Release testing

For release testing, we will partially be using the test classes provided by spring, but may also have to write some custom code ourselves. In the process to creating the product we will be creating some Java based bots to interact randomly with the stock market, and these bots will have been tested as part of a unit test, therefore can be assumed to work. The environment any of these release tests will be running in will be the stock market itself running on an AWS instance, and then multiple bots connecting to that instance and attempting to trade with the stock market. The exact conditions will change from test to test, as different parts of the integrated system will be tested

|  |  |
| --- | --- |
| Stress test | CSS Bristol has run many events in the past and estimates that an event utilising the Stonks project will attract around twenty teams to come and join the competition. By running a simulation of the event using thirty bots (an extra 50%) running multiple requests per second (random, so much faster than expected condition), we can see if the program would be able to withstand abnormal operating conditions. |
| Anti-cheat test | As all the competitors will be computer scientists studying both programming and security, there is the very real possibility of teams attempting to attack/game either the system or each other in an attempt to gain an edge. This test will check our guards against such attacks. This will involve bots that attempt to PUT multiple more bots onto the system, bots impersonating other bots with fake data, multiple requests simultaneously in an attempt to force undocumented behaviour, and any other possible weak spots identified over the course of development |
| Regular use test | Over the course of general use the server is not expected to have to deal with many outlying conditions. This test will involve connecting a small simple python-based bots (the language we expect competitors will use) to the server and running a full length 10 hour session of the stonk market. During the course of this, the website will be regularly checked, and the bots will keep track of what they expect their score to be at the end. After the simulation is over, these expected scores will be compared to the scores the system gives them to check if the system is running as expected. |

There is the possibility for one more test. Assuming a beta version of the stink market is finished by March, CSS Bristol has expressed their interest in running a short small scale event to gather data about what extra features could be added to the final release. This real world test of the program could serve as a test of the system as a whole up until that point Of course, all other tests will have been run on the product beforehand to minimise the chance of an error occurring during actual use, however such real world conditions may highlight any previously unnoticed flaws in the system.