

## CIT 591 Group Project Design - CRC (Class Responsibility Collaborators):

**Proposal:** Our team members are Tiffany Choi, Rob Stanton, and Joseph Kravets. Our project is a program where a user can enter a stock ticker and get useful technical and statistical analysis of its trading history. For example, a user will enter the ticker "AAPL" and can see a linear regression of the price, the 200 day moving average, the 50 day moving average, the zscore of the price, and other useful metrics.

Joseph Kravets will pull the stock data and make sure it is usable. Tiffany will choose and implement the financial metrics. Rob Stanton will design and implement the user interface.

---

### Scanner Class

- The basic responsibility of the Scanner class is to prompt the user entry of the stock ticker and desired indicator. Will be used for coverage on headless environments.

<b>Class:</b> Scanner	
<b>Responsibility:</b> <ul style="list-style-type: none"><li>• Prompt the user for the stock ticker</li><li>• Prompt the user for the desired indicator</li></ul>	<b>Collaborators:</b> Stock FinancialIndicators Portfolio DataPull

### DataPull Class

- The basic responsibility of the DataPull class is to pull the data for the stock from an external source and store the data in an internal file.

<b>Class:</b> DataPull	
<b>Responsibility:</b> <ul style="list-style-type: none"><li>• Read data from external source</li><li>• Store data in separate file</li></ul>	<b>Collaborators:</b> RefreshData

### RefreshData Class

- This class should refresh the stock data

<b>Class:</b> RefreshData	
<b>Responsibility:</b> <ul style="list-style-type: none"><li>• Refreshes the stock data</li></ul>	<b>Collaborators:</b> DataPull

### Stock Class

- The basic responsibility of the Stock object is to maintain information about a single stock.

<b>Class:</b> Stock implements Serializable	
<b>Responsibility:</b> <ul style="list-style-type: none"><li>• Create new object given the stock ticker</li><li>• Read data from file</li><li>• Store the stock's ticker</li><li>• Store the stock's name</li><li>• Store the stock's date, time, and price</li><li>• Update the stored information about the stock</li><li>• Updates itself</li></ul>	<b>Collaborators:</b> DataPull

### Portfolio Class

- The basic responsibility of the Stock object is to maintain information about a single stock.

<b>Class:</b> Portfolio implements Serializable	
<b>Responsibility:</b> <ul style="list-style-type: none"><li>• Store list of Stock Objects</li><li>• Store Alerts Class</li><li>• Saves State</li><li>• Add/Remove/Get Stock</li><li>• Add/Remove/Get Alerts</li></ul>	<b>Collaborators:</b> Stock

### FinancialIndicators Class

- The basic responsibility of the FinancialIndicators object is to calculate the financial indicators for the selected stock.

<b>Class:</b> FinancialIndicators	
<b>Responsibility:</b> <ul style="list-style-type: none"><li>• Calculate the Moving Average</li><li>• Display the Moving Average</li><li>• Calculate the Bollinger Bands</li><li>• Display the Bollinger Bands</li></ul>	<b>Collaborators:</b>

<ul style="list-style-type: none"> <li>• Calculate the RSI</li> <li>• Display the RSI</li> <li>• Calculate the MACD</li> <li>• Display the MACD</li> </ul>	
--	--

### Alerts Class

- Maintains list of Stocks with their respective chosen alerts

<b>Class:</b> FinancialIndicators	
<b>Responsibility:</b> <ul style="list-style-type: none"> <li>• HashMap&lt;Stock, List&lt;StringInd&gt;</li> <li>• Calculates each Indicator</li> </ul>	<b>Collaborators:</b> FinancialIndicators

### BaseGUI Class

- This class should create the base GUI (without the stock information).

<b>Class:</b> BaseGUI	
<b>Responsibility:</b> <ul style="list-style-type: none"> <li>• Creates the Base GUI</li> <li>• Contains functionality between components</li> </ul>	<b>Collaborators:</b> Menu ChartGUI TableGUI

### ChartGUI Class

- This class should plot the selected stock and its metrics on the GUI.

<b>Class:</b> ChartGUI	
<b>Responsibility:</b> <ul style="list-style-type: none"> <li>• Plots the details and metric from a chart object on the GUI</li> </ul>	<b>Collaborators:</b> Chart

### Chart Class

- This class will take a stock and assemble all necessary details to form a ChartGUI.

<b>Class:</b> BaseGUI	
<b>Responsibility:</b> <ul style="list-style-type: none"> <li>• Calculates plot locations for specific ticks as well as prices and indicators.</li> </ul>	<b>Collaborators:</b> Stock FinancialIndicators

<ul style="list-style-type: none"> <li>Should be passed to ChartGUI for construction of a chart</li> </ul>	
--	--

### TableGUI Class

- This class should list elements of Portfolio and contain information concerning their indicators and price as well as weightings in the portfolio

<b>Class:</b> TableGUI	
<b>Responsibility:</b> <ul style="list-style-type: none"> <li>Display all stocks in Portfolio and their respective Alerts</li> <li>Allows adding alerts or removing</li> </ul>	<b>Collaborators:</b> Portfolio

### ProgramController Class

- The basic responsibility of the ProgramController object is to carry out the various use cases.

<b>Class:</b> ProgramController	
<b>Responsibility:</b> <ul style="list-style-type: none"> <li>Calls UI based on Environment</li> <li>Calls Refresh Class</li> <li>Calculate the FinancialMetrics</li> <li>Create the StockGUI (from inputted data)</li> <li>Saves State</li> <li>Loads Previous State if available</li> </ul>	<b>Collaborators:</b>

### Runner Class

- The basic responsibility of the Runner class is to run the program.

<b>Class:</b> Runner	
<b>Responsibility:</b> <ul style="list-style-type: none"> <li>Run the program</li> </ul>	<b>Collaborators:</b>