**Scrum Report Format**

* Prologue
  + Team Name: Nird
  + Team Slogan: Nirds of a feather
  + Team Members: Patrick Gallagher, Spencer Russell, Tianyi Zhang, and Hank Zhu
  + Project Name: Stock Ticker

Sprint 3 Report:

2.1 Teammates Roles:

* Scrum Master: Tianyi Zhang
* Product Owner: Spencer Russell
* Team Members: Patrick Gallagher and Hank Zhu

Hours:

Patrick Gallagher: 10 hours

Spencer Russell: 6 hours

Tianyi Zhang: ` 10 hours

Hank Zhu: 7 hours

* Combined Hours on the job: 33 hours

2.2 User Stories

* As a user, I want a help function for the application. So that I know, exactly what the program is doing - 8pts
* As a user, I want to make a password for the application, so that I can protect my stocks. -16 pts
* As a user, I want to have data graphs for different time periods, so that I can see the trends more conclusively - 4 pts
* As an investor, I want to keep hold of my personal shares. So that I can monitor my own financial gain.-4 pts

2.3 Design, Requirements, Test Plan (Data User Story)

2.3 Implementation details and issues (Data User Story)

2.3 Statement of outcomes (Data User Story)

2.3 Names of Implementer(s) (Data User Story)

2.3 Design, Requirements, Test Plan (Login and Registration User Story)

* I designed both login and registration function for stocker ticker in order to protect user’s data in a better level. My user interface is well neat and simple with two columns in the main page (username and password), and three columns in the registration page (username, password and confirm password). If the user doesn’t have a username and password, all they need to do is go to registration page where they can get a new one.

2.3 Implementation details and issues (Login and Registration User Story)

* Implementation process was painful and fun, since I had to write an algorithm to ask registration page validation between “password” and “confirm password” column, if these two doesn’t match, the program will reject the registration and ask them to try it again. If the registration completed, it will direct user to the main page where they can login in their username and password that they just registered. On the background, the program will write the username and password to the local file, so the main login page is able to compare the information with the user input in the main page and decides whether user typed right password or not.
* The issue I have right now is the individual profile for every unique user. Basically, there is only one profile for whoever login in the program.

2.3 Statement of outcomes (Login and Registration User Story)

* The “Login and Registration” function works perfectly and smooth, I debugged dozen of times, and test every possibility that might cause the program compromise. Turns out, everything works fine.

2.3 Names of Implementer(s) (Login and Registration User Story)

* Hank Zhu

2.3 Design, Requirements, Test Plan (Help Window User Story)

* Build a panel that would list helpful information for anyone new to the product. Provide paragraphs for each frame to explain functionalities. This should be contained to a separate frame and should be closable by the user. It is implemented by a button on the homepage that creates a frame to populate.
* clearly organize points to allow future adjustments to the help window
* Minimal functionality suggests that the only desire is to have the window pop up and close without incorrectly displaying text or impacting the program on its close.

2.3 Implementation Details and Issues (Help Window User Story)

* While working on the frame, the panel was not visible from all machines. This was corrected by sharing the file with teammates to have them view it; while not the most efficient method it allowed the team to give their input on the design. The panel now effectively populates the pop-up frame consistently. A grid layout way used as a vertical box layout after the latter wouldn’t correctly compile with the rest of the product.

2.3 Statement of outcomes (Help Window User Story)

* The window does display information about the other panels and their functionality in the product. The code is clean enough to add more information sections in possible future iterations.

2.3 Names of Implementer(s) (Help Window User Story)

* Spencer Russell

2.3 Design, Requirements, Test Plan (Save User Story)

* All of the saved stocks are posted for the user to see in the main menu, under all of the user information. The design for the save function, is allowing the users to add the stocks that they like and also seeing all of the stocks that they have bought. All of the bought stocks are stored in a text file, once the user pushes the add button with the stock information the program will write to the text file with all useful stock information. In the main frame of the project the users can see all of the added stocks. The program reads the text file that has all of the bought stocks. The program reads the text file and makes the stock names into buttons that can be clicked to go to the stock graph page. The design of the buttons has the stock name, how many stocks the user has bought, and the percentage increase or decrease from when the user has bought the stock. If the percentage is above zero then there would be a green arrow and a red arrow if the percentage is less than zero.
* The requirements for the save function is that the program needs a way to save all the stocks that the user has bought. The group found that a text file costs less time than using a database. Another requirement is that reads the text file and prints out stock information for the user. When printing out the stocks into buttons, the team thought that the user wanted to have more information about the bought stocks. The buttons had the percentage change and how many stocks the user has bought the same stocks.
* The test plan for the save function is testing everytime the code has changed. There was a test plan for writing the stocks, reading the stocks, placing stock names into the buttons, and showing the correct information of the stock into the button for the user.

2.3 Implementation and Issues (Save User Story)

* The group implemented a text file instead of using a database. Most of the implementation was quick and easy, until the stock button had to display the percentage information for the user. At first, the program wrote the stock name then the price of the stock when bought. This caused trouble down the road because the when reading the stocks name first, the program executed the code that revolved around the stock information. That information dealt with the percentage change, so the code needed the price first. The price was equal to zero at the time, so the first stock that was read had a 100% decrease. The user would not like that error because they spent money on the stock and they do not want to be given false information. This issue was solved by printing the price of the stock first, then the name because the price of the stock would be the correct information and give the user the correct information.

2.3 Statement of Outcomes (Save User Story)

* Saving works fine, the bought stocks gets written into the text file with no errors. Reading the stocks and displaying the information works fine. The only thing that still needs to be worked on, is when adding the stocks the program comes back to the main menu but the frame has not been refreshed, so the user can not see the recently added stocks. The only way to fix that so far is quitting the application and rerunning it.

2.3 Names of Implementer(s)

* Patrick Gallagher

2.4 Integration Testing

* The testing that went into this sprint was having the password function as the main class that runs first and also after implementing to everyone’s code to see if there were any errors. The second integration testing was how the save function would work with the bought stocks and the search stocks. There were some errors in the search stock, but after debugging the code everything started to work fine now.

2.5 Scrum 3 Retrospective

* Spencer’s Perspective:
  + I didn’t have much free time during the final sprint, luckily my teammates covered the more challenging user stories. I primarily focused on the help window which populated the frame implemented by the button Patrick put on the main menu. We, the Nirds, improved our communication during the final sprint. Emails were sent more frequently, and meeting times were more relaxed to allow everyone to work at their own pace. While I am still not the strongest coder on the team, I am pleased with the product we created as a team.
* Tianyi’s Perspective:
* Patrick’s Perspective:
  + In this Sprint, I worked mostly on the save function. The save function was a step up to the last sprint user story, the user story was adding the function of reading the stock and displaying the graph when the button has been clicked. After completing this sprint, I am very proud how the save function turned out to be. Even though I would like to add some new functions into the save function, like the remove function. But we did not have enough time to implement the the function. Our team worked the hardest in this sprint and we communicated much more often than other sprints. I think that everyone worked on something very important in this sprint and they helped each other when there was a problem during the sprint.
* Hank’s Perspective:

2.5 Product Owner Statement:

* This final sprint did not cover all of the user stories originally outlined by the Nirds. However, this program does include most of the user stories, as well as all of the outlined functionality. There was mentioning of a comparison tool to see differences between stocks, however that could still be done manually by the user whereas collecting the information requires the automation. Considering the previous owner statements during the course of this assignment, this sprint wrapped up loose ends to create a satisfactory product. I am also pleased to see the product devoid of advertisements as briefly mentioned some of the team members.

2.5 Scrum Master Statement:

* In this Sprint, we have made a great progress of our software. We have finalized most of functionalities, including the login/register interface, the help window, the trading function and the multiple graphs. During the process of development, we have encountered some bugs caused by bad coding skill and habit, Specifically, the password verification system was not working because of the if statement logic was wrong; the graphs didn’t show up when clicking on the button on different time period, because we initialized the graphic variables in the wrong place, and the price trends line was drew on the wrong coordinates because the coefficient was miscalculated. Fortunately, we fixed the errors by revalidating and refactoring our code. I think every team member has learned a lot in this group work.

·

Signatures:

Scrum Master Name (Printed) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Scrum Master Name (Signature) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Product Owner Name (Printed) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Product Owner Name (Signature) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member Name (Printed) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member Name (Signature) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member Name (Printed) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member Name (Signature) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_