Family Financial:

The Final Project

Conner Getchell and Alex Greer

Professor Zhonghang

CS 396: Intermediate Software Project

December 1st, 2019

**Production Description**

The project that was assigned to build a Django based website that can allow users to interact with a DBMS via a web interface. The DBMS has to support systems for users to manage income, spending, investments, debt, and of course, budget planning. The system is capable to support users of several families as well as multiple users per family which then can be read to create intuitive bar charts and pie charts to interpret the data.

**Implemented Functions**

The required functionality of the site varies. Based on the previous projects that integrated different systems into Django, we are now combining all of those efforts into one project that will support the following implemented functions:

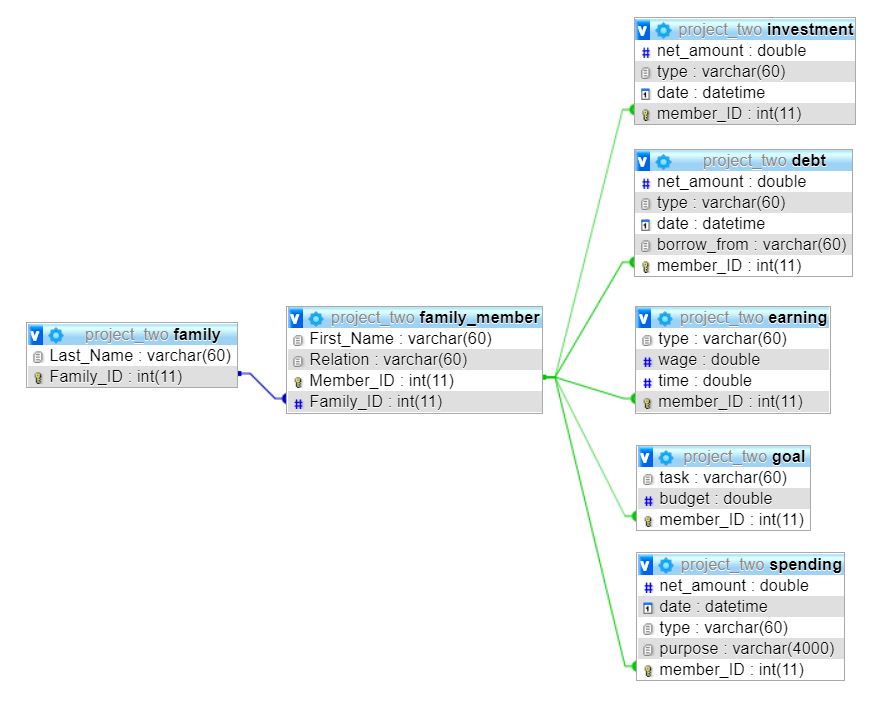
* The user can post a memo about income, spence, or travel plans, and all family members can discuss by replying to their comments.
* The user can query a family member’s name, income amount and its source (a job, awards, or investment, etc) for a given year.
* The user can query a family member’s spending on entertainment for a given month and show the total for the whole year.
* The user can upload a data file (e.g. historical data of a family’s income) to the system.
* The system can provide the user interface to display a plot of a family income in the past 10 years through the browser.
* Use bar chart to show the performance of a family’s investment in the past 10 years. This figure shows the return rate vs year over a 10-year period. The return rate is computed by the amount difference between two years divided by the previous year’s amount. Suppose that the initial funding is $10,000.
* Predict the amount of the user’s investment account next year.
* Compare the ways of spending in the past two years. Use two bar charts side by side for comparison. Each chart shows spending type vs amount (food, entertainment, travel, sport, education, etc, at least five types).

Upon meeting those requirements, we found that our web interface must be intuitive enough that any user can visit the site and find and demonstrate proficiency in all of these features will no training or explaining.

**Technical Details**

The project was constructed using the Django web framework. Within the views.py of the finance application is the majority of the unique code. The views is where the Chartit library calls and renders the information from the DBMS (SQLite 3). The rendering is then passed to the template where it displayed onto HTML with only three lines of code per template.

The following Figure #1 below outlines the relations of the tuples I decided to make for this DBMS. The “family” table contains the Family\_ID which acts as the

Figure #1

primary unique key. The “family”\_member” uses the Family\_ID as the foreign key to specify a specific user to the family or families that they belong to. The primary key of the family\_member is the Member\_ID. The Member\_ID is used as a key in the remaining tables (investment, debt, earning, goal, and spending) to attribute their values to a specific user and subsequently a family.

The site is additionally able to parse a user inputted CSV file. The allows the functionality of users to use a program such as Microsoft’s Excel to work on financial processing and then use the site for planning. The parsing was implemented using Python parse CSV function.

**Additional Features**

By using Chartit, I was given some additional features that I was able to pass onto the end-user. Firstly, the user is able to hover their mouse of the chart to more clearly reveal information being displayed on the chart. Secondly, users are able to view the chart in the full screen if they chose to do so. Thirdly, the users are able to download the specific chart they want in varying file formats: PNG, JPEG, PDF, and SVG. Lastly, the chart can be downloaded as a CSV file inorder to be used in the excel sheets.

**Final Points**

Django overall has been a fun experience for the both of us. Using the web framework over more traditional methods has been a struggle. The learning curve to using Django for the both of us was not easy. In the end, we both believe that the Django web framework is a helpful and powerful tool, given the knowledge to use it. Many hours were wasted researching how to set up accounts properly, integrate charts, convert DB management systems, and many more small things.

Using Chartit was a blessing find. Chartit was a Javascript library that an individual had converted to be used in Python. Chartit uses highcharts API to create beautiful and animated charts. Alternatively we could have used D3.js. D3 we have found in the past is extremely complicated and not very intuitive to use.