

Test Plan:

defineSector():

Input validation

- Null or empty sector name

Boundary cases

- Try to add same value twice.
- Try to add same value in capital and small case.
- Try to provide already registered sector name.
- Provide blank sector name.
- Try to add the sector which has the special characters
- Try to add the sector having leading and trailing spaces
- Try to add the sector which has mixed case characters

Control flow

- Try to define the sector with some random values.
- After attempting to define the sector, verify if the sector was successfully added to the database.
- Retrieve the list of sectors from the database before and after defining a new sector to ensure the new sector is present.

Data flow

- Try to call the function before and after every other function in the system.

defineStock():

Input validation

- Enter Null or empty company name, stock symbol, or sector.
- Non-unique stock symbol.
- Undefined sector.

Boundary cases

- Try to add the same stock symbol twice.
- Try to add the same stock symbol in capital and lowercase.
- Try to add many stocks for same company and vice versa.
- Try to enter same stock for different sectors.
- Provide blank company name.
- Provide empty stock symbol.
- Provide blank stock symbol.
- Provide empty sector name.
- Provide blank sector name.

- Provide registered stock symbol.
- Provide unregistered sector name

Control flow

- Try to define a stock.
- After attempting to define the stock, verify if the stock was successfully added to the database.
- Try to add the same stock multiple times
- Try to add the multiple stock for same company
- Try to add the multiple stock for same sectors

Data flow

- Retrieve the list of stocks from the database before and after defining a new stock to ensure the new stock is present.
- Try to call the function before and after each and every function.

defineProfile():

Input validation

- Null or empty profile name.
- Undefined profile sector holdings.
- Sector holdings do not sum up to 100%.

Boundary cases

- Try to define the same profile twice.
- Try to provide the same profile name in capital and lowercase.
- Try to add two same values for profile sectors.

Control flow:

- After attempting to define the profile, verify if the profile was successfully added to the database.
- Define the profile for the stocks that are not in the sectors.
- Define the profile with total percentage not summing to 100.

Data flow:

- Retrieve the list of profiles from the database before and after defining a new profile to ensure the new profile is present.
- Try to call the function before and after the call of every function.

setStockPrice():

Input validation

- Enter Null or empty stock symbol.

- Negative or zero per-share price.
- Undefined stock symbol.

Boundary cases

- Try to set the price for a newly defined stock.
- Try to set the price for stock that is not in the list.
- Try to enter negative stock price.

Control flow

- Set stock price zero.
- Try to set stock price to 0.
- Try to update the price before and after selling the stocks for a particular account.
- Try to set the price multiple times for same stock symbol.

Data flow

- Try to call the function before and after the call of every function.
- Retrieve the stock price from the database before and after setting the price to ensure it was updated correctly.

`addAdvisor():`

Input validation

- Enter a 0 or number string in input.

Boundary cases

- Try to add the advisor name that is already present in the database.

Control flow

- Try to add the same advisor many times in the database.
- Try to add the advisor with very long name that has spaces in it.
- Try to add the advisor that includes number or special characters as its name.

Data flow

- Try to add the advisor before and after every function of the system.

`addClient():`

Input validation

- Enter a 0 or empty number in input.

Boundary cases

- Try to add the client name that is already present in the database.

Control flow

- Try to add the same client many times in the database.
- Try to add the client with very long name that has spaces in it.
- Try to add the client that includes number or special characters as its name.

Data flow

- Try to add the client before and after every function of the system.

`createAccount():`

Input validation

- Enter a 0 or empty number in input for client and advisor id.
- Enter null or empty string for account name and profile type.
- Enter empty for Boolean value

Boundary cases

- Try to add the account name that is already present in the database.
- Try to create multiple accounts for same client.
- Try to create multiple client and advisor pair accounts.
- Try to update the Boolean value once it is already defined.

Control flow

- Try to add the same client many times in the database.
- Try to add the client with very long name that has spaces in it.
- Try to add the client that includes number or special characters as its name.
- Create multiple advisor client pair for the same account.
- Create account that has Boolean true first and then false ahead.
- Create multiple accounts for same client.
- Assign multiple accounts to same advisor.
- Create different account with same account name.

Data flow

- Try to create the account before and after every function of the system.

`tradeShares():`

Input validation

- Enter a 0 or empty number in input for trade shares and account id.
- Enter null or empty string for stock symbol.

Boundary cases

- Try to give many stocks to same account until its cash is over.
- Try to call function when account is not there or stock symbol is not there.
- Try to call the function when the trade share is 0 or less than 0.

Control flow

- Create different account with same account name.
- Buy more than 1000 stocks for a account.
- Try to trade share when there is no cash in the account.
- Try to sell the same amount of stocks that were bought.
- Try to do trade share for stocks that the account does not have.

Data flow

- Try to create the account before and after every function of the system.

`changeAdvisor():`

Input validation

- Enter a 0 or empty number in input for advisor id new and account id both.

Boundary cases

- Try to call the function when the trade share is 0 or less than 0.
- Try to call the function for the account that is not present.
- Try to call the same advisor for the account that it already has.
- Try to call the function for advisor that is not present.

Control flow

- Try to call the function multiple times in a row.
- Try to call the function for the account that has no advisor.
- Try to call the function before and after the stocks has been updated.
- Try to call the function for the advisor that has lowest or highest profit.

Data flow

- Try to create the account before and after every function of the system

`accountValue():`

Input validation

- Enter a 0 or empty number in input for account id

Boundary cases

- Try to call the function when the trade share is 0 or less than 0.
- Try to call the for the account that has just been created and no trade done.
- Try to call for the account that is not present.

Control flow

- Try to call the function multiple times in a row.
- Try to call the function for the account that has no advisor.
- Try to call the function before and after the stocks has been updated.
- Try to call the function for the advisor that has lowest or highest profit.
- Try to call for the account that has maximum or minimum account value.
- Try to get the value before and after multiple trades.
- Try to get the account value after selling all stocks or after buying all stocks with max cash capacity.
- Try to call for the account that has maximum profit.

Data flow

- Try to create the account before and after every function of the system

`advisorPortfolioValue():`

Input validation

- Enter a 0 or empty number in input for account id

Boundary cases

- Try to call the function when the trade share is 0 or less than 0.
- Try to call the for the advisor that has just been created and no trade done.
- Try to call for the advisor that is not present.

Control flow

- Try to call the function multiple times in a row.
- Try to call the function for the account that has no advisor.
- Try to call the function before and after the stocks has been updated.
- Try to call the function for the advisor that has lowest or highest profit.
- Try to call for the advisor for accounts that he has that has maximum or minimum account value.
- Try to get the value before and after multiple trades.
- Try to call for the advisor that has multiple accounts and no accounts at all.

Data flow

- Try to create the account before and after every function of the system

`investorProfit()`:

Input validation

- Enter a 0 or empty number in input for client id

Boundary cases

- Try to call the function when the trade share is 0 or less than 0.
- Try to call for the client that has been just added.
- Try to call for the client that is not present.
- Try to call for the client that is in loss or has no shares.

Control flow

- Try to call the function multiple times in a row.
- Try to call the function before and after the stocks price has been updated.
- Try to get the profit for one account that is in profit and other which is in loss.
- Try to get the profit when the stock price is maximum.
- Try to get the profit when there is one stock in the holdings.

Data flow

- Try to create the account before and after every function of the system

`profileSectorWeights ()`:

Input validation

- Enter a 0 or empty number in input for account id.

Boundary cases

- Try to call the function when all sectors are not defined.
- Try to call the function when the account id is not present.

Control flow

- Try to call the function multiple times in a row.
- Try to call the function when the trade share is 0 or less than 0.
- Try to call for the account that is in loss or has no shares.
- Try to call for the function that has no ttrade share done.
- Try to call the function when the all stocks from all sectors are bought.
- Try to call the function when the account has only one type of sector stock and cash only
- Try to call the function when the account not exists.

- Try to call the function where each sector has same weightage.

Data flow

- Try to create the account before and after every function of the system

divergentAccounts ():

Input validation

- Enter a 0 or empty number in tolerance level and account id.

Boundary cases

- Try to call the function when all sectors are not defined.
- Try to call the function when the sectors id is not present.
- Enter 100 tolerance and 0 tolerance.

Control flow

- Try to call the function when the account not exists.
- Try to call the function where each sector has same weightage.
- Try to call when the accounts has no investment.
- Try to call the function when the investments somehow becomes as it was defined in the profile.
- Try to call the function when there the difference between defined and current sectors is very negligible or very high.

Data flow

- Try to create the account before and after every function of the system

disburseDividend ():

Input validation

- Enter a 0 or empty number in dividend per share.
- Enter null or empty stock symbol.

Boundary cases

- Try to call for the stock symbol that is not present.
- Try to call the function when the stock symbol is not bought by any account.
- Try to call the function when the dividend per share is negative.
- Try to call the function when the dividend is 0.

Control flow

- Try to call the function when the account not exists.
- Try to call when the accounts has no investment.
- Try to call for the stock symbol that is hold by every account.
- Try to call for the stock symbol before and after updating the stock price.
- Try to call when the dividend is given by one stock multiple times.

Data flow

- Try to create the account before and after every function of the system

[stockRecommendations\(\)](#):

Input validation

- Enter a 0 or empty number for every input integer parameter.

Boundary cases

- Try to call for the stock symbol that is not present.
- Try to call the function when the stock symbol is not bought by any account.
- Try to call the function for account that has every stock bought or has no stock bought.
- Try to call the function when there is only 1 numComparator.
- Try to call the function when there is a tie between the recommendations number of stocks.

Control flow

- Try to call the function when the account not exists.
- Try to call the function multiple times after and before buying the recommended stocks.
- Try to call when the accounts has no investment.
- Try to call for the stock symbol that is hold by every account.
- Try to call for the stock symbol before and after updating the stock price.
- Try to call the function when the numComparator is greater then the total accounts present.

Data flow

- Try to create the account before and after every function of the system

[advisorGroups\(\)](#):

Input validation

- Enter a 0 or empty number for every input integer parameter.

Boundary cases

- Try to call for the advisors is not present.
- Try to call the function for maximum 100 and minimum 0 tolerance.

- Try to call the function for maximum groups that are possible and for 0 groups as well.
- Try to enter negative values for both input parameters.

Control flow

- Try to call the function with 0 max groups and 0 tolerance.
- Try to call the function with 100 tolerance and max groups as size of the total advisors.
- Try to call the function when all the advisors have similar sector holdings.
- Try to call the function when all the advisors have different sector holdings.
- Try to call the function with advisors having loss or least profit.

Data flow

- Try to create the account before and after every function of the system