APPLE 2.0

Apple 2.0

Stock Portfolio Project

Scrum Report

Team Members: Courtney Colbert, Mike Gutkind, Henning Kjoeita, Lindsay Scott
Scrum Master: Mike Gutkind
Product Owner: Courtney Colbert
Estimate of total person-hours spent on all aspects of job until now: 17

Summary: During sprint two we focused on a lot of the functionality of our program. In doing this we found that we should organize our “service layer” and “Back end” so that the last two sprints we can focus on making the GUI much more practical and useful for the user. This sprint was a lot of trial and error. At the end of sprint one, we set out to create an accurate depiction of any selected stock symbol and parse that specific stocks information in a graph form. We also aimed to connect our stock portfolio program to a backend service (database) so that we can store a specific user’s information. On top of all this we needed to add an interface that allowed for a user to sign into the application with their specifically designated credentials. As of 4/10/18 we were successfully able to meet all these goals and then add even more functionality than we expected. We were able to accomplish several goals related to functionality during this sprint. We still have more functionality we would like to add along with many other UI goals we’d like to meet in the upcoming sprints. Thus we will be able to design the best user interface and efficiently compliment the functionality of the program.

User Stories:

- Working Charts - As a user, I would like to have charts of my stocks so I can visualize trends and see how my portfolio is divided. **Implementers:** Mike and Henning
    - Story Points: 8

- Password Protection- As a user, I would like to have the option of setting a password so my portfolio information is secure. **Implementers:** Courtney and Lindsay
    - Story Points: 8

- Specific Stock News- As a user, I would like to be able to select an individual stock such that it opens a browser displaying any news in regards to said stock. **Implementers:** Mike and Henning
    - Story Points: 4
- Search Functionality- As a user, I would like a search bar with stock suggestions so I can follow market trends and potentially find stocks to add to my portfolio. **Implementers:** Mike, Henning, Courtney and Lindsay
    - Story Points: 4
- Backend Database - As a user, I would like to sign into my profile on any computer so that incase my computer dies, my personal account data will be saved in the back end. **Implementers:** Lindsay
    - Story Points: 8

Integration Testing:

Integration testing took a decent amount of time and sensitivity to make sure we could make the different parts of the program run as smoothly as possible. We got together three separate times in order to integrate three different features. The first feature we integrated was our StockSearch page and our Data page. This involved integrated the Stock Search page's ability to parse json data from our api and pass that to the chart displayed on the Data page. The second integration we underwent was integrating the Swing user interface login screen. Courtney worked on the functionality and build of the login page and then helped integrate it with the entire project. Our final integration tests included backend service with Courtney's login user interface. We are still working on visually integrating these components on the UI but the functionality of the database layer interacting with the service layer works when the project is run.

The overall quality of this project is very functionality focused right now. So far everyone has been pulling their weight very well and producing beautiful components that integrate almost seamlessly. Something we plan on improving in the future is the look and feel of the user interface. Considering the amount of functionality we currently have, we will have more than enough time to improve and build a high quality user experience for our final product.

Scrum II retrospective: Each team member's perspective of their individual contribution and accomplishment

Michael's Contribution (Scrum Master): Apple 2.0 accomplished a lot this sprint. As scrum master i approached our development in a slightly different way. At the very beginning of the sprint i divided each tab into different classes. Mike did this so that we could each work on a specific page without stepping on each other's toes. This allowed me to focus most of my energy on the Stock Search functionality and back-end API calls. first, mike fully integrated Maven into our project and imported specific libraries regarding charts, parsing and json retrieval so that we could get all the information from the API cleanly. The next thing Mike did was implement the fully functional search function which allowed for any stock to be searched for and then displayed in chart form with fully accurate data. Mike then integrated this with the entire project upon finishing testing.

Courtney's Contribution (Product Owner): My main focus for this sprint was creating a menu screen that requires a password to access the stock ticker. I made the password hard coded into the program. The password can either be submitted by clicking the login button or pressing the enter key. If the text in the JPasswordField matches the designated password, it closes the menu frame and opens the stock ticker, and also clears out the text in the JPasswordField. I also added a logout button to the summary page that closes the stock ticker and reopens the menu screen. I rearranged the summary page a bit and created an array list that will store the user's favorite stocks. I added some placeholder stocks to the array list to make sure my loop displays all of the stocks from the array list on the frame properly.

Henning Kjoeita's Contribution: This sprint I worked on implementing a way to graph the data from stocks, I used JFreeChart as the base and worked from there to implement it into our project. I created a separate class that can be used to get a JPanel containing the graph. (The getChart() function places the chart on a ChartPanel, and then the ChartPanel on a Jpanel which it returns, that is so it's easy to use to chart for my team members). The chart class handles everything to create the chart, it only needs to be given some series (list of data points) that it can graph. I also enabled the graph to show more than 1 series (or more than 1 stock in this case) on the same chart. I then implemented the class into the dataPanel tab to demo it's functions, and test that it works as intended. For this purpose I have created two buttons, one that displays a random stock, and one that adds a random stock as a series. To to that I also made the dataPanel implement the actionlistener interface.

Lindsay Scott's contribution: For the second sprint, I deployed a NoSQL database using MongoDB Atlas to house user credentials and portfolio data, though I focused solely on creating and accessing user credentials for this sprint. MongoDB stores collections of data as documents in BSON format, which is an extension of the JSON format. The application's database consists of a "users" node, or collection, where all users are stored. Each user has a unique id, a field for storing a username, and a field for storing a password. Though it's considered bad practice to store passwords as plain text, for the sake of time and simplicity, user passwords will be stored in the database as a string. This can be easily improved by hashing the password before storing it in the database. I created a MongoConnect class to connect to and access the database. The class currently has two main functions, user creation and user authentication. The user creation method takes a username and password string from the user and scans the "users" collection to see if the username passed already exists in the database. If it does exist, the user will be asked to pick another username for their account. If the username doesn't exist in the database, the username and password are pushed to the database to create a new user. Like the user creation method, the user authentication method gets username and password strings from the user trying to sign in. The "users" collection is scanned for the username entered. If the username exists, the document is converted to a JSON object and parsed to get the stored password for that username. If the password retrieved from the database matches the password entered by the user, then the user has successfully signed in.

Product Owner's statement of quality of product: In its current state, the UI is simple and easy to use. There is a search bar that gives suggestions which allows for easy access to data on the stock of my choice, and displays a chart so I can see how that stock has been doing. Displaying real time data is one of the most important aspects, so I am satisfied with the way it currently displays and works. A password was also implemented on a menu screen which will keep my portfolio secure.

Scrum Master's Statement: Overall, the team collectively put in approximately 17 hours of work this sprint. A lot of these hours were dedicated to each person's designated task and then about an hour designated to integrated each specific feature. We were able to add much more functionality and features to our program without the risk of interfering with each other's code. While we have a lot more functionality than our last sprint, we must spend time on finishing the functionality aspect and cleaning up the user interface. Our user interface has not been our first priority and that is why we must cut off functionality in the next sprint so that our program is not only functional but also intuitive and easy to use. Another feature we plan on adding in the upcoming sprint is finishing the back-end schema so that user's can make new accounts and can save their favorite data accordingly.

Set up for sprint 3, New User Stories for Sprint Three:

- Intuitive UI- As a user, I would like to have a much more accessible user interface so my experience can be a lot more enjoyable than staring at a disorganized grey screen.
    - Story Points: 16
- Functional Summary Page- As a user, I would like a summary on the homepage so I can easily check and analyze my favorite stocks.
    - Story Points: 8
- Quick Access to Stock Gains and Losses- As a user, I would like a page displaying gains and losses so I can see the state of my portfolio by having my stock gains and losses in one place.
    - Story Points: 8
- Color Coded Gains/Losses- As a user, I would like to color code stocks according to gains and losses so it's easier to see whether stocks grew or fell.
    - Story Points: 1
- Graph Variable Time Ranges- As a user, I would like to set a time range for graphs so I can identify patterns in designated periods of time.
    - Story Points: 2

Signatures:

Scrum Master ____

Product Owner ____

Team Member #1 ____

Team Member #2 ____