J-HEX

Eric Li, Xing Tao Shi, Jacqueline Woo, Henry Zheng
SoftDev pd8
Project02 -- The Final Frontier
2018-05-15

Description

J-HEX presents Clover, a financial manager where users will be able to input data about their finances, savings, and expenditures. The user will be able to see a history of their past spendings and set budgets for their future expenditures. Users will also be able to clearly see breakdowns of their expenditures and projected savings via graphs and charts generated by d3. Another feature of Clover is that users will be able to track and manage their stocks. They will be able to easily see the profit they make from their stocks via d3 charts and search up new stocks to add to their portfolio.

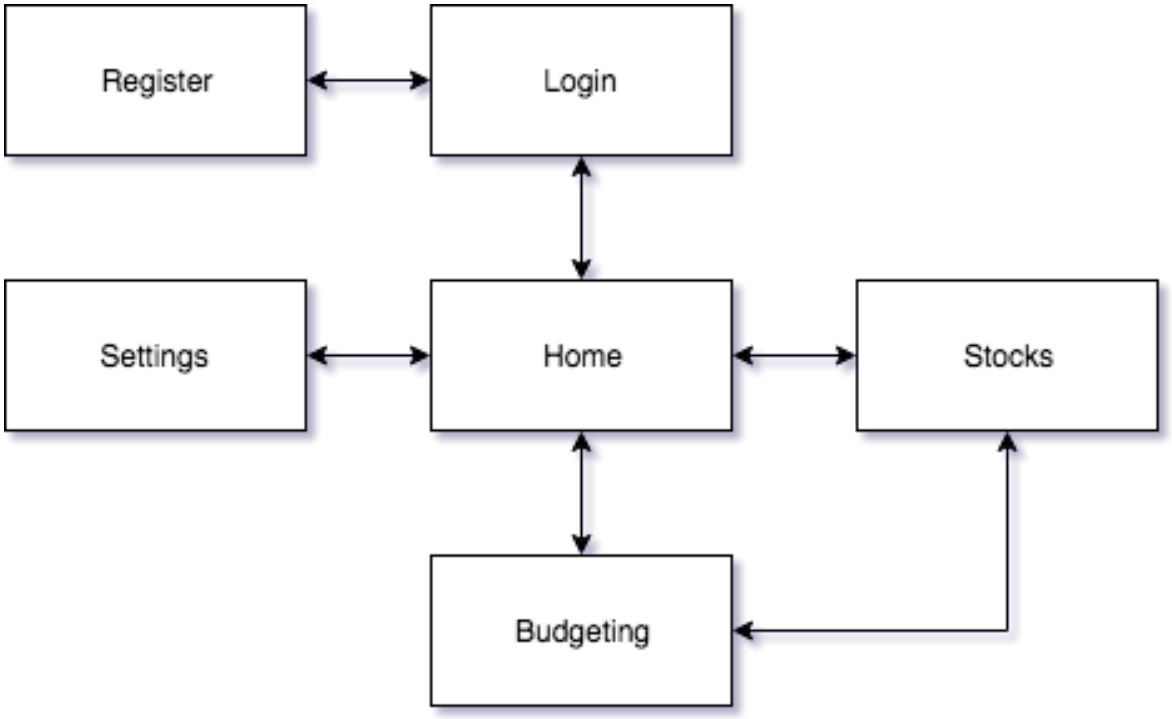
API(s)

Alpha Vantage - JSONs of real time and historical stock and crypto data

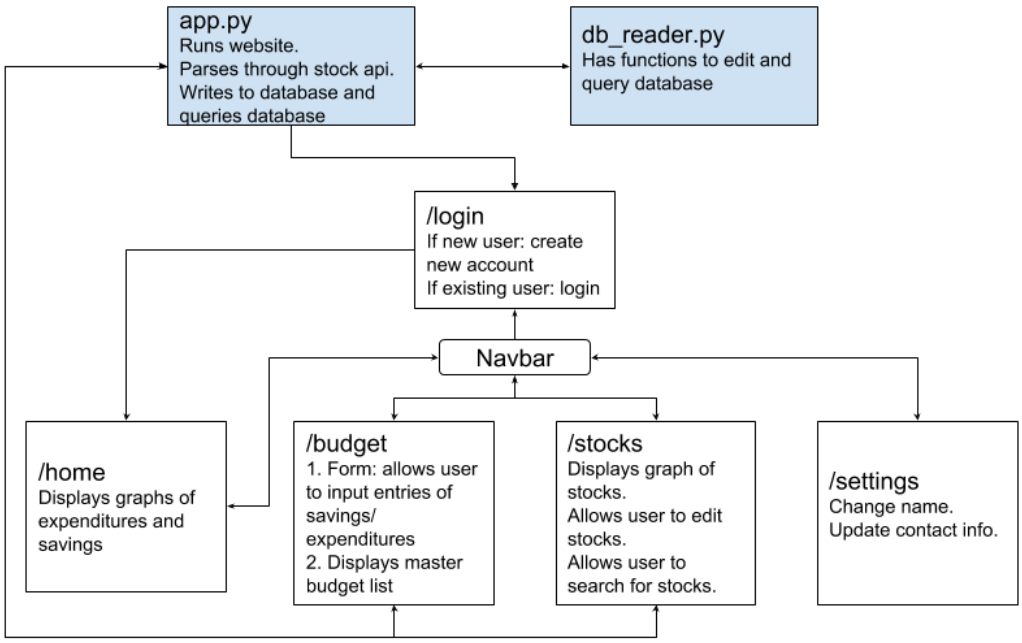
Stages of Development

1.	Setup login and register sites	5/16
2.	Setup database for user portfolios	5/18
3.	Create ability to submit spending and income	5/22
4.	Display spending & savings	5/24
5.	Create ability to buy/sell stocks	5/29
6.	Display stocks	5/31
7.	Display projections of savings and stocks	6/4
8.	Make pretty	6/11
9.	Buffer period	6/13
10.	Profit	6/15

Site Map



Components



HTML

- home.html
    - Displays charts/graphs of :
        - How much user spends each year, month, week (default is monthly). User can choose to compare it to his/her budget.
        - Savings and income for each year, month, week (default is monthly)
        - User's expected income for the next two or three months based on income, expenditures, stocks, and dividends.
        - The percentage of money user spends for food, clothes, housing, education, etc.
    - Displays breakdown of all charts and graphs
- stocks.html
    - Displays graph of loss/gains from stocks
    - Displays chart of stocks you own, including how many stocks and the worth of each stock
    - Can edit portfolio
    - Search bar to search market stocks
- budget.html
  There will be a toggle button at the top of the page to toggle between the form and master budget list. The form allows users to input entries into the database
    - Form fields
        - Input type (savings, spendings)
        - Description of input
        - Amount
        - Duration of saving/spendings
            - Ex. user gets paid every month, user spends money on food every week

  The master budget list will display all the inputs the users has put in. It can be sorted by type (saving/ spendings) and amount. When the master budget list is toggled, the user can also edit past entries.
- settings.html
    - Change name
    - Contact info
- register.html
    - Can register for account or log in

CSS

All pages have navbar

- home.css
    - d3 heavy for displaying graphics
- stocks.css
- budget.css
- settings.css

Python

- app.py
  - get_api()
    - Gets api information and parses it
  - parse_api()
    - Parses api stock information
    - Returns stock information
  - home()
    - Runs home.html
  - stocks()
    - Runs stocks.html
  - search_stocks(stock_name)
    - Returns list of stocks by stock_name
  - budget()
    - Runs budget.html
  - settings()
    - Runs settings.html
  - verify_user()
    - Verifies user
  - set_name()
    - Sets name of user
  - set_pass()
    - Encrypts and sets password of user
  - create_new_user()
    - Creates new user

- db_reader.py
  - get_stocks()
    - Returns list of stocks
  - add_stock()
    - Adds element to stock list
  - remove_stock(id)
    - Removes stock by id
  - get_budget_master()
    - Returns budget master list
  - add_budget_master()
    - Adds element to budget master list
  - remove_budget_master()
    - Removes element from budget master list
  - get_spendings()
    - Returns list of spendings
  - calc_spend_percentage()
    - Calculates the percentage the user spends on food, goods, services, etc

- get_savings()
  - Return savings
- add_savings()
  - Adds element to savings
  - Recalculates total savings
- get_income()
  - Returns income
- set_income()
  - Sets income
- calc_proj_income()
  - Calculates projected income for next three months

Databases

users

INT userId	STRING user	STRING pass	STRING name

- userId = user id, used for easy reference to an account
- user = username to login
- pass = salted password
- name = name of user

money

INT userId	FLOAT currentMoney	FLOAT monthIncome	FLOAT otherIncome	FLOAT savings

- currentMoney = current amount of money user has
- monthIncome = monthly income of user, automatically adds to currentmoney
- otherIncome = other income the user may have, resets every month
- savings = total savings

fixedCosts

INT userId	INT id	STRING fixedName	FLOAT fixedAmt	STRING fixedDesc

- fixedName = name of the fixed cost (rent, bills etc), user created
- fixedAmt = monthly cost of fixed expense, automatically deducted
- fixedDesc = description, user's description of the fixed cost

variableCosts

INT userId	INT id	STRING expName	INT expType	FLOAT expAmt	FLOAT expBud	STRING expDesc	STRING date

- expName = expenditure name, user creates it
- expType = type of expenditure (food, leisure, etc.), user selects the type; referenced as by an INT id
- expAmt = expenditure amount, user entered
- expBud = expenditure budget, user sets how much they wanted to budget
- expDesc = description, user's description of the expense

stocks

INT userId	INT id	STRING ticker	INT shares	STRING purchDate	FLOAT purchPrice

- ticker = stock abbreviation
- shares = number of shares
- purchDate = date of stock purchase
- purchPrice = purchased price of stock, generated from purchDate

Roles

Project Manager and Data Visualization – Henry
Backend – Xing Tao, Jackie
Databases / Frontend – Eric