

Searching for Stocks:

searchForStock(search: String, availableStocks: List)

```
FOR int i from 0 to the length of availableStocks:  
    IF name of stock at index i equals search:  
        return stock at index i  
    ENDIF  
END
```

return NO_STOCK_FOUND

Sorting stocks based on search criteria:

sortStocks(availableStocks: List)

```
LET temp = first stock in availableStocks list  
LET min = first stock in availableStocks list  
LET place = 0  
FOR int i from 0 to the length of availableStocks:  
    FOR int j from i to the length of availableStocks:  
        IF sort criteria of min < sort criteria of stock at index j  
            LET place = index j  
            LET min = stock at index j  
        ENDIF  
  
        temp = availableStocks[i]  
        availableStocks[i] = min  
        availableStocks[place] = temp  
  
    END  
END  
  
return availableStocks
```

Relevant news articles for respective stocks:

getNews(stock: Stock, newsArticles: List)

```
LET newList = new list
FOR int i in 0 to length of newsArticles list
    LET tags = newsArticles[i]'s list of tags
    LET relevanceCounter = 0
    FOR int j in 0 to length of stock's tag list
        IF tags contains tag from stock's tag list at index j
            relevanceCounter++
        ENDIF
    END

    IF relevanceCounter >= 3
        add newsArticle[i] to newList
    END

return newList
```

Display Ads or Not:

IF valid_ad_page (The current page user is on has a spot to display an ad):

IF ad_free_version = false:

FOR all advertisements stored:

before: IF user clicked "don't view this ad anymore" on this specific add

SKIP ad

END IF

IF ad payment is up to date from advertiser:

DISPLAY ad

END loop

END IF

```
        END FOR
    END IF
END IF
```

Friend's List Recommendation:

```
IF user_contact_permissions = 1:
    LET list_on_app to hold contacts
    FOR each contact:
        IF contact.number registered:
            ADD contact TO list_on_app
        END IF
    END FOR
    DISPLAY list_on_app
END IF
ELSE
    DISPLAY "Allow contact permissions to view people you know on this app too!"
END ELSE
```

Displaying Portfolio Information:

SET theme = green

IF beginning_day_total > end_day_total:

 SET theme = red

END IF

SET total_amount_diff = beginning_day_total - end_day_total

SET total_percent_diff = (beginning_day_total - end_day_total)/beginning_day_total

LET stock_list for stock changes

FOR each stock owned

 SET stock_theme = green

 IF beginning_day_stock > end_day_stock:

 SET stock_theme = red

 END IF

 SET stock_amount_diff = beginning_day_stock - end_day_stock

 SET stock_percent_diff = (beginning_day_stock - end_day_stock)/beginning_day_stock

 LET stock(theme, stock_amount_diff, stock_percent_diff, beginning_day_stock)
 (stock object)

 ADD stock TO stock_list

END FOR

IF stock_list.length <= 1:

 DISPLAY stock_list

END IF

ELSE

 DISPLAY "There are no stocks in your portfolio"

END ELSE

Purchasing/Selling Stocks:

Purchasing:

BuyStock(User, Stock ID, amount):

```
    WHILE amount > 0:  
        ADD Stock ID to User's List of Stocks  
        SUB Stock price from account balance  
        SUB 1 from amount  
    END WHILE
```

SellStock(User, StockID, amount):

```
    WHILE amount > 0:  
        IF User's List of Stocks has remaining stocks matching StockID  
            REMOVE StockID from User's List of Stocks  
            ADD Stock Price to account balance  
            SUB 1 from amount  
        END IF  
        ELSE  
            DISPLAY "Number of stocks attempted to sell greater than the amount  
owned. All stocks sold"  
        END ELSE  
    END WHILE
```

Stock Recommendation Algorithm:

RecommendStock(User):

```
    INITIALIZE Recommended List of Stocks  
    INITIALIZE List of Categories  
    FOR each Stock in User's Stock List:  
        DETERMINE category of stock  
        ADD 1 to category value in list of categories  
    END FOR  
    SORT List of Categories by Value (Occurrence)  
    FOR each Category in List of Categories:  
        GET Categories List of Stocks  
        SORT Category List of Stocks by shares owned  
        FOR x = 0 to 3:  
            GET Stock(x) from Category List of Stocks  
            ADD Stock to Recommended List of Stocks  
        END FOR  
    END FOR
```

Return Recommended List of Stocks