Crypto Ticker App

Group 12: Edgar Martinez, Rameez Baig, Zakee Jabbar, Ahmed Khan

Product Summary

The **Crypto Ticker App,** based upon **Bitcoin Ticker App**[1], is a one-stop shop for any individual that is interested in tracking different kinds of cryptocurrencies. This product tracks various cryptocurrency from the more well known crypto like Bitcoin to the smaller, less well known crypto like ZCash. This application provide users with cryptocurrency prices for the selected day, month, or year along with their respective highs, lows, and averages. To make this information more visually appealing to the user the application provides a graph that shows the price manipulation over the time period that the user has selected, all while allowing the user to select the type of currency that they would like to see the information in. The application also provides the user with news related articles, specifically tailored to the cryptocurrency that the user has selected.

Testing

Since the application's main feature is to display prices and news on the specified cryptocurrency, to the user, the bulk of the code that was tested and inspected included code related to how the price and news data is fetched. Testing was conducted on code that handles the API calls that establishes the communication between the application and the respective API. Tests were also conducted for the API response and making sure that all data sent from API was correctly retrieved and displayed by the application. The testing procedures used for the test was creating a proper URL and calling the API using that URL and then receiving the JSON object back and making sure that all expected data was received. As such no exceptions were thrown when code was tested and all data was retrieved correctly indicating that the code passed the test and inspection. Tests were performed various times with all three different API's that are used by the application which include the cryptocurrency API, currency exchange API, and news API. Each time the code passed the tests no matter what API was used. An additional feature that was tested was news articles, and making sure that when a news article is clicked by a user the application launches the browser and opens the news article. The test consisted of opening the application and clicking on an article, then having that article open in the browser and making sure that the same article that was clicked is the one that was displayed in the browser. Tests

[1] Boyle, Jie, Thomas, and Jake Vercimak. "Bitcoin Ticker App Project Report.", 2014.

were ran a couple of times, clicking on different articles and the making sure that they were same articles that were clicked on and thus passing the test.

Some other features that were also tested was the code to read and write to a CSV file and loading the last selected currency. The goal of both of these tests was to make sure that the persistent data was correctly saved, opened, and loaded correctly. The testing procedure was simple as opening the application, selecting a currency and time period, and then closing the application The application was reopened to verify if the currency, cryptocurrency,and time period selected was that same as what had been previously selected. If it was the same the test was passed.These tests were ran multiple times and the cryptocurrency and currency selected along with the time period was different each time the application wa ran. All the times the test was conducted they passed.

The front end tests deal with all the different types of changes the user can make and the UI changes that happen as a result of that. All test were run to verify that all the components were updated properly. Different cryptocurrencies, currencies, and time periods were used for all tests to make sure all combinations work.

Lingering Issues

Some issues that were present while developing the project include the need to have multiple threads for the application. This is due to the fact that when data gets pulled from the API it will temporarily freeze the UI due to the heavy computation work that is being done in fetching and displaying the data. Simple solution would be to have a background thread that handles all work that involves retrieving and displaying the data. This would prevent the UI thread from freezing and allow the user to interact with the application as the data is loaded from the worker thread.

Conclusion

Before project development started the group got together and decided to divide the work up into two different teams, a front-end and back-end team. This helped the development process by allowing both teams to use pair programming thus limiting errors, by allowing an extra pair of eyes to review the code as its typed. This helped tremendously when the code was combined because it allowed group members from opposing teams to get together to quickly and efficiently combine the projects while allowing the group to fix any issues as they popped up due to the knowledge the group members had on their respective part.