FINEX

Team 8 Design Document

Hugh Bromund, Peyton Williams, Aditya Naik, Daniel Joseph, Nathan Ashta, Niyati Sriram

Table of Contents

Purpose

Managing your finances is an important and often time-consuming part of every adult's life. However, personal finance tools tend to be very specialized and can also be difficult to navigate. Without knowledge of how to invest their money, many people can be missing out on investing opportunities. Financial literacy in terms of managing income is also very important as it can be easy to lose track of where people are spending their money. The goal of FINEX is to make all of this information not only available, but understandable to everyone.

In addition to bringing investment and financial literacy to our users, FINEX aims to bring stock insights to our users that aren;t usually available to the general public. Day Traders and the strategies of small time investment managers can be difficult to learn without the right tools and education. We hope to provide a new avenue for motivated investors to learn more about how to interpret market movements.

The purpose of this project is to create an easy-to-use tool that allows the user to track various aspects of personal finance. These aspects include not only budget and savings management tools, but also investment information through stock market tracking. This will be different from existing or similar services due to its navigable UI as well as its combination of existing services into one application. It will also attempt to further the public's financial literacy through the use of pictorial representations and other devices to help those who may not understand the financial terms used in other, similar products. By combining all of these different products into a single, easy-to-use platform, FINEX will help our users make the most of the money they earn and consistently make smart spending and investment choices.

Functional Requirements

1. User Account
   a. As a user, I would like to be able to register for an account.
   b. As a user, I would like to be able to login to my account.
   c. As a user, I would like to be able to reset my password if I forget it.
   d. As a user, I would like to be able to edit my personal information
   e. As a user, I would like to be able to change my associated email address
   f. As a user, I would like to be able to sign in with my social media accounts so that I can have other ways of creating an account
   g. As a user, I would like to be able to add a profile picture
2. Budget Tool
   a. As a user, I would like to be able to create a monthly budget
   b. As a user, I would like to be able to get personalized financial budgeting information
   c. As a user, I would like to be able to add income to my finances so that I can manage my budget effectively.
   d. As a user, I would like to be able to add expenses to my finances so that I can manage my budget effectively.
   e. As a user, I would like to be able to have a dashboard of all of my personal finances.
   f. As a user, I would like to view commonly used budgets from other users
   g. As a user, I would like to be able to view personalized advice on how to manage my money so that I can make informed decisions.
   h. As a user, I would like to generate a savable document of my finances so that I can access the information offline
   i. As a user, I would like to keep track of my actual spending as compared to my projected budget plan
   j. As a user, I would like a monthly analysis of where I am spending the most money (food, transportation, rent, etc.)
   k. As a user, I would like to view my overall financial summary so that I can track my budgeting more closely
   l. As a user, I would like to be able to be educated on how to create a budget so that I can make more responsible decisions in my daily life
3. Stock Tool
   a. As a user, I would like to be able to search for a stock
   b. As a user, I would like to be able to view information about a stock
   c. As a user, I would like to be able to view a stock's volume
   d. As a user, I would like to be able to view a stock's volatility
   e. As a user, I would like to be able to view a stock's trend/range
   f. As a user, I would like to be able to save the stocks I'd like to watch so that I can quickly access the stocks that I have interest in.

   g. As a user, I would like a graphical representation of stock prices and how they are changing over time.
   h. As a user, I would like a list of the top 5-10 stocks that increased most in value on a day-to-day basis
   i. As a user, I would like to be able to change the time frame on stocks I'm viewing
   j. As a user, I would like to be able to compare one stocks data to another's
   k. As a user, I would like to be able to view a cryptocurrency's current price
   l. As a user I would like to view the exchange rate between various currencies so that I can better estimate the financial impact of certain purchases
   m. As a user, I would like to change the timeframe over which my financial summary applies so that I can track the effectiveness of changes I am making
4. Analytics and Education
   a. As a user, I would like to be educated on building a strong stock portfolio so that I can make more informed purchases
   b. As a user, I would like to be informed of the risks involved in financial investments so that I can make informed decisions.
   c. As a user, I would like to be educated on risk management for investing so that I can make safer decisions
   d. As a user, I would like to be educated on common stock trend graphs so that I can make more informed decisions
   e. As a user, I would like to be able to simulate stock investments so I can see how much net gain or loss I would incur before investing in the actual market
   f. As a user, I would like to receive recommendations on what stocks to buy and sell
   g. As a user, I would like to be able to view breakout boundaries on the stock's graph
   h. As a user, I would like to be able to view predictions of a stock's future price so that I can better make buying and selling decisions
   i. As a user, I would like to be able to learn more about cryptocurrency so that I can understand different aspects of the financial markets
5. Communication and Sharing
   a. As a user, I would like to receive email or text notifications when the stocks I am following increase or decrease in price beyond a certain threshold
   b. As a user, I would like to be able to share my stocks to social media outlets
   c. As a user, I would like to email a document containing my finances so that I can move my information

   d. As a user, I would like to be able to share stocks with another user.
   e. As a user, I would like to be able to share my budget plan with other users

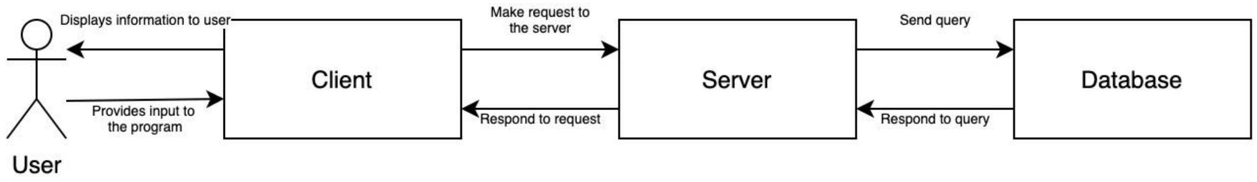
Non-Functional Requirements

1. Usability
   a. As a user, I would like to be able to be able to switch between dark mode and a light mode so that I can have an easier time viewing the application
   b. As a user, I would like to be able to use the application on a tablet
   c. As a user, I would like to be able to view the website on my phone and have it format correctly so that I can view my information from multiple devices
   d. As a user, I would like to be able to get help on how to use specific features of the application.
   e. As a user, I would like the application to have an easily navigable UI
   f. As a user, I would like the application to be aesthetically pleasing
2. Security
   a. As a developer, I would like user password and personal info to be encrypted to protect the privacy of the user.
3. Server
   a. I would like the server to support real time server client communication
   b. I would like the server to be able to store users data in a database

Design Outline

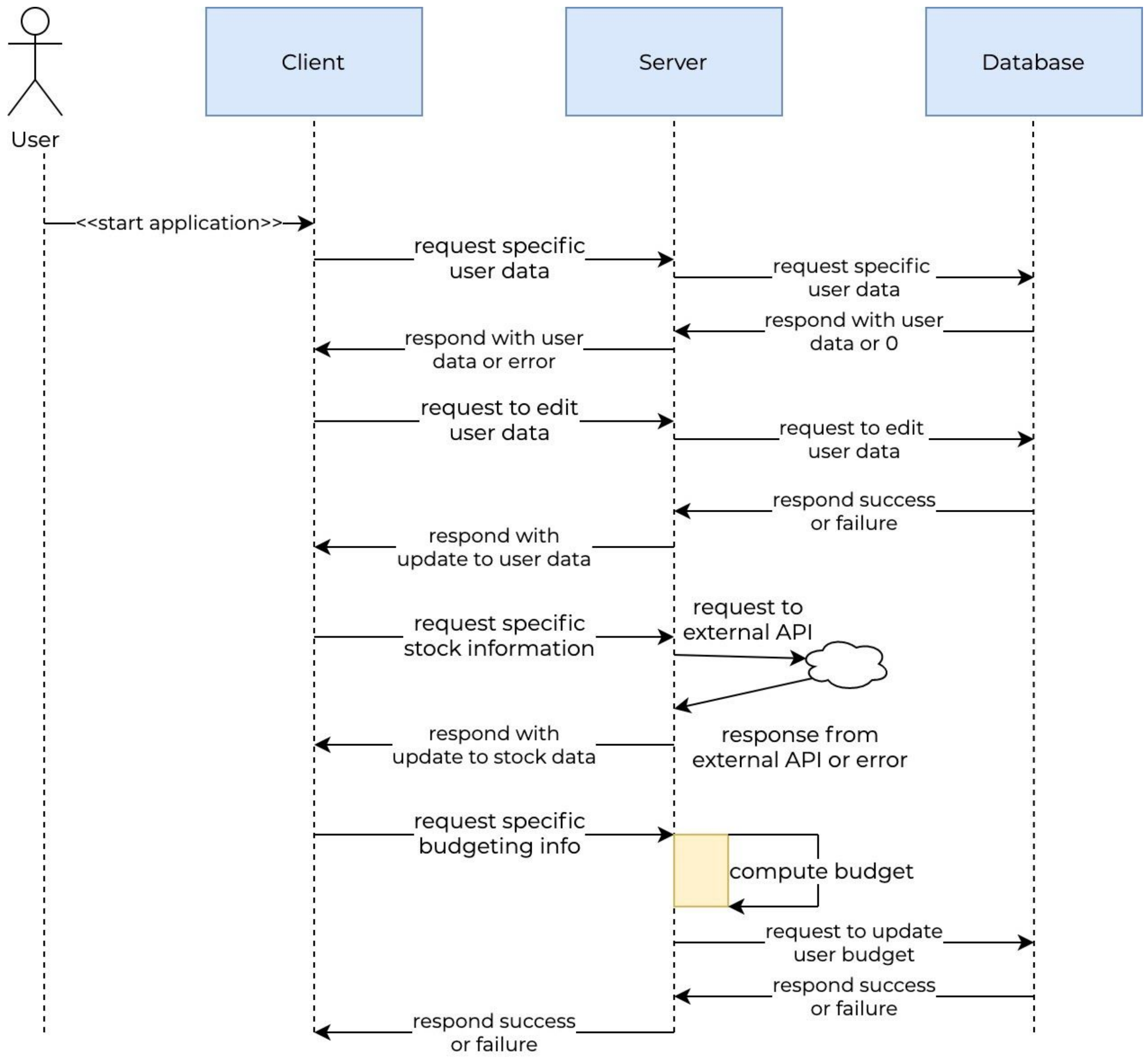
High-Level Overview

This project will be a web application that will allow users to store personal information and data on personal finances. The application will use the client-server model where one server will handle the requests of a large number of clients using the Express.js library in Node.js. This server will then act as an intermediary between the client and the database, as well as a way to get information from APIs such as stock information.


Displays information to user
Provides input to the program
User
Client
Make request to the server
Respond to request
Server
Send query
Respond to query
Database

1. Client
    a. The client will provide the user with a way to interact with our system.
    b. The client will make requests to our backend server to change and update user data.
2. Server
    a. The server will receive and handle requests from the client
    b. The server will perform validation on that request and will directly make queries to the database to create, update, receive, or delete data.
    c. The server will also make requests to retrieve stock information as well and relay that to the client.
3. Database
    a. We will use a nonrelational database to store user data, including account login and personal finance information.
    b. For the purposes of making our product as ethical as possible, we will encrypt all user passwords with the Advanced Encryption Standard (AES) 256-bit encryption algorithm that MongoDB offers.

Sequence of Events Overview and Diagram


Client
Server
Database
User
<<start application>>
request specific user data
request specific user data
respond with user data or 0
respond with user data or error
request to edit user data
request to edit user data
respond success or failure
respond with update to user data
request specific stock information
request to external API
response from external API or error
respond with update to stock data
request specific budgeting info
compute budget
request to update user budget
respond success or failure
respond success or failure

The sequence of events above shows the typical interaction among clients, server and the database. We begin the sequence with the user starting the web application. In the main page, the user can either log in or create a new account if they do not already have one. When a user tries to log in to an existing account, this prompts a database query to make sure the username already exists in the database and then ensures the password matches what is in the database. Depending on how we choose to encrypt our passwords, this could involve re-encrypting the entered password and comparing the encrypted version to what the database contains. Alternatively, if the user is creating a new account, this will prompt a database query to make sure their email is unique and then add their email, username, and encrypted password to the database. The user then logs in, which triggers a request from the client to the server. The server then sends a query to the database, which then receives user data from the database. After that, the server responds to the client with the user information as the user logs in. Once the user is logged in, the client can make requests to the server for various purposes. For example, it can make a request to add user data, which would be handled in a similar way as with the user login (client sends request to the server, which sends data to the database, and then gets a response code to relay to the client). It can also make edits to already defined user data in the same way. Additionally, the user can request information on specific stocks, which will trigger a request to an external API. The external API will then handle a response on the server side or an error, and then handle a response to the client to display the stock information and display it on the screen. As for the budgeting information, the user will trigger a request to the server to get budgeting information. The server will compute the budget, update the database with a query and receive a response, and relay that information back to the client.

Design Issues

Functional Issues

1. What information do we need for signing up for an account?
    a. Username and password only
    b. Username, password, email address
    c. Username, password, email address, phone number

   Choice: b
   Justification: We need a username and password to log the user in. If we want to be able to alert the user to changes in stocks, we must have some way of contacting them, either by email or by text. While it may be nice for some users to get text alerts, we believe email is a safe, secure, and effective option for communicating with the user.
2. What information do we provide on stocks?
    a. Price and Volume
    b. Price, Volume, trend/range and more information.
    c. Price, Volume, trend/range and more information with visuals.

   Choice: c
   Justification: We need to provide relevant stock info for the user in order for them to make intelligent decisions, which means we must at least choose option b. However, it can be very beneficial for the user to have visual tools which leaves c as the best option. In addition, this will make the later features of stock trend analysis simpler as we will have a pre built tool to display any stock's trend.
3. What tools do we provide for user budgeting?
    a. Add income, expenses
    b. Add income, expenses and create budget
    c. Add income, expenses, create budget, and simulated long term savings

   Choice: b
   Justification: We need to provide users with the basic tools necessary to practice budgeting. Simulated long term savings can be nice but is outside the scope of this project. Additionally, we want to keep the interface simple, clean, and user friendly. We can better achieve this by leaving out any unnecessary info from the UI.
4. What Social and Sharing features should we provide.
    a. Stock price email notifications
    b. Stock price email notifications and sharing to social media
    c. Stock price email notifications, sharing social media, and inter-website communication

   Choice: c

Justification: email notifications allow us to alert the user of stock price changes. We additionally want to allow users to share their followed stocks on social media as an additional means of interaction. Finally we decided on inter-website communication for sharing stocks and budgets to foster a stronger community within the site. This will also include featuring popular stocks and budget plans giving new users a place to start.

5. What accessibility options do we give to our users?
    a. Color Blind Support
    b. Color Blind Support and Vision Impared support

Choice: b

Justification: Since our website design is based around only a small range of colors, adding in support for users to remap both our positive (green) and negative (red) color will make our sight readable for all our users. Additionally, by including a dynamic website design, our content should be able to zoom without losing any quality. Our graphs will also be tagged so that a screen reader device can accurately describe what the graph shows to those who may not be able to read the screen.

Nonfunctional Issues

1. What web hosting service should we use?
    a. **AWS**
    b. Azure
    c. Google Cloud
    d. Heroku

   Choice: a

   Justification: We wanted to pick a web hosting service that is commonplace in industry, and none are more ubiquitous than AWS. We feel that it would be great to get experience with AWS, and the abundance of documentation should help us with any errors that we encounter.
2. What backend language/framework should we use?
    a. Spring Boot (Java)
    b. **Express.js (Node.js, JavaScript)**
    c. Flask (Python)

   Choice: b

   Justification: We wanted to pick a framework that was easy to learn and of the choices, we found that Express would be the easiest to learn. Also, by having one language for both the frontend and backend, it will simplify the learning process for the team since there will only be one language to learn for both the frontend and the backend. Although we already know Java, making Spring Boot a likely candidate, we wanted to use a more modern framework such as Express.
3. What frontend language/framework should we use?
    a. Go
    b. **React (JavaScript)**
    c. Angular (JavaScript)
    d. Vue (JavaScript)

   Choice: b

   Justification: We wanted to pick a library that was easy to learn again, making React the better choice in this aspect due to its widespread use as well as a few of our group members having familiarity with the library. Since it is very widespread in the industry, there are plenty of tutorials and documentation on the library, making it extremely easy to pick up.
4. What database should we use?
    a. MySQL
    b. **MongoDB**
    c. DynamoDB
    d. AuroraDB
    e. CouchDB

   Choice: b

Justification: A majority of the documentation for full-stack Express applications describe MongoDB as the ideal database because of simple integration and efficient usage. Additionally, some of our team members have already built and integrated MongoDB databases in the past so we felt it would be good to eliminate the learning curve that would come with choosing another database software.

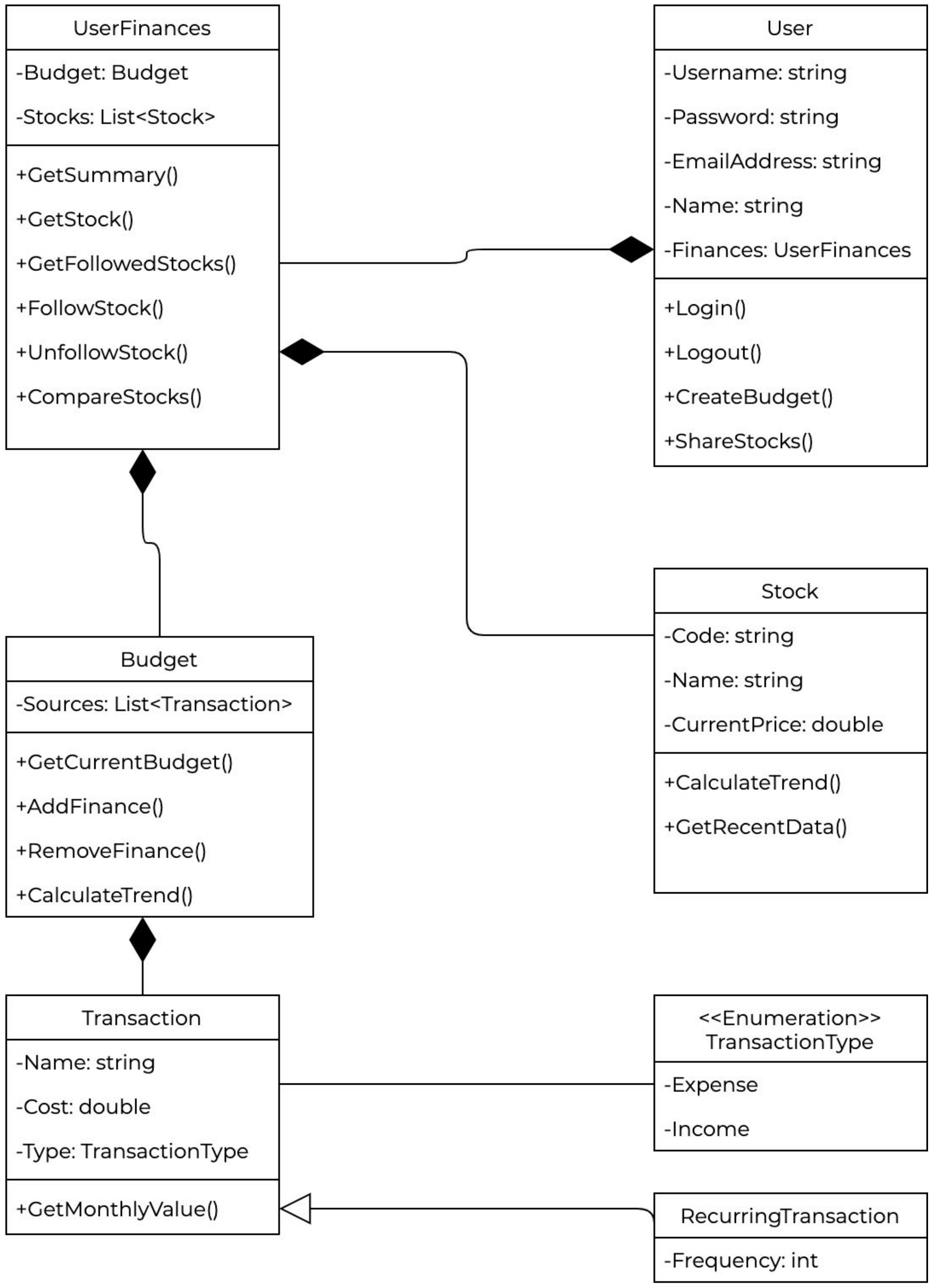
5. Which API should we use to access stock data?
   a. **Alpha Vantage**
   b. Yahoo Finance
   c. IEX Cloud
   d. World Trading Data

   Choice: a

   Justification: Alpha Vantage provides the easiest to use and most well supported API. Yahoo Finance has been depreciated and other API platforms were more complex to pull the same amount of data. Alpha Vantage will be easy, fast, and reliable.

Design Details

Class Design



Descriptions of Classes and Interaction between Classes

The classes represented above are created based on what the user needs to effectively use our web application. Each class represents either the user, or something the user will require access to. The classes are arranged in a manner that allows each user to create their own budget, manage their own stocks and add their own transactions. As such, the user only needs a connection to the larger, UserFinances class.

It is in this class that the stocks and transactions are compiled into an overall summary. Essentially, this is the pipeline through which the UI and user can request and access certain data. Within this UserFinances class is a list of stocks tracked by the user as well as a budget.

The Budget class needed to be separate from the overall UserFinances because it does not include stock values. It contains individual transactions, which can be sources of income or expenses, indicated by the enumeration. Individual transactions need to be maintained in their own class because there could be many, with different names, quantities. They also may differ in frequency if they are recurring, hence the need for the RecurringTransactions class.

Lastly, the Stock class is maintained separately because it must use API requests to access the data. It also stores any information that needs to be readily accessible such as the name and stock symbol.

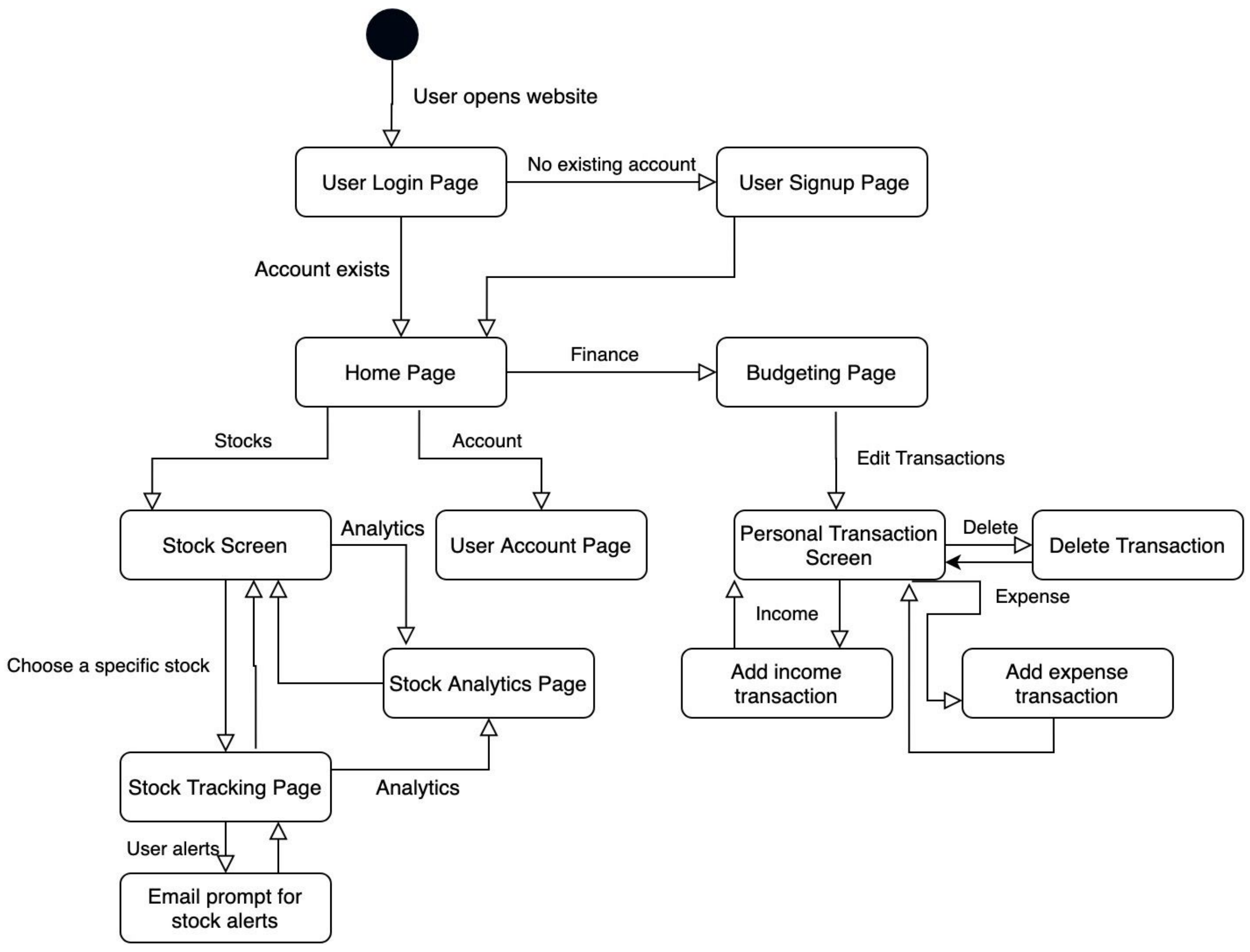
1. User
    a. A user object is created upon the creation of an account on our platform.
    b. Each user will have a unique username which will be used for login purposes.
    c. Each user will have a password associated with their username in order for them to login. Additionally, this password will be encrypted so that only the user will have access to their account.
    d. Every user will create a budget that will then be associated with their account.
    e. The user, based on stocks they have followed, can share these with other users as well.
2. UserFinances
    a. This class is a wrapper for many of the various finances that may be added by the user.

   b. One instance of this class will contain a list of stocks followed by the user as well as a budget, containing any of the recurring or one-time expenses the user may have.
   c. Because of the various finance types, various methods are supported by this class including the ability to follow and unfollow stocks, compare them, and obtaining a list of them.

3. Budget
   a. The budget class facilitates all non-stock related expenses and sources of income.
   b. The user may choose to add an individual finance to their budget.
   c. The overall budget will contain a list of all individual finances and coalesce them into a more usable format
4. Transaction
   a. Each transaction will have an identifiable name, so that the user knows what a certain expense or source of income is.
   b. Transactions should also have an associated cost, which is then incorporated into the budget.
   c. The transaction may have a type, indicated whether it is a source of income for the user, or an expense to be subtracted from the overall budget.
5. TransactionType
   a. The type of transaction (income or expense) must be monitored in order to distinguish what calculation should be performed on the larger budget.
   b. This will be of type enumeration since there are a finite number of options for this data type (e.g. expense or income).
6. RecurringTransaction
   a. If a transaction occurs repeatedly over a period of time, the user may wish to specify this.
   b. The frequency will be tracked in weeks and will then be inserted into the monthly budget via the summary.
7. Stock
   a. The stock class will contain a symbol and a name of the stock so that it is identifiable and its data is retrievable.
   b. Each instance of a followed stock will have a stock object associated with it.
   c. The data (i.e. price and previous prices) will be pulled via API's and will be done in this class.
   d. The data will be updated on a periodic timeline as the user utilizes the application.

Navigation Flow Map

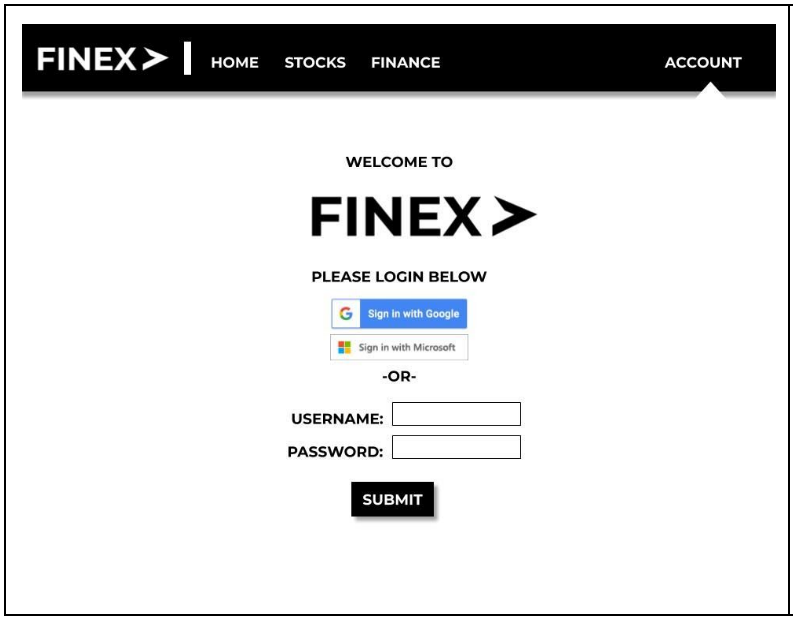
Our website is designed to be easily navigable. Once a user enters the site, they may either register an account, which will take them to a signup page, or login directly if they already have a registered account. Both actions will bring the user to the home page. From there, they can use the static top bar (refer to UI mockups on pg. 16) to access their budgeting and stock dashboard, as well as edit account information. The budgeting dashboard will include ways to add a budget goal and monetary transactions, both income and expenses. It may also, time permitting, guide users on creating a basic budget. The stock dashboard will display the user's list of chosen stocks and related info, as well as provide quick access to detailed stock information and an option to receive email notifications when a stock price has fluctuated by a threshold of their choosing. It also gives users access to an analytics/teaching tool which will provide basic analytics of stocks while also teaching the users how to interpret them.

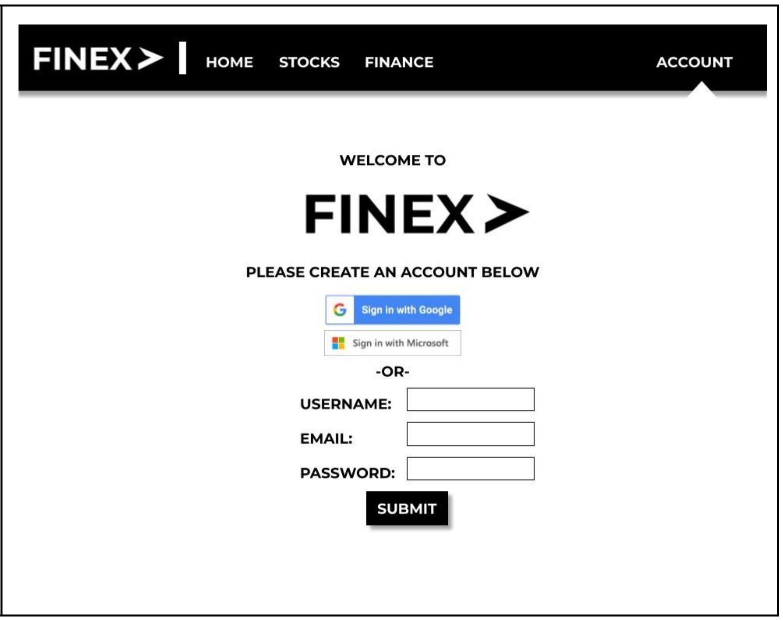


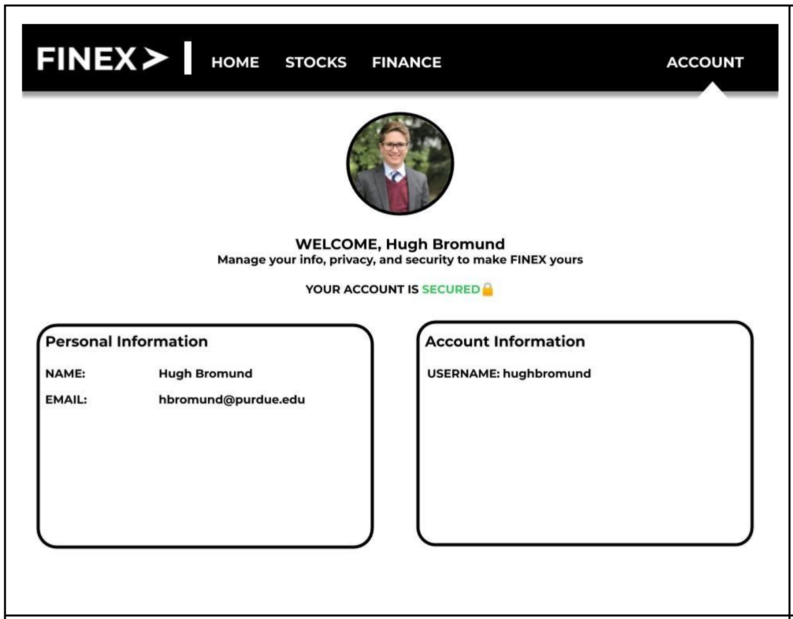
UI Mockup

The FINEX UI is designed with simplicity and ease of use at its core. To achieve this goal we decided to go with a simple black-and-white color scheme. Our goal was to make information both easy to ready and convey a lot of information with only a glance. This is achieved through the sparing use of two colors. By default these colors are green, for good things, and red, for bad things. The entire interface is built to make it very easy to see these colors and quickly understand fundamental information about the content of the page. For example, when viewing a data about a particular stock which is doing well, such as the stock price increasing, will be colored green. This color scheme follows through the entire interface and every screen will use color only when necessary.

The landing page has a simplistic log-in request with options to use 3rd party services instead of creating a FINEX account. The account page contains your account and personal information along with options to edit your settings. Our stocks and finance pages contain the bulk of our features. Shown below are pages that demonstrate the user's personal finance and portfolio as well as individual stock lookup and budget.







WELCOME, Hugh Bromund
Manage your info, privacy, and security to make FINEX yours

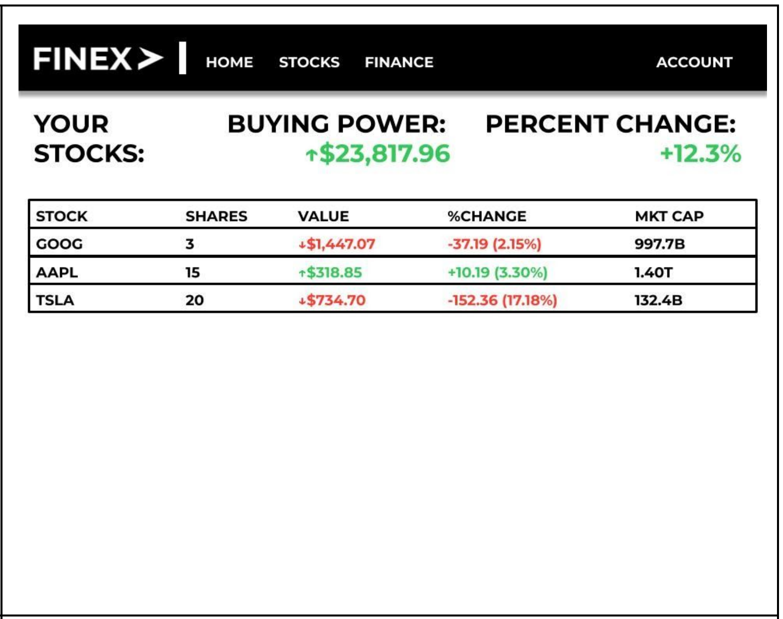
YOUR ACCOUNT IS SECURED

Personal Information

NAME:	Hugh Bromund
EMAIL:	hbromund@purdue.edu

Account Information

USERNAME: hughbromund

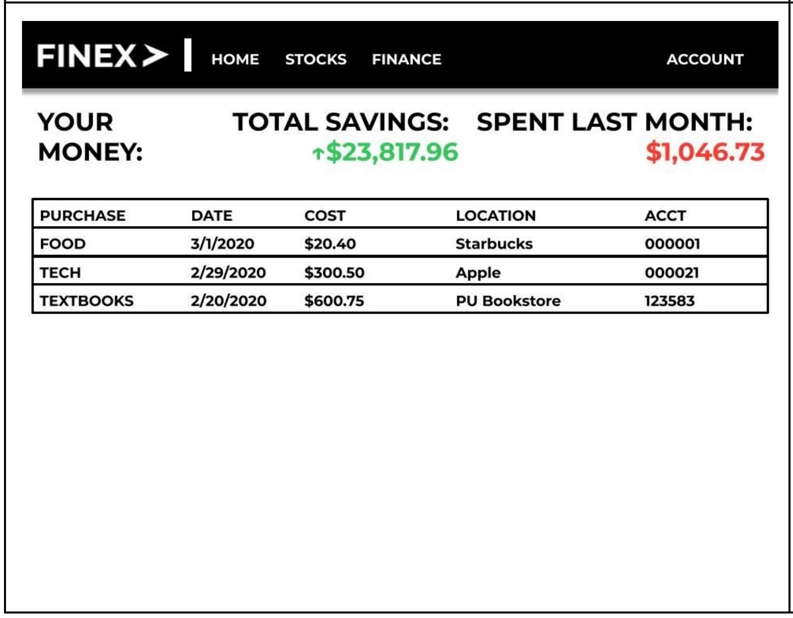


YOUR STOCKS:

BUYING POWER: ↑$23,817.96

PERCENT CHANGE: +12.3%

STOCK	SHARES	VALUE	%CHANGE	MKT CAP
GOOG	3	↓$1,447.07	-37.19 (2.15%)	997.7B
AAPL	15	↑$318.85	+10.19 (3.30%)	1.40T
TSLA	20	↓$734.70	-152.36 (17.18%)	132.4B

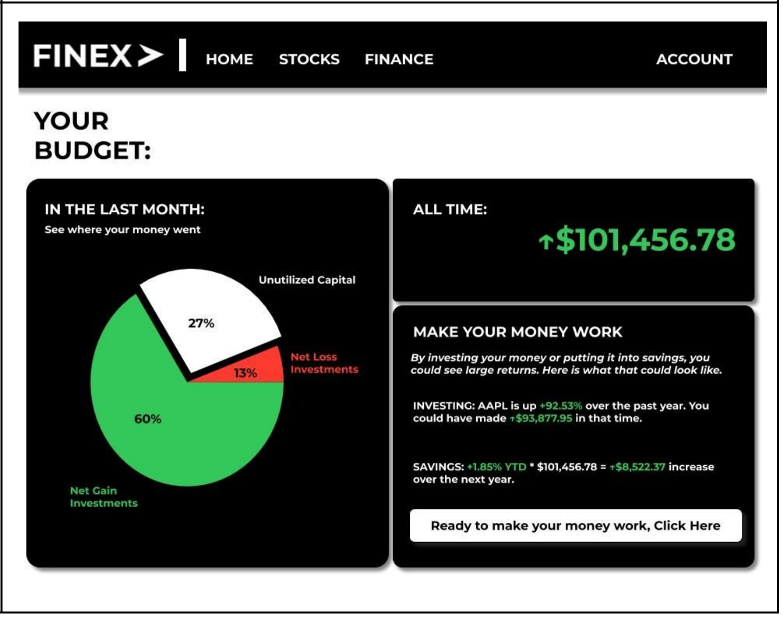


YOUR MONEY:

TOTAL SAVINGS: ↑$23,817.96

SPENT LAST MONTH: $1,046.73

PURCHASE	DATE	COST	LOCATION	ACCT
FOOD	3/1/2020	$20.40	Starbucks	000001
TECH	2/29/2020	$300.50	Apple	000021
TEXTBOOKS	2/20/2020	$600.75	PU Bookstore	123583



YOUR BUDGET:

IN THE LAST MONTH:
See where your money went

ALL TIME:
↑$101,456.78

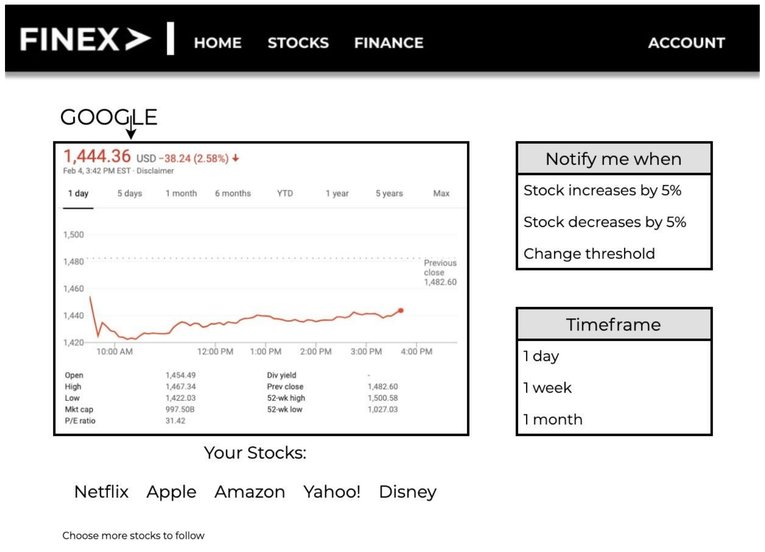
MAKE YOUR MONEY WORK

By investing your money or putting it into savings, you could see large returns. Here is what that could look like.

INVESTING: AAPL is up +92.53% over the past year. You could have made ↑$93,877.95 in that time.

SAVINGS: +1.85% YTD * $101,456.78 = ↑$8,522.37 increase over the next year.

Ready to make your money work, Click Here



1,444.36 USD −38.24 (2.58%)
Feb 4, 3:42 PM EST · Disclaimer

Open	1,454.49	Div yield	-
High	1,467.34	Prev close	1,482.60
Low	1,422.03	52-wk high	1,500.58
Mkt cap	997.50B	52-wk low	1,027.03
P/E ratio	31.42		

Notify me when
Stock increases by 5%
Stock decreases by 5%
Change threshold

Timeframe
1 day
1 week
1 month

Your Stocks:

Netflix Apple Amazon Yahoo! Disney

Choose more stocks to follow