Architecture

The client wants a simulated stock market that can be run on some server. Multiple teams should be able to write and run trading programs (bots) that can interact with the server using HTTP over the internet, with the purpose to buy and sell stock, thus increasing the teams score (The total value of the assets they have at hand). The end goal of each team is to have the highest finishing score. To aid with this, there should also be a website where members of CSS Bristol overseeing the event and the various teams can see graphs representing the history of the stocks on offer, a leaderboard that shows the current positions and scores of all of the teams, and also a history of transactions made by the team’s bots

The stock market will be written in Java using the SpringBoot framework, which will allow us to use other plug ins like Thymeleaf (to interact with/ act as the website) and REST, which will make it easier for the student written bots to interact with the server. The application will receive incoming HTTP requests and use those to update the state of the program, thus allowing the bots to send and receive information to/from the server.

The website will be written using HTML/CSS/Javascript, using JSReact to show graphs showing a stocks price over time, with the javascript dynamically updating what graph is being showed depending on how the user interacts with it. The Javascript can also interact with the RESTfull elements of the server to get up to date information both on what score each team currently has, and also the state of the stock market at that point in time.

The stock market will be made up as a single object, containing multiple stock objects it shall create by reading in a csv file. This will allow the exact stocks used run to run to be easily varied, and each stock be updated on its own. The stock markets prices will be changed by three events. A buy, a sell, and news. The news system will generate news about various sectors and will alter the prices of stocks based of what sector they were stated to be in as part of their entry in the csv. Buy and sell events will raise and lower the stock price accordingly. To add an element of randomness to the stock market (because even seasoned economists are supprised by the way it acts sometimes) we will create a number of java based bots behind the scenes to buy and sell stock pseudo-randomly. Certain bots will favour certain stock, with weights attached to buying and selling. This will allow predictable trends to emerge in the market for team’s bots to notice and exploit. This will also stop the situation from arising where completely random bots cancel out to a net 0 influence on all stock, as each stock is equally likely to be bought as sold and thus over a period of time no change will occour.