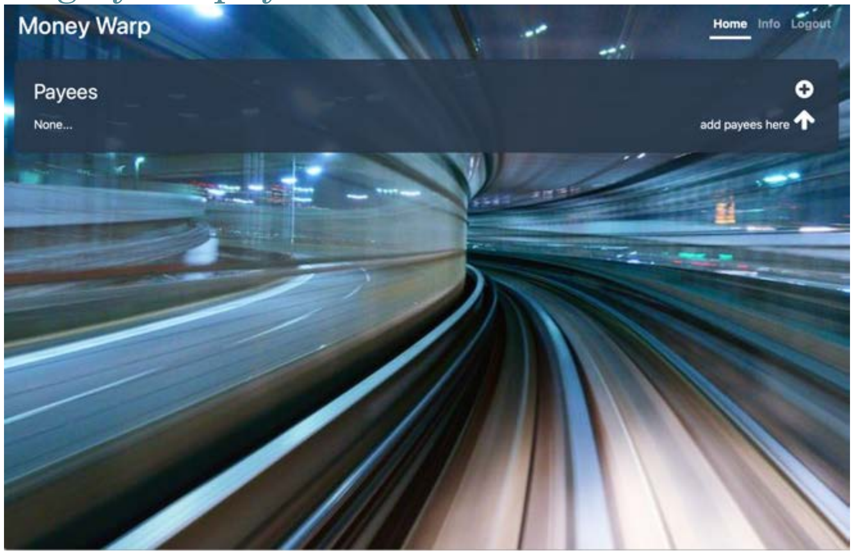
Routing Number: 111222333444555

Active: Yes

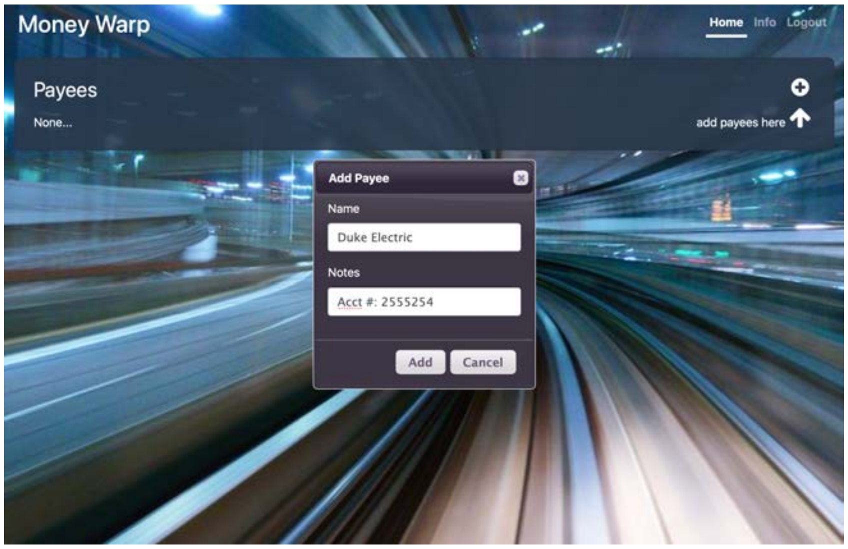
You can also edit an account by clicking on the pencil icon to the right of any account in the list. You can make the account active/inactive update the name, account type, account number and/or routing number. Just change what you want and then click “Update”

Multiple accounts are allowed, simple click on the “+” to add each of your accounts.

Payees | Manage your payees



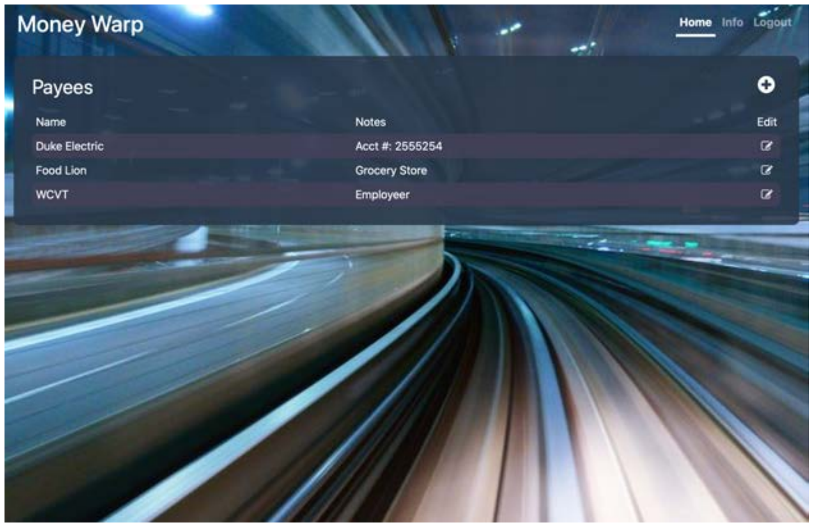
From the home screen click on “Manage Payees” to access the payees page. Payees are people or entities you get or give money to. For example your employer, name of your electric company, your grocery store, whoever you get or give money to would be a payee. The Payees work much like the accounts page. By default there are no payees, click on the “+” to add a new payee.



Name: Duke Electric

Notes: Acct #: 2555254

Fill out the form and click “Add”. The notes field can be used for any additional information you may want to store about this payee.

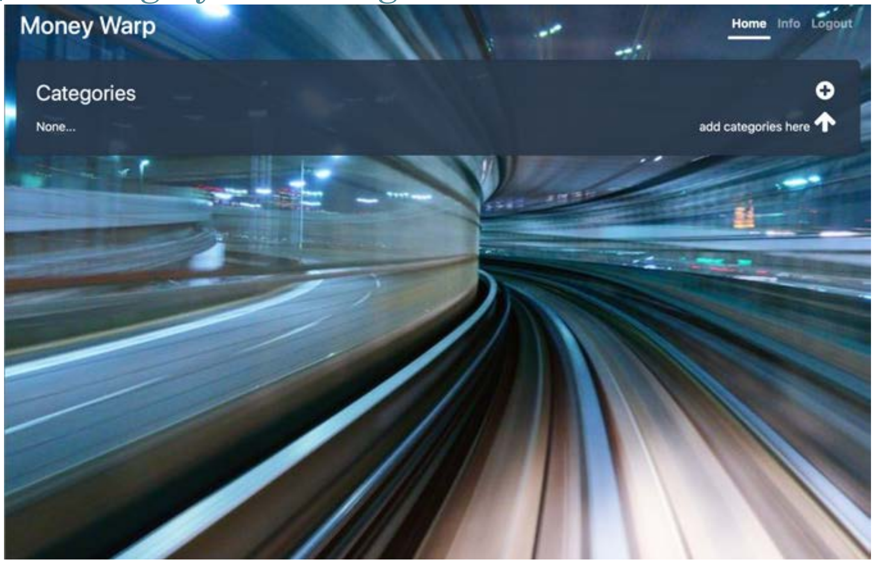


Payees

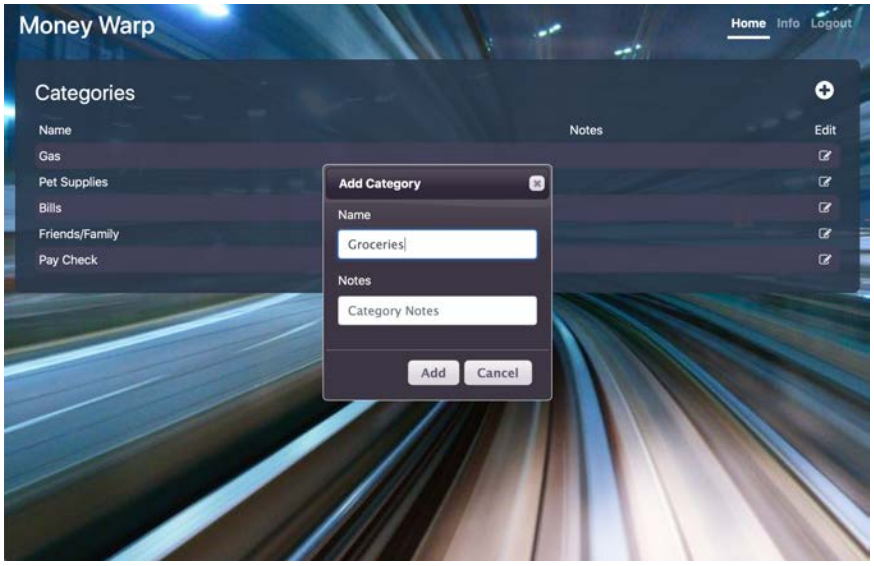
Name	Notes	Edit
Duke Electric	Acct #: 2555254	
Food Lion	Grocery Store	
WCVT	Employeer	

Just like accounts Payees can have multiple entries, and you can edit a payee by clicking on the pencil to the right of a payee.

Categories | Manage your categories



From the home screen click on “Manage Categories” to access the categories page. Categories are a way to group payments together by a common category. For example, Income, Food, Gas, Electricity, Taxes, Pet Supplies, etc. The Categories work just like the payees page. By default there are no categories, click on the “+” to add a new category.



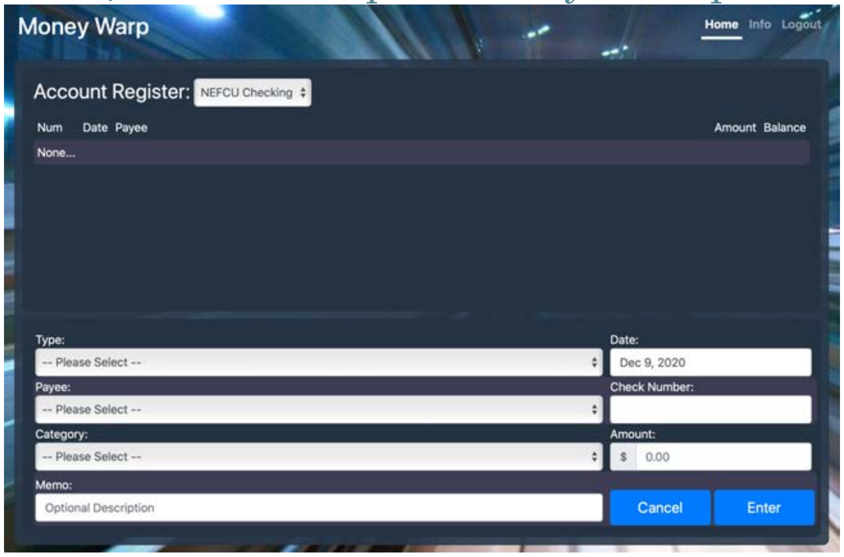
Name	Notes	Edit
Gas		
Pet Supplies		
Bills		
Friends/Family		
Pay Check		

Name: Groceries

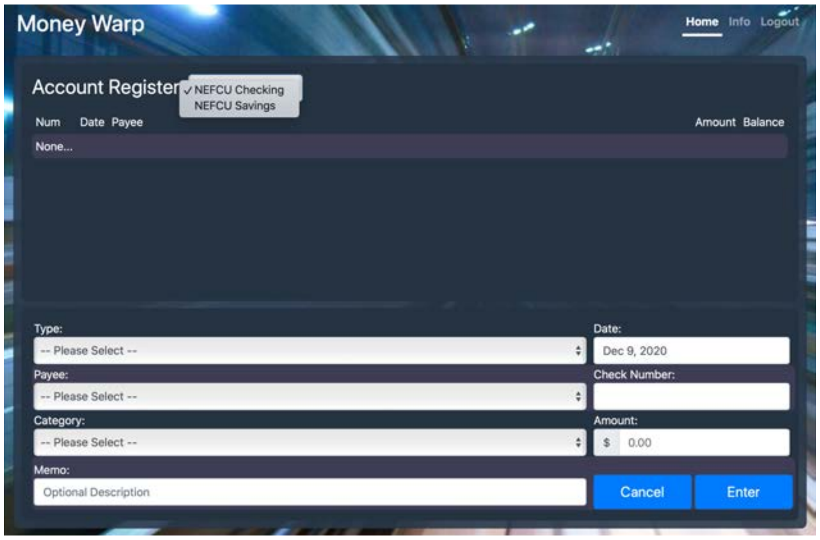
Notes: Category Notes

Fill out the form and click “Add”. The notes field can be used for any additional information you may want to store about this category. Just like accounts and payees, categories can have multiple entries, and you can edit a category by clicking on the pencil to the right of a payee.

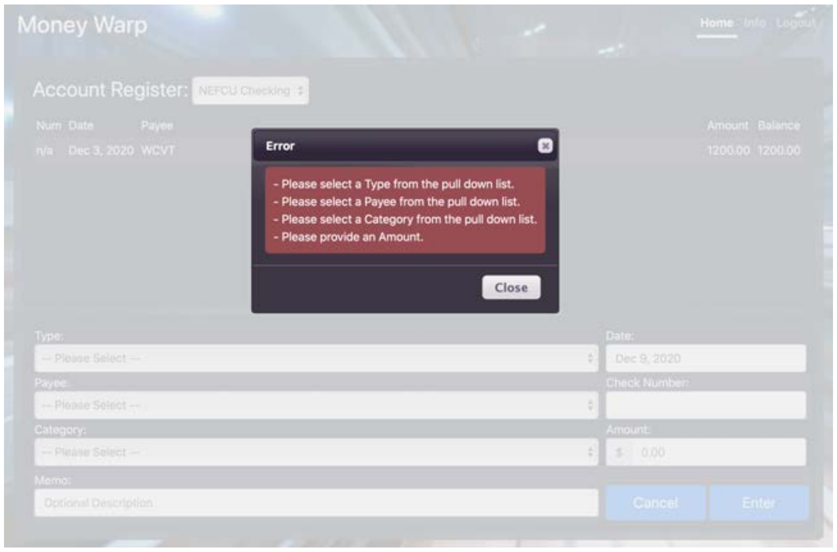
Account Register | Used to keep track of your expenses and Income



From the home screen click on "Account Register" to access the account register page. The account register page is akin to managing a check book. You would use this to add new deposits or debits and keep track of your accounts balance. The account register page has a built-in form for easily adding/editing register entries.



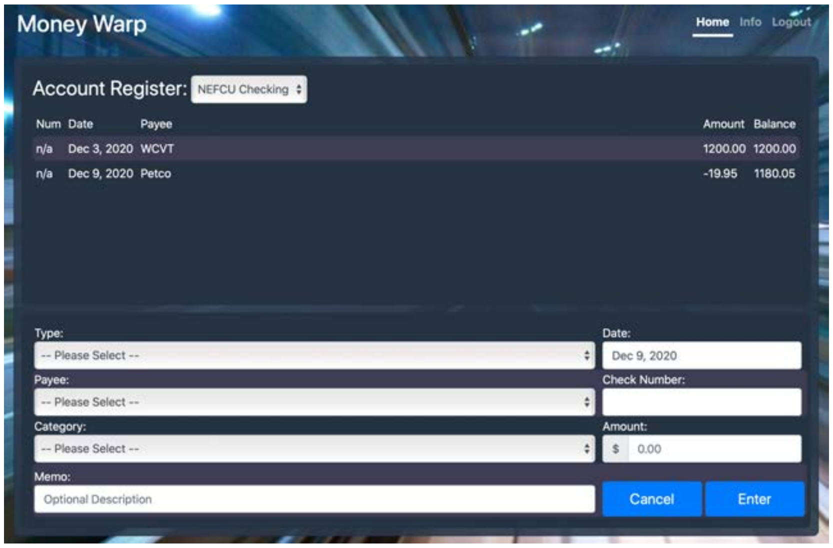
You can select which account you want to view and manage by changing the account in the pulldown menu at the top. Upon choosing a different account the register will refresh and show the information for that account.



Error

- Please select a Type from the pull down list.
- Please select a Payee from the pull down list.
- Please select a Category from the pull down list.
- Please provide an Amount.

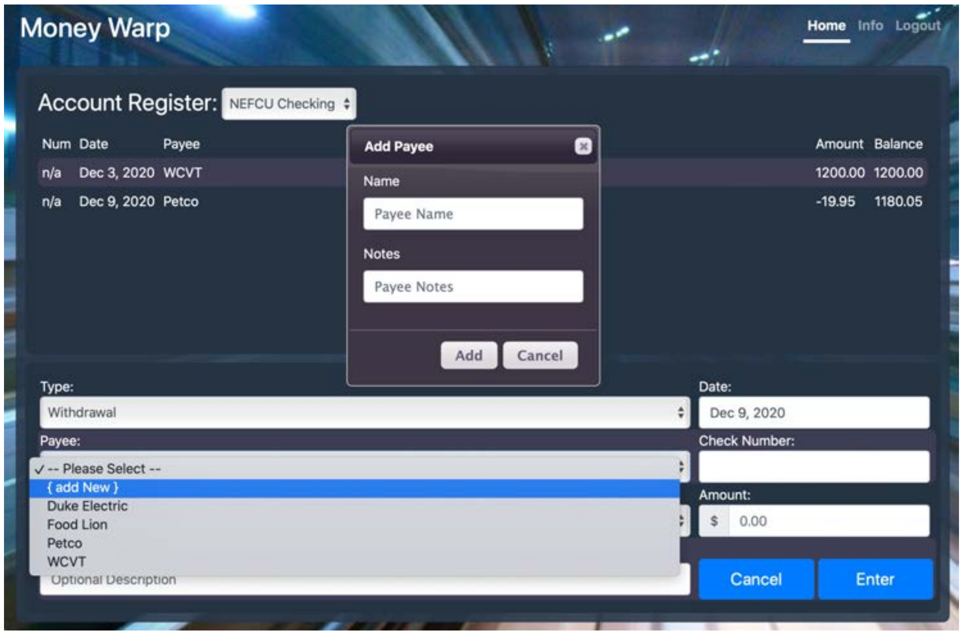
To add a new entry to the register, fill out the form at the bottom and click on “Enter”. If there is any missing or invalid data, you will need to correct it and you will not be able to add the entry until all errors are resolved.



Num	Date	Payee	Amount	Balance
n/a	Dec 3, 2020	WCVT	1200.00	1200.00
n/a	Dec 9, 2020	Petco	-19.95	1180.05

Type: -- Please Select --	Date: Dec 9, 2020
Payee: -- Please Select --	Check Number:
Category: -- Please Select --	Amount: $ 0.00
Memo: Optional Description	

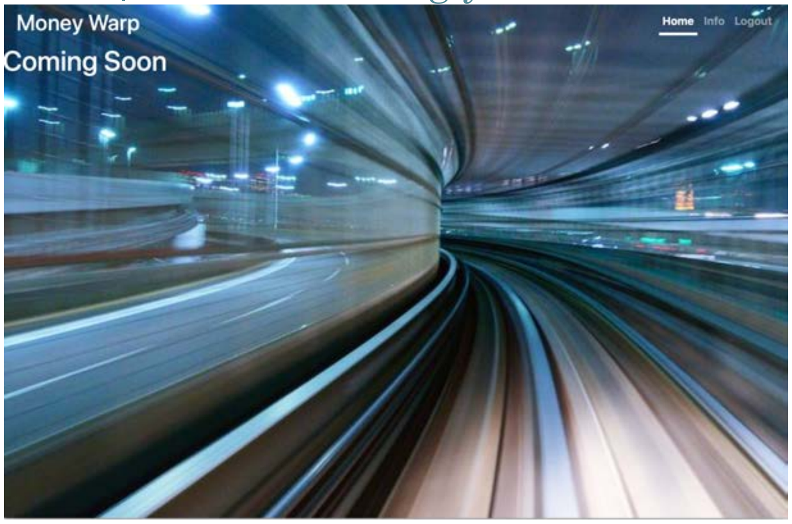
As you add entries to your register the system will show you a running account balance. You can edit a registry entry simply by double clicking on it. It will populate the bottom form and you can change what you need to and hit the “Enter” button and as long as there are no errors it will update that entry, you can also hit “Cancel” if you change your mind.



Num	Date	Payee	Amount	Balance
n/a	Dec 3, 2020	WCVT	1200.00	1200.00
n/a	Dec 9, 2020	Petco	-19.95	1180.05

If you are paying someone new, or want to add a new category, there is no need to leave the account registry screen, simply select the “add new” option under payee or category, and you can easily add the new entity.

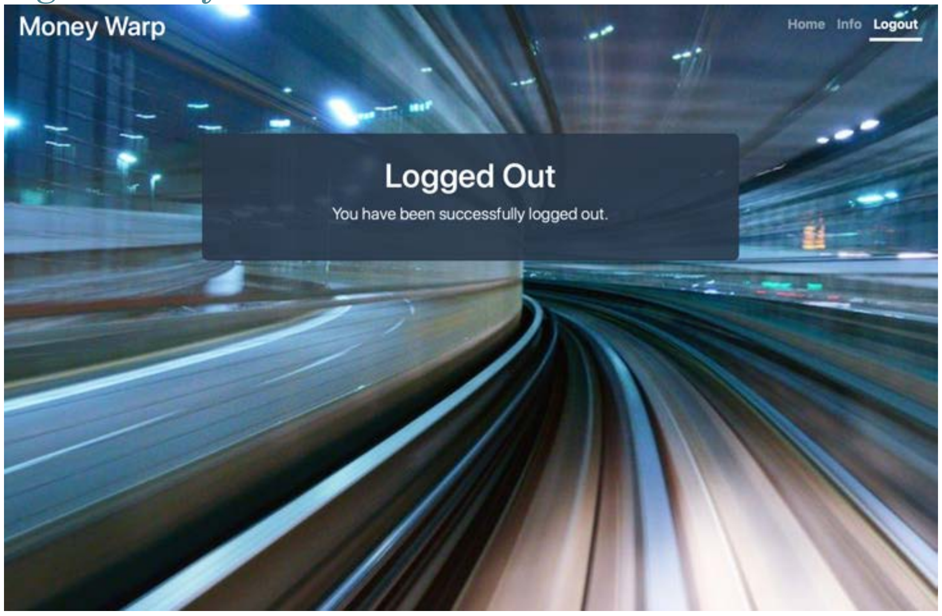
Warp My Accounts | Future Casting your accounts



Coming Soon

From the home screen click on "Warp My Accounts" to access the Money Warp function. The money warp functionality will be added in phase II. Once phase II is released you will be able to enter your reoccurring deposits and withdrawals, and then "advance" time into the future and see what all your account balances would look like based on your reoccurring bills and deposits.

Logout | Logout of system



Logged Out

You have been successfully logged out.

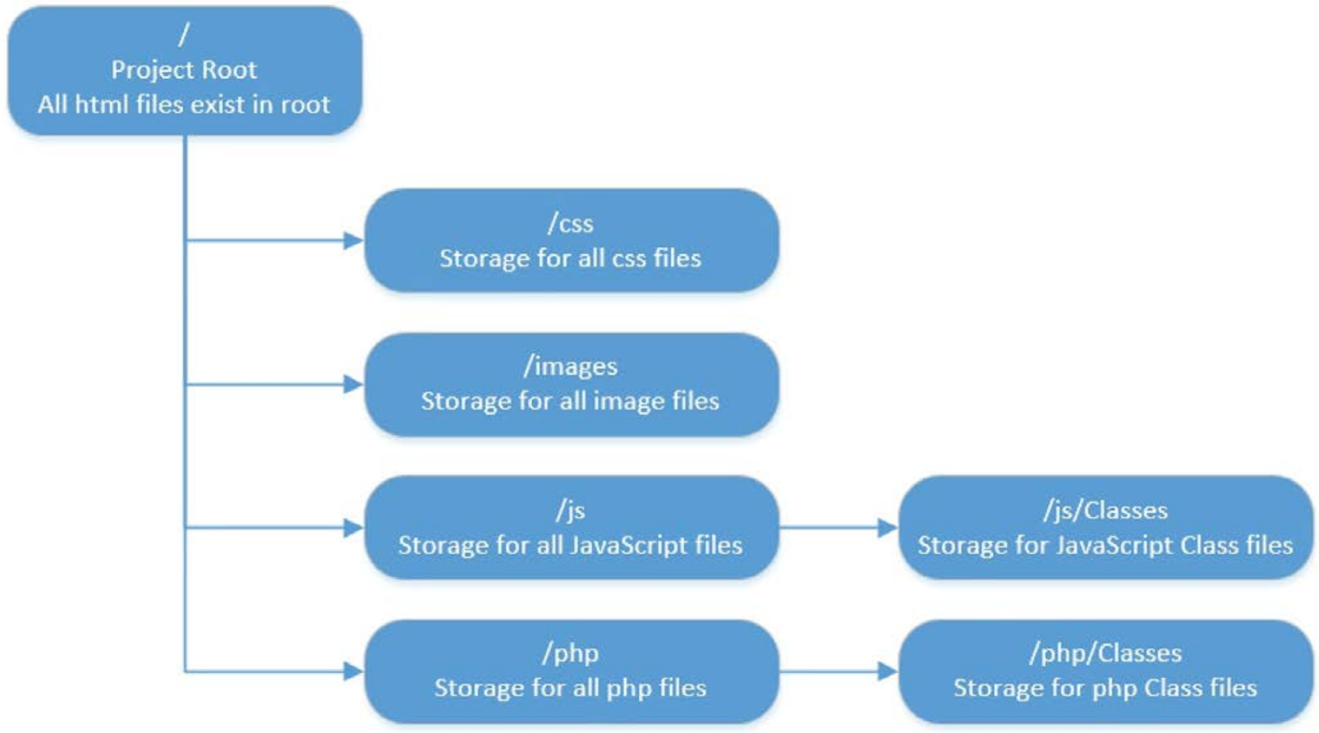
Clicking on the Logout menu item will securely end your session and log you out of the system. In order to access your accounts again, you will need to log back in.

Reports

Money Warp doesn't currently include any reporting functionality.

System Architecture

I designed the architecture to be easy to organize and find specific files for easy and clear development and maintenance.


/
Project Root
All html files exist in root
/css
Storage for all css files
/images
Storage for all image files
/js
Storage for all JavaScript files
/js/Classes
Storage for JavaScript Class files
/php
Storage for all php files
/php/Classes
Storage for php Class files

Under the "root" directory lives all the HTML files, and several directories. The css directory contains any and all cascading style sheets for this program. Similarly, the images directory contains all images for the project, the js directory contains all the JavaScript, and the php directory contains all the php files. Under the js and php folders, there is an additional folder named Classes, this folder contains all the program Classes for JavaScript and PHP respectfully.

Source Code Structure

Source code structure introduction. The following is a summary of the source code directories and their contents:

Code Directory	
Directory	**Usage**
/	This is the root directory and it contains all the static HTML files, README.md and LICENSE information.
/css	Contains the static cascading style sheet file(s).
/images	Contains any images used in the program.
/js	Contains all the JavaScript programming. JavaScript is a client-side program hence it is executed in the client's browser and can be manipulated.
/js/Classes	Contains the reusable JavaScript Classes. These classes are included and used on many different JavaScript pages within the project.
/php	Contains all the php code for the project. PHP is server side code and cannot be manipulated by the client. All error checking and input validation is done in the php code.
/php/Classes	Contains the reusable php Classes. These classes are included in many of the php files to handle things like Database Access, Error reporting, Debug Logging, and Session management.

Highlighted rows indicate directories containing source code.

Executables

This is a PHP/JavaScript Web Application. The Executables consist of PHP files that validate input, process changes, interact with the database, and handle session management.

includePHP.php (/php/includePHP.php)

This php is used to dynamically include all the program PHP Class files in the /php/Classes/ directory. By calling this one program from any other php program all the program classes will be available to that php.

config.php (/php/config.php)

This php contains all the dynamic variables needed for the application to work. It contains the database host, name and password, along with system email addresses. An installer would modify this file for their specific environment.

login.php (/php/session.php)

This php is used to login a user if proper credentials were provided.

session.php (/php/session.php)

This php works with the Sessions Class and can either validate if there is an active session or terminate a session.

accounts.php (/php/accounts.php)

This php handles the management of accounts. Depending on how it is called, it can do any of the following: add a new account, update an account, or retrieve account(s). All input validation for new entries are handled here.

categories.php (/php/categories.php)

This php handles the management of categories. Depending on how it is called, it can do any of the following: add a new category, update a category, or retrieve a category or categories. All input validation for new entries are handled here.

payees.php (/php/payess.php)

This php handles the management of payees. Depending on how it is called, it can do any of the following: add a new payee, update a payee, or retrieve a payee or payees. All input validation for new entries are handled here.

register.php (/php/register.php)

This php handles the management of account registry entries. Depending on how it is called, it can do any of the following: add a new registry entry, update an existing registry entry, retrieve registry entries, get the dynamic form entries, get the last active account, and update the last active account. All input validation for new entries are handled here.

Code Architecture

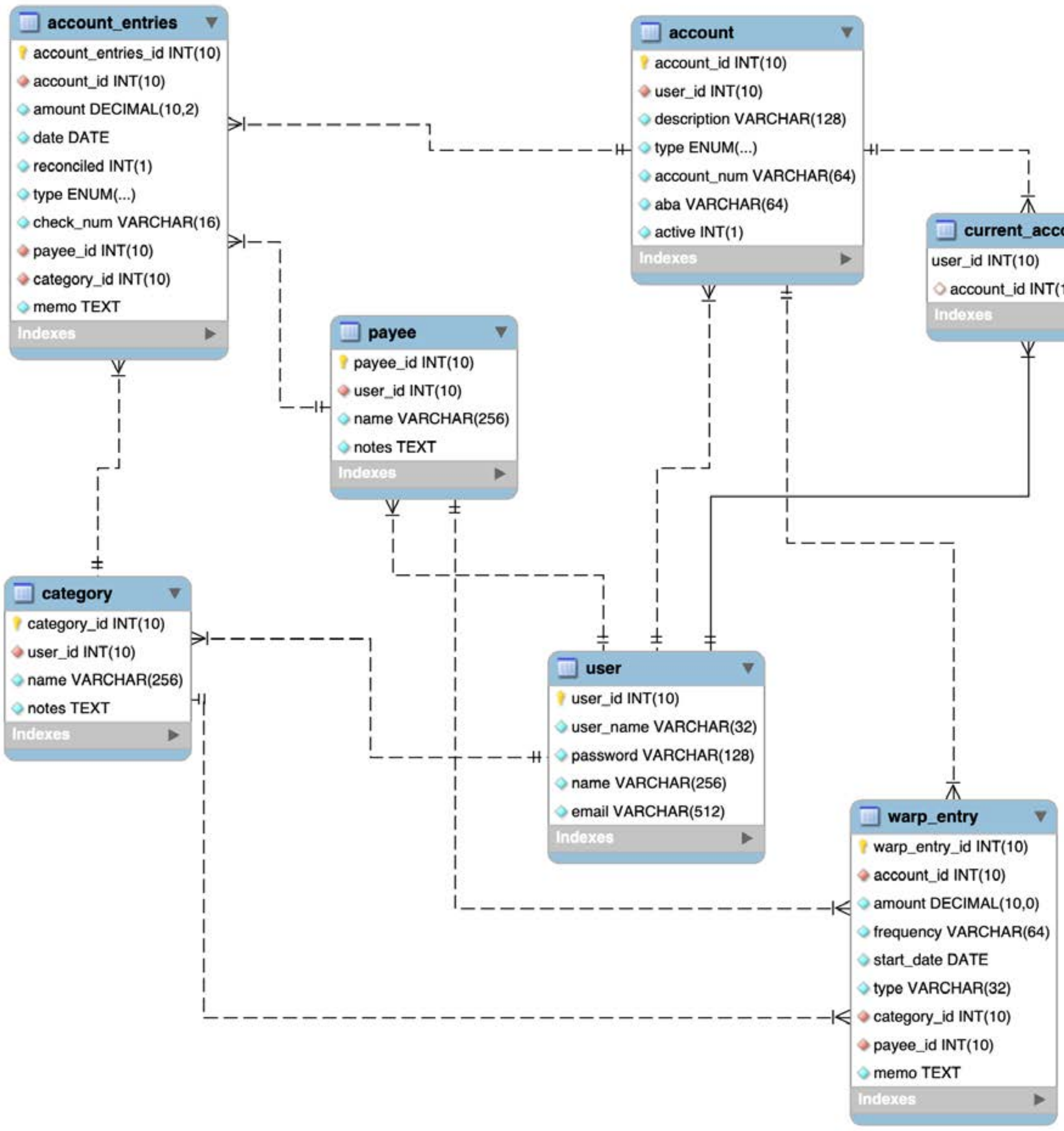
The html is the means of accessing the program. The html includes bootstrap, jQuery, jQuery UI, and that specific html's custom JavaScript file. Each html file has a corresponding JavaScript file, normally named after the html file. For example /payee.html includes /js/payee.js. The html file itself is responsible for the general layout of that screen, and then the JavaScript file handles presenting the dynamic information onto that screen and or manipulating the screen.

The JavaScript programs work with the php files to retrieve and send information to the server. Since JavaScript is executed on the client's side, and any validation done there could be circumvented, no data validation is done via JavaScript. Php on the other hand is executed on the server and validation performed there cannot be circumvented so php handles all input validation.

After input validation, the php handles all interactions with the database, adds, updates, and retrievals. Data is then sent back to the JavaScript files for dynamic display or further manipulation.

Database or Data Store

The database is centralized around the user table. Everything is either directly or indirectly tied to a user. The account, payee, and category tables have a many to one relationship with the user table. The account_entries table has a many to one relationship with the account, payee and category tables. The warp_entry table has a many to one relationship with account, payee, and category tables. The current_account table is used to store the last used account and has a many to one relationship with the account and user tables. The database structure is all relational. If you delete a user, it will cascade through all the tables and remove all data associated with that user. On the flip side you cannot add an entry into the other tables without the corresponding user entry etc. These relational relationships assist in keeping data integrity across the system.



Views, Stored Procedures and User Defined Functions

My database does not include any views, stored procedures and/or user defined functions.

External Files & Data

Data for this application is stored in a MariaDb or MySQL server. Access to the database should be restricted to only necessary personal. The php, html, JavaScript, css, and config files live on an apache server. Edit access to these files should be limited to only necessary personal.

Programming Language | php, JavaScript, jQuery

This program is an accumulation of php and JavaScript. The JavaScript portion is assisted by the external library jQuery.

The php portion of the program does all the heavy lifting. Php is responsible for directly accessing the database, data validation, data manipulation, receiving and returning data to the application.

JavaScript assisted by jQuery handles all the sending, receiving, and displaying dynamic content for the program. jQuery is used mostly for its cross browser functionality and integration. jQuery also as a lot of built in functions and procedures that are missing in the core JavaScript framework.

Project Classes

Classes within the project are used to abstract re-usable pieces of code. Classes are also used to group related values, known as properties. The project utilizes two different type of classes, JavaScript Classes and php classes:

Database Access | /php/Classes/MyDatabase.php

This is a php class for connecting to the database using the /php/config.php variables. It provides a consistent way to access the database via the returned database handler.

Error Message Management | /php/Classes/MyErrorMsg.php

This is a php class for holding any error conditions that the php may find. You can dynamically add error messages, check to see if any errors exist, and return all error messages concatenated via a delimiter of your choosing into a string for easy display.

Debug Logging | /php/Classes/MyLog.php

This is a php class used to write out messages to the php error log. This function can be very handy to debug json return items where you can't output anything to the screen because it will disturb the json response. Instead, this will take any php variable and display its contents in the php log file.

Json Return | /php/Classes/MyReturnJson.php

This is a php class for returning dynamic php data to the browser encoded in json. This class handles sending the necessary browser headers, encoding the php data into json, and sending that json data to the browser, which is received and decoded by JavaScript.

Session Management | /php/Classes/MySessions.php

This is a php class for securely managing php sessions. It handles creating, checking, and returning session information. It regenerates session ids and prevents session fixation, by requiring two forms of identification. The session id itself and then a separate unique cookie. The cookie is provided by the server only after a successful login. Any attempt to access the system authenticated without the additional cookie will be rejected.

Dynamic Popup Modal | /js/Classes/MyAttention.js

This is a JavaScript class that is used for displaying a pop-up modal box within the program. It is used to dynamically display various bits of information. It has built in options for Error, Warning, Informational, and generic content display. It also appends the necessary html to the page when initialized for easy and proper usage.

Session Management | /js/Classes/MySessions.js

This is a JavaScript class that is used to verify sessions status. This class is used on every page that can only be accessed after you have logged in. It integrates with the session.php file and returns if there is an active session or not. It also includes a session log out function.

Waiting Indicator | /js/Classes/MySpinner.js

This is a JavaScript class that is used to dynamically add a waiting indicator to any html page. This is like the modal class; in that it adds the required html to the page it is initialized on. Once initialized the waiting icon or “Spinner” can be shown or hidden with ease.

Project Modules

The modules in this project would be the procedural php files that do data verification and database access.

accounts.php | /php/accounts.php

Used for data verification and database add, update and retrieval for accounts.

categories.php | /php/ categories.php

Used for data verification and database add, update and retrieval for categories.

payees.php | /php/ payees.php

Used for data verification and database add, update and retrieval for payees.

register.php | /php/ register.php

Used for data verification and database add, update and retrieval for register entries.

Program Start and End Flow

A user accessing the program would be initially present with the login screen. The user has the option to log in or click on the Info page.

The info page show static info, and from there the user has access to click back on home to log in.

Once a user logs in, they are presented with five options, plus the Info, and Logout Pages. The Five options are: Warp My Accounts, Account Register, Manage Accounts, Manage Payees, and Manage Categories.

Warp My Accounts: Will be implemented in phase II. Clicking on home will return you to the main menu.

Account Register: Displays registry entries and allows for the adding and editing of account registry entries. From the payee drop down, you have an option to adding additional payees. From the category drop down you have the option of adding a new category. Clicking on home will return you to the main menu.

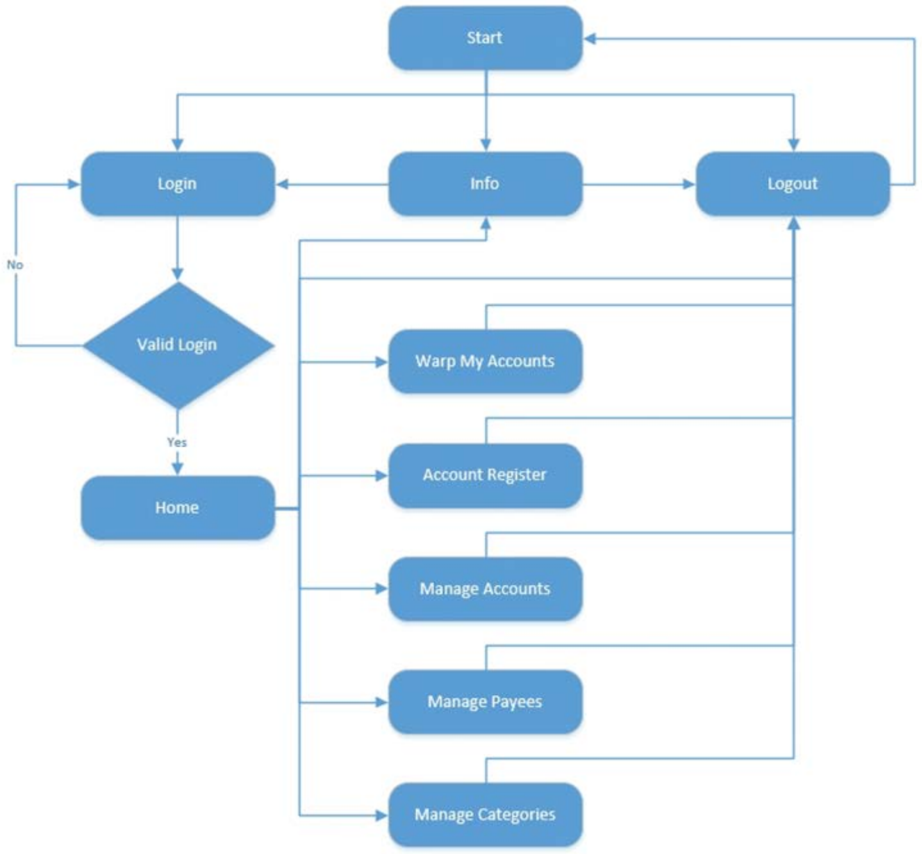
Manage Accounts: Allows for the addition and editing of accounts. Clicking on home will return you to the main menu.

Manage Payees: Allows for the addition and editing of payees. Clicking on home will return you to the main menu.

Manage Categories: Allows for the addition and editing of categories. Clicking on home will return you to the main menu.

Logout: Will log you out of the application and terminate the session destroying any temporary items.

Diagram



Summary

This project is written using php, JavaScript, and html utilizing a MariaDb database for long term storage. It is intended to be installed on a webserver running apache 2.4.6 or later, php 5.4 or later, and MariaDb or MySQL server 5.5 or later.

The project was designed using a naming convention to group items together. The directory structure was designed with folders that are named for what they contain. The php folder contains all php files, the js folder contains all JavaScript files, the two Classes folder under php and js contain the program classes for those programs.

Each item or page in the application has a .html, .php, and .js file that all work in unison with each other. The html file loads the JavaScript file, which in turn calls the php files.

The html files are used for general layout, the JavaScript files are used to send data to and retrieve data from the php files, and to display that data dynamically inside the structure the html file provided. JavaScript is also used to maintain the applications display based on different screen sizes. The php files do all the data verification and database interaction.

Where appropriate program classes have been utilized to handle more complex reused items. The database is relational to maintain and force data consistency.

Additional external resources were used to assist in the development and functionality of the application. These include Bootstrap, jQuery, jQuery UI, and FontAwesome. Bootstrap was used for general display layout and css styles. jQuery and jQuery UI were used for dialog boxes, cross browser support, ajax, and json parsing and processing. FontAwesome was used for professional scalable icons.

Credits & References

"Mobile Has Thrust the Purchase Funnel into Warp Speed." Digiday, 9 June 2015, https://digiday.com/marketing/mobile-thrust-purchase-funnel-warp-speed/.

APPENDIX B (BUILD AND RELEASE PROCESS)

Since php, html, and JavaScript are all scripting languages that are not compiled, updates are as simple as modifying the necessary code and saving the changes to the server.

APPENDIX C (CLIENT INSTALLATION INSTRUCTIONS)

The client would access the application via any modern web browser by simply typing in the web location url or clicking on a link provided by an administrator.

APPENDIX D (DEVELOPER SETUP INSTRUCTIONS)

This application was designed to be implemented on webserver running apache 2.4.6 or later, php 5.4 or later, and MariaDb or MySQL server 5.5 or later.

To install this application provision a webhosting site with the above minimal requirements using a https secure site. Provision a MariaDb or MySQL server with full read/write access to a database. Edit the /php/config.php file and enter the required site information. Upload all the files to the root of that webhost site, and then import the moneywarp.sql file into the database to provision the database. Test site access.