Advanced Software Engineering
Project Proposal
Christophe Rimann, Nigel Schuster, Julian Serra, Jordan Vega
Team: Buy Bitcoin

# MIWF
## Motivating Investment - Wells Fargo

Our project will double as a project for our class, and a project for Wells Fargo's challenge.

**Synopsis**

The project will involve a web based application that will inform bank clients, i.e., Wells Fargo clients about investment opportunities, and the benefits of investing. Some of the more important parts of a functional society and growing economy are investments and savings, and in America (and most of the world) only certain niches of the demographic do this. Through our stock simulation, we would like to motivate greater investment and saving by showing clients what portfolio growth they would've had, and could have if they had decided to invest.

The product will look at user specific bank data to determine spending habits, potential industry interests, and risk quantification (for the purpose of this project and challenge, we will be using a provided database of fake user data to develop and test our product). From this, suggestions will be made of specific assets that a user could invest in, based on an analysis of potential industry interests. For example, if a user orders Domino's Pizza once a week, it would be our goal to recommend an investment in that industry.

Moreover, from the user data, we will extrapolate potential amounts that users could have invested if the client was rounding up, saving a percentage of their expenditures, or even storing a percentage of their income. From this, we would attempt to calculate the present value of what would have been an investment with the aforementioned funds, in order to show the potential portfolio growth that a client could have experienced from investing in stocks or other digital assets. Hopefully, this information will give users a what-if moment, getting them to begin to invest and save.

The project will have a natural UI with an easy flow, allowing any user to understand and explore the application. Moreover, it will be simplified in its presentation of stock data, designed for a wide audience of people that have varying experience with financial instruments. After all, the main user target is someone that is not investing, and thus, surely has little to no exposure with financial facts and figures.

Overall, the purpose of the project is to create a product to help incentivize and grow savings and investment culture. We want people to spend less on their avocado toast and make smarter investment decisions.

**Flow**

1. Login
2. Connect bank account
3. Choose risk option (or skip)
4. See recommendations and suggestions
5. Modify recommendations by choosing individual assets of interest to display, simulate.
6. (out of scope? actually perform investment according to the designated risk index).

**Users**

- *Risk Loving Individual (Young, High-Expenditure Patterns, or option selected)*
  Suggest and show history of riskier investments.

- *Risk Neutral Individual (Average Expenditure Patterns, or option selected)*
  *Suggest and show history of safe investments with significant returns.*

- *Risk Averse Individual (Safe Expenditure Patterns, or option selected)*
  Suggest and show history of extremely safe investments.

* Note that one of the goals of this project is to motivate newcomers to the market without experience to enter and participate. Newcomers may fall into any of the aforementioned 3 categories.
** We do not distinguish between different user categories. Our only category is customers.

**User Stories**

- Log in - a user wants to log in to account.
- Connect Plaid - a user connects bank data to our platform.
- Check my balances - a user looks at his/her current balances.
- Check my expenditures - a user looks at his/her total expenditures (from a specific date).
- Check my income - a user looks at their total income (from a specific date).
- Start investment - a user can begin to invest in a certain bucket of assets.
- Sell investment - a user can divest and withdraw USD out of service.
- Check my potential portfolio - A user can check and review one of multiple pre selected portfolios.
- Check my potential portfolio growth - user can interact with their portfolio growth by looking at a graph displaying said growth.
- Analyze stock bucket - A user can see what stocks he is/ would be invested in.
- Modify stock bucket - User can modify percentages and industries he would like to be invested in.
- Select funding method - User chooses how they would like to fund their account, out of 4 provided options.
- Analyze funding method value - user looks at how much money they would have saved by selecting a funding method and a specific date.

**Potential Error Sources**

- Incorrect Login Credentials
    - Solved through authentication plugin
- User selects incorrect risk identity
    - Check if user is sure about risk identity
    - Allow risk status to be changed
- Problems with user bank data
    - Ask user to review transaction history of bank data

**Tools**

Server side:

- Python 3.6.x
- Static Analysis:
    - Flake8: style checker (made up of PyFlakes, pycodestyle, and Ned Batchelder's McCabe script)
    - Pylint: style checker, error detection
    - Bandit: code smells, centered around security detection
- Testing: Pytest + snapshottest
- Web: Graphene + Gunicorn

Client side:

- Compilation: yarn task runner + babel + webpack
- Static Analysis: ESLint + Flowtype
- Framework: React + Relay (Prototype with Storybook)
- Testing: Jest + Snapshots

Deployment: Github + Travis CI + Heroku

**Architecture:**

Our Django application is split into several sub applications:

- Stocks: this application imports stocks and tracks past and present performance, for graphing purposes
- Trading: this application will actually allow users to purchase and sell stocks
- Authentication: This application centers around user authentication. There are two primary authentication processes. The first is google oAuth (the way the user authenticates with our app - user management will exclusively be through google oAuth). The second is bank authentication: the user must log in to their bank from our application so that we can process transaction data
- Web: this is the application where all primary web views that serve the application are served from
- Wf: this application pulls in wells fargo specific data
- BuyBitcoin: this is the application source, and contains settings, the url router etc.