Team: Buy Bitcoin

Growth Chart

Responsibilities:

- Graphical Display
- Investment Buckets
- Growth projections
- Funding
- Historical performance of buckets

Collaborators:

- Portfolio

Funding

Responsibilities:

- Rounding
- Income
- Expenditures
- Time

Collaborators:

- Portfolio
- (User Story: C)

Portfolio

Responsibilities:

- Balances
- Funding
- Stocks
- Investment Buckets

Collaborators:

- Investment Bucket
- Funding
- (User Story: D,E)

Investment Bucket

Responsibilities:

- Stocks
- Name

Collaborators:

- Stock
- Growth Chart
- Portfolio
- User
- (User Story: E)

Team: Buy Bitcoin

User

Responsibilities:	Collaborators:
• Name	• Bank
• Google Login	
• Bank Login	• (User Story: A)

Bank

Responsibilities:	Collaborators:
• Transactional Data	• User
• Balance Data	
• Location Data	• (User Story: B)

Stocks

Responsibilities:	Collaborators:
• Historical Stock Data	• Investment Bucket
• Daily Stock Quote	• Growth Chart
• Ticker	• Portfolio
• Name	• Daily Stock Quote

Daily Stock Quote

Responsibilities:	Collaborators:
• Ticker	• Stocks
• Name	
• Price	

Team: Buy Bitcoin

Growth Chart
• displayType: string • time: string
render()

The Growth Chart, which is a graphical representation of a customer's balances (historical and projected) is at the highest level of classes in our system. It relies on several different classes for its information (as will be shown in the diagram below). It has 3 important values it takes, which are the customers asset values considering three different paths to be taken: nothing, save, invest.

Funding
• balances: int • time: string • type: string
calcRounding() calcIncomePercent() calcExpenditurePercent() calcFlat()

The Funding Class, contains information on a user's value and assets determined by different types of income accrual. It has different operators which calculate user balances depending on different saving methods, as well as selected dates.

Portfolio
• stockPercent: int • currentVallue: int
modifyPortfolio()

The Portfolio, is a collection of all the assets for a user. It shows the percentage of holdings in different investment buckets, and their current value. One of its operators allows for users to update the percentage breakdown of assets in that portfolio.

Team: Buy Bitcoin

Investment Bucket
• stock: int • percentage: int
addStock() deleteStock() modifyStockComposition()

An investment bucket represents a combination of stocks to which a certain amount of money will be distributed according to a certain Stock Configuration. It supports operations to add, delete stocks, as well as to modify distribution configurations.

User
• name: string • id: int • google-id: string
authorize ()

A user is the key to the platform, and is the one who tests the platform with their personal data. Users must login through google in our platform, so we need their google account information. The operator is to verify that the user is logged on and active.

Bank
• name: string • userId: string • password: string
getTransactionData() getBalances()

The bank class represents the available information collected from an external API that allows us to access a user's bank details, transactions, among other data. The operators allow us to request specific information requests from said bank.

Team: Buy Bitcoin

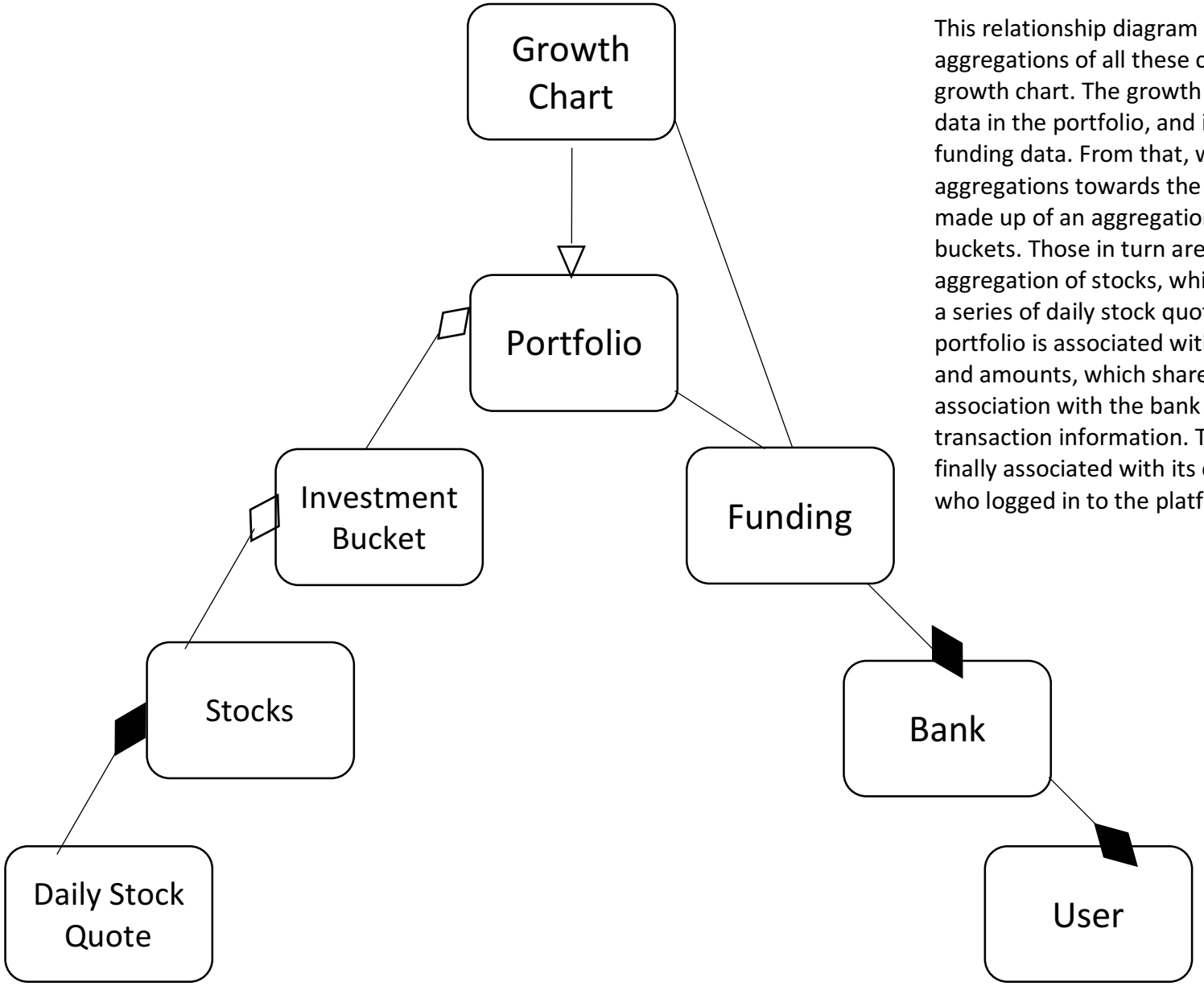
Stock
• name: string • ticker: string • time: string
getHistorical()

The Stock Class represents a specific stock and ticker with its historical price data. Its operator allows access to stock data available for 10 years prior, if that stock was listed.

Daily Stock Quote
• name: string • ticker: string • value: number

This class simply represents a daily quote calculated for a specific stock, which is taken from the stock price over a continuous amount time, and calculated to simplify to a daily value.

Team: Buy Bitcoin



This relationship diagram represents how the aggregations of all these classes leads to a growth chart. The growth chart inherits the data in the portfolio, and is associated with funding data. From that, we see a series of aggregations towards the portfolio, which is made up of an aggregation of investment buckets. Those in turn are made up of an aggregation of stocks, which are comprised of a series of daily stock quotes. Finally, the portfolio is associated with funding methods and amounts, which shares an important association with the bank class and its transaction information. The bank class is finally associated with its corresponding user, who logged in to the platform.

Team: Buy Bitcoin

ADDITIONAL CLASSES:

graphQL
• graphQlTypes: string
makeResponse()

This class exposes the Graph QL endpoint to Django HTTP requests. Its operator makes a response from said endpoint so Django can process it.

Stock market
• name: string • ticker: string
validateTicker() fetch()

The stock market class opens up endpoints for yahoo finance data to access both historical and current stock information. The operators validateTicker guarantees that a certain stock exists, and fetch will retrieve the information.

Pair info: We all worked on this assignment, with Jordan and Julian dealing with CRC cards and class descriptions, and Nigel and Christophe handling class diagrams and operator, type definition.