Template:

**Meeting Date**

* Date

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Agenda Item 1

|  |  |
| --- | --- |
| **Item** | **Notes** |
|  |  |

**Action Items**

* Decision 1

**Meeting Date**

* W1: 17th Sep

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Project decision
* Subgroups for Sprint1
* User Stories/Epic

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Project decision | Initially decided FilmFinder however decided to do InvestmentSimulator instead |
| Subgroups | Frontend   * Jet, Dom, Shelby   Backend   * Youshi, Hazel |
| Epic Stories | Users   * Registered users: Have an account and can buy/sell stocks * Unregistered users: Can only browse stock prices   User story outlines   * As any user, I want to search stock by code, so I can tell how it's doing(find my target precisely?) * As a registered user, I want to be able to add stocks to my watchlist, so I can easily see how all the things im interested in are doing * As an unregistered user, I want to be able to make an account so I can do transactions * As any user, I want to be able to see a graph of the stock performance, so I can be informed of any decision I will make * As a registered user, I want to be able to buy/sell stocks, so I can profit * As a registered user, I want to be able to see how much money I lose/gain through my sales of stock, so I can make an informed decision * As a registered user, I want to be able to see the details of the stocks I own, so I can keep track |
| Novelty Ideas | * VIP accounts/subscribers? * Loans (predatory) * Risk analysis of ‘various’ accuracy * News via scraping or fake news * Setup groups of investors * Private forums like r/wallstreetbets * Cryptocurrency * Meme stock market * Sign in with Google/Facebook/API * Auto selling under certain conditions * Recommendation of relevant stocks * Sort * Basic funds？ * Email notification + verification + password reset * Live markets * 2 factor authentication via SMS/ email/ authenticator app * Multiple watchlists + searchable + filters * Sell / buy at set price * History of transactions * Account initialize |
| Front end | * <https://javascript.info/> * React or Angular recommended? |

**Action Items**

* Ask Tutor about clarification on ‘pseudo’ owning stocks?
  + Doesn’t have to be the real world but can be
* Flesh out user stories
* Start on storyboards/ user interface
* Establish tech stack
* Start on proposal

**Meeting Date**

* W1 Sunday: 20th September

**Participants**

* Dom
* Youshi
* Hazel
* Shelby
* Jet

**Agenda**

* Flesh out user stories
* Start on storyboards/ user interface
* Logo / company name
* Establish tech stack
* Start on proposal

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Flesh out user stories | <https://docs.google.com/document/d/1Awn7Bi8ryDeiEGBwH6whPLpMKPUNGDXtrpTM_YFXNKI/edit>  All user stories fleshed out, scrum master to put into JIRA |
| Storyboards / User interface | <https://miro.com/welcomeonboard/7PCG4MmXNaH1LY5wjx85Qtoz6NAJzbXqcQ9G0AKrGRHWuAwoFgMtMEDVREF9nG4o>  Dom- 1.1and 1.2  Hazel - 1.3 and 2.1  Jet - 3.1, 3.2 and 3.7   * <https://miro.com/welcomeonboard/ijJ8024F8gEDdxvnhGOhLWm8pp3m2lCxIF3gVCsr8csjcVRFY7ywxPdYFQnOMcj8>   Shelby - 2.3, 2.4 and 2.8  <https://miro.com/welcomeonboard/P6mIoDOlTc7Et48aGS1gVvdh9G3auSBeKyrBDGjQmbCuoWRVkCuvHC8HgxxMWqs4>  Youshi - 2.5 and 2.6   * <https://miro.com/welcomeonboard/QJVjEnZUx5qIfZNyT1Nan8W7hO4tk5Xtv8QAntbYVW34gF071CkHHaowRYCCT8aQ>   Due by Thursday |
| Logo + Company | Stonks logo like github  TradeHub  “Where everyone starts as a billionaire… and ends up broke” |
| Tech stack | Probably too early to determine now, get most of the proposal done first |
| Start on proposal | Did not discuss |

**Action Items**

* Individual storyboards for each user story due by 24th September
* Ask tutor to look over user stories + storyboard during tute
* Dom to start adding user stories to JIRA
* Start completing work diaries

**Meeting Date**

* W2: Thursday 24th September

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Individual storyboards for each user story due by 24th September
* Ask tutor to look over user stories + storyboard during tute
* Dom to start adding user stories to JIRA
* Start completing work diaries
* Deciding sprint sizes + dates
* Proposal plan

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Storyboards | * All done * Check at next meeting |
| Rachid’s notes on user stories | * On track, looks good so far * Novelty should be something completely new   + Genie Cat   + Debt Simulator * Merge epic story 1 and 2 * Missing story sizes (in terms of days) |
| Dom to start adding user stories to JIRA | * Done * Need changes due to today’s changes |
| Start diaries | * Dom hasn’t started * Reminder to start completing at least weekly |
| Decide Sprint Sizes and Dates, and draft sprint backlogs | * S1 ends wk5 * S2 ends wk7 * S3 ends wk9/10 |
| Things to do for proposal | * Introduction + research into ASX sharegame? * Decide what each sprint contains   + Decided and shown in user\_stories\_v2 doc * Tech + Architecture   + Decided below * Deciding the novelties   + Action plan set * ER diagram |
| Tech Stack + Architecture | Frontend   * React.js   Backend   * Python * Flask * PostgreSQL |
| Jet + Youshi post-meeting | * Discussion of React, Anaconda and virtual environments * Jet will send out tutorial on intro to React for everyone to get familiar with React * Jet will look into setting up some React and Virtual environment things for us to easily start working on code |

**Action Items**

* Before next meeting, everyone chooses or thinks up their own favourite novelty idea and does a basic feasibility study
* Youshi push newest user stories and meeting minutes to repository
* Dom updates the Jira
* Update diaries everyone!
* Go through storyboards together at next meeting

**Meeting Date**

* W2: Sunday 27th September

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Jet’s progress on frontend setup
* Novelty ideas review + selection
* Weekly diary check
* Go through storyboards
* Proposal plan

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Frontend setup | * There is a new channel on Discord for notes on installation and setup * Jet will put up explanation + links |
| Novelty Ideas | Youshi   * Cat idea, random number generators in background * <https://www.forbes.com/sites/frederickallen/2013/01/15/cat-beats-professionals-at-stock-picking/#3559730e621a> * <https://www.forbes.com/sites/rickferri/2012/12/20/any-monkey-can-beat-the-market/#3e3666d7630a> * Won’t be difficult to implement, and definitely ‘novel’   Dom   * Compete against other people/groups * Agree on a certain period of time to compete * Players can’t see opponent’s moves * People can gamble into a pool of money * If you win, you get the pool * If you lose, no penalties * Gambling might be a problem, but can just have a prize instead   Jet   * Lottery/raffle, put in existing stocks as a ‘bet’ to win more stocks * More stocks you put in, more chance of winning   Shelby   * Guidance system, for people who are new to stocks, have floating windows or question marks that teach you what things mean * Hover over and it pops up mini window * Should be simple to implement, considering that CSS has popups * Is this considered novelty?   Hazel   * Users try to predict stock performance, and bet money * If they get it correct, they get money back + reward * If they get it wrong, they lose their bet money   **Summary:**   * Random investment * Gambling * Education   Need to check these ideas with tutor |
| Diary check | * Hazel and Shelby’s missing from the git |
| Storyboard check | Need to fix story numbers  Jet   * 2.1, 2.2 and 2.7 * Add main page of website that contains login button * Remove references to username   Youshi   * 2.5 and 2.6 * Change flow so it starts with stock details page   Dom   * 1.1 and 1.2 * Have arrows from clicked elements to the next page rather than between pages * Only need the pages without login * Clicking on stock in search options navigates to details page * Add stock ‘…’ to indicate ‘all’ stocks * Remove references to selecting date   Hazel   * 1.3 and 1.4 * Instead of single choice of date, allow choosing start and end dates to look between * Better error for not good enough password when registering * Page titles on storyboard * Arrow indicating logout takes you back to regular home page * Copy page from Jet’s storyboard about $1bn start money   Shelby   * 2.3 and 2.4 * Better weak password warning   Missing 2.8   * Shelby to do it |
| Proposal Plan | * Hazel: Background * Dom: Product backlog + Planned sprints -> Done * Shelby: Novel functionality -> Ongoing * Jet: Interface/Flow diagrams -> Ongoing * Youshi: System Architecture   Link:  <https://docs.google.com/document/d/1ubaKJTazUwHFcwt6JsJmdmkdwXKjKx_IfxqQJpYGdSg/edit#> |

**Action Items**

* Dom will email Rashid about our 3 main ideas, as well as if we need UML/ER diagrams
* Everyone needs to fix story numbers + add fixes based on meeting
* Shelby to complete 2.8
* Everyone sets up template/draft for their sections by 1st October
* Hazel and Shelby to upload their diaries to the git
* Jet to put links in #react and people to get familiar with frontend

**Meeting Date**

* W3: 1st Oct

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Decide on novelty
* Check storyboards all updated
* Diary check
* Proposal draft section check
* Frontend setup check

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Rachid Talk | 1. Need at least 3 competitors studied, and their URL 2. Got the go ahead for the 3 novelties 3. Just need more detail in implementation |
| Novelty selection | Final selection:   1. Random investment advisor 2. Challenge users/system 3. Bet on your predictions |
| Storyboard update check | Youshi: done  Jet: done  Hazel: done  Dom: done  Shelby: done |
| Diary check | Youshi: done  Jet: done  Hazel: done  Dom: done  Shelby: done |
| Proposal draft section | Youshi setup the draft outline for proposal |
| Frontend setup | Youshi: Setup system, haven't gone through tutorial  Jet: done  Hazel: needs to setup virtual environment  Dom: Setup System, haven’t done the tutorial  Shelby: needs to setup virtual environment |
| Novelty User Stories | * Flesh out user stories for the novelty ideas, created priorities and sizes. * Changed the ‘challenge’ idea to challenge the system, not other users. * Updated the user story numbers. |

**Action Items**

* Shelby: Storyboards for novelty Ideas
* Front end people: watch tutorials on react; what props are etc.
* Everyone: Do the tutorial.
* Everyone: Finish assigned proposal sections before Sunday. If you finish before Sunday, post your stuff in the chat.
* Dom: update Jira with new user stories. Add new user stories to github

**Meeting Date**

* W3: 4th Oct

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Novelty idea storyboards
* Proposal Sections
* React tutorial plans

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Proposal Sections | Youshi: Software Architecture Diagram   * Draft of what is needed, just need to complete details   Dom: User stories   * Got draft of what is needed * Sprints to be planned out for now-**done** * Something about short selling-**done** * Incorporate Shelby’s work-**done** * Executive Summary   Hazel: Background   * Will add a table * Add some pictures * Highlight why people of different levels of experience would still want to use investment simulators   Jet: Interface Diagrams   * Rest of us will figure out how to fit it in the body * Add current profit column-**done** * User 1.2. Add latest available unit price-**done** * Rewording on Shelby’s novelty stories   Shelby   * Change references from Cat to Advisor-**done** * Change stories to fit new layout * Makes boards standardised to Jet’s board   Everyone include references  Everyone checked zID  When each person finalises their section, let rest of us know so can ask for help |
| Frontend + React tutorial plans | * Focus on proposal * Can discuss on Monday or something |

**Action Items**

* Finish Proposal
* Start coding

**Meeting Date**

* W4: Thur 8th October

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Everyone start coding today
* Switch from PostgreSQL to SQLite

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Coding | Choose Approach   * **All backend then frontend:**   + **Ensures that backend is done**   + **May have less focus on good looking UI for first demo** * Simultaneous:   + Could be confusing/lead to deadends   + Might be more efficient   + Lead to better looking interface for first demo   Backend   * Set up SQLite * Set up yfinance API * Don’t forget unit testing |
| SQLite | * Flask uses SQLite really easily, while PostgreSQL takes significantly more setup and requires layers of abstraction * ER diagram necessary |
| API  Jet  Dom  Shelby | API decision   * yfinance API used to retrieve stock information * Users are not allowed to choose interval, system make the decision * Do not allow users to pick same start and end date * Do not allow to have start date > end date   Done backend functions for US1.3  Raised Errors exceptions  Tested in api\_test |
| Backend  Youshi  Hazel | Implemented basic functionality  Encountered issue with how to handle flow control |

**Action Items**

* Everyone start working on backend

**Meeting Date**

* W4: Sun 11th October

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Flow control and exceptions
* Installing new modules in venv, how to reflect in other people’s venv?
* Front end tutorial
* Jet walkthrough of front end
* Youshi walkthrough on pull requests

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Jet Demo | * Login isn’t working yet but registration is done * Account page needs some empty text and logout button * Use @app.route to define functions that get called fetch   + Use .then to process the results of your function * SetState updates the state and re-renders * Npm start in the react-app folder automatically goes into flask * componentDidMount()   + Defines what you want the page state to be upon reset   + Every time you refresh it does a set of things you want as standard * Remember to bind event handlers like this.handleChange = this.handleChange.bind(this)   + Binding occurs in constructor * Index.js in src folder contains the imports for any new components   + First add in an import   + Then add route to const routing * Component folder, css folder and image folder in the src/ folder   + Component folder is where you write jsx   + Put new folder in components to put new page code in   + Filetype must be .jsx   + New .jsx files must end with ‘export default [classname]’   + Need a <div> wrapper for render return of a jsx   + Navbar is a global class that you can include to the top of every new page/route * Binding such as onClick = {this.onClick}   + Dont do this.onClick() as if youre calling the function   + Youre binding the function reference rather than calling function   + This is different in the arrow function method * Everyone starts with stock details page * All the jsx stuff is client side, the way register works is there is a constant update of the state when there is a change via onChange, and eventually when register is clicked, the state is jsonified and sent to server |
| Exceptions | * Don’t use exceptions for now * Helper functions directly called as part of flow in flask backend rather than in the original functions |
| Front end tutorial | * Youshi: Done * Dom: Done * Hazel: Done * Shelby: Done * Jet: Way beyond that |
| Dependencies | * Everything in the setup so far * Pip install flask python-dotenv * Npm install jquery * Npm install bootstrap * Npm install react-router-dom * Sqlite * Bcrypt * flask-cors   To start, run these commands in two terminals in react-app folder   * Npm start (This runs React) * run run start-api (This runs the flask server) * Reason there are two is because npm start gets the frontend localhost:3000, but the backend flask does localhost:5000 |
| Pull Requests | * Formal way of merging branches * Displays code changes * Can start reviews on code changes, for others to see and discuss * Once reviews are resolved, the merge can go ahead. |
| Availabilities | * <https://www.when2meet.com/?10056867-JMzYV> * Monday 4pm onwards: * Wednesday Night Demonstration Practice |
| More backend fucntions in API | * stock-last(name): gives   + Market cap   + Current price   + Daily volume   + Number of shares   + Annual yield * Stock-last-week(name):   + Last business 5 days of stock-last(name) |

**Action Items**

* Complete login and stock details page tonight
* Big work session from 4pm Monday, otherwise everyone can see each other’s availabilities and just message each other

**Meeting Date**

* W5: 12th October

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Big work session from 4pm Monday
* Discuss presentation

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Stock Details | * Added High, low, daily percentage to the historical component * Added daily to overview   What’s left:   * Making the backend work with frontend   + Round everything to 2dp unless its whole number **DONE**   + Convert daily and annual to percentage numbers **DONE** * Getting dates correct **DONE** * Getting formatting + styling correct **DONE** * Add non-interactive buttons **DONE** * Fix our stories in JIRA **DONE** |
| Presentation | 1. JIRA Tracking    1. Dom kicks off by introducing each of our stories and what the sprint purpose is    2. 3 min 2. Search    1. Shelby explains how search works and navigates to details    2. 1 min 3. Details    1. Jet explains important stuff    2. 1 min 4. Register    1. Youshi explains checks    2. 1 min 5. Login    1. Hazel explains checks, cookie, logout, timeout    2. 2 min |

**Action Items**

* Complete search user story 1.3
* Move everything to done in JIRA when done

**Meeting Date**

* W5: 14th October

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Presentation Practise
* Quick discussion

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Presentation Practise | 1. JIRA Tracking    1. Dom kicks off by introducing each of our stories and what the sprint purpose isju   For our capstone project we decided to implement the investmentSimulator. The purpose of the first sprint is to build a foundation onto which most other user stories depend on. We decided on four user stories that were most suited to achieve this. The user stories are as follows: 1. As a user, I want to search stocks by code and/or name, so I can navigate different stocks more efficiently, as you can see it is marked done.  2.As a user, I want to be able to see the details of specific stocks, so I can be informed about specific stock’s details. This is also marked as done. 3.As an unregistered user, I want to be able to make an account, so that I can get access to registered features. (show done) 4.As a registered user, I want to log in to the website with my credentials, so that I can access my account (show done)  Each user story was completed within the sprint with all functionalities fulfilled. Now Shelby will showcase the search stocks user story.   1. Search    1. Shelby explains how search works and navigates to details   In order to find the target users are interested in and navigate stocks more efficiently, we need to have a search function. Click on the “market” in the navigation bar, it leads us to the search page, u can do it no matter logged in or not.  After a few seconds, this page automatically displays the first 10 stocks in alphabetical order. This list displays companies’ full name, code, latest 5 day changes and the latest closing price. Users can search by code or by name. It is case insensitive.If the user types in and there are no search results, it displays a placeholder stock. If the user types “ANZ” in the search bar, and searches, it gives out the specific stock. And if the user types in “banking” and searches it gives out all companies’ names with banking in it.  From this search page, users can navigate to each stock’s details page. Now Jet will introduce the stock details page for us.   1. Details    1. The stock details page should always show basic stock information such as stock name, market cap, current price, daily volume, number of shares and annual yield.    2. It will show two different tables, one showing the latest stock details and the other showing the stock’s performance over 5 days.    3. Will always retrieve valid information even if one of the dates’s information is invalid, it will fetch the previous or next available valid date’s information.    4. Buttons at the right side for future sprints 2. Register    1. Youshi explains checks   To be able to trade stocks in the future, unregistered users must create an account. Being able to register as a new user is included as part of this sprint.  On the navigation bar, unregistered users can click on login, and then register to create an account. Then they can enter an email and password combination.  Having an invalid format triggers an error  Having a weak password as outlined in the NIST password guidelines also triggers an error  Special characters are sanitised and trigger regular errors  The NIST guidelines specifically state not forcing conditions on the password and allowing special characters.  Once logged in you can see you’re rich  And navbar now contains logout.  *Accounts start with 1 billion dollars*   1. Login    1. Hazel explains checks, cookie, logout, timeout   [Users are able to login thoughthe home page.  In order to successfully login, users need to input a registered email and correct password. If the user types the wrong email or wrong password, the ‘invalid account details error’ will be displayed. If the user still doesn’t input correct info after the third try, the timeout error page will be shown and users will be banned from the login feature until 10 seconds.  Users input correct email and password under the third try,they are able to assess the register feature like an user wallet page. Our website will be allocated a cookie for individual login users.  Finally, login users are able to click the login out button on navbar to end their login session.]  Runtime: 9min |

**Action Items**

* Everyone set alarms

**Meeting Date**

* W5: 15th October

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Retrospectives
* Plan for sprint 2

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Rachid’s ‘Feedback’ | * Change invalid search message to something more meaningful   + Jet * Expand NIST guidelines to actual text   + Youshi * Validate password, type it in twice   + Youshi * validate email by sending an email requiring confirmation * add a login name/username * apparently don't have simple timeout for logging in * Navbar contains text showing who they are logged in as |
| Retrospective A | <https://docs.google.com/document/d/1f1Gv6lP09nUrwUfcI5qeVuSFX4RgN6kPPF1F7dZ94vg/edit?usp=sharing>  what went well, what didn’t go so well, and what the team should try to over the next sprint to improve their work process  Someone enforcing things to try  Date, time, and team members present or absent at the retrospective  To discuss again on Sunday, gives ourselves time to think more |
| Email | Hi Rachid,  Thanks for your feedback on our system.  You suggested that we implement a feature to validate the email address when registering an account. However this feature is not required in the project objectives, nor is it in our backlog of user stories.  Would we be penalised for not adding a user story to implement this additional feature that you proposed? |
| Sprint2 | 2.5 Watchlist   * Dom * Hazel   2.6 Buy/Sell Stocks   * Jet * Youshi * Shelby |

**Action Items**

* Retro A revisit
* Sprint 2 discussion
* Choose scribe for the week

**Meeting Date**

* W5: 18th October

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Choose scribe
* Retro A revisit
* Sprint 2 Discussion
* Start working on assigned component

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Choose scribe for the week | * 2 days rotation * Responsible for taking down meeting minutes and setting up agendas/actions for both meetings. * Ensure meetings go in orderly fashion/don’t waffle * Upload minutes to github |
| Sprint2 | 2.5 Watchlist   * Dom * Hazel   2.6 Buy/Sell Stocks   * Jet * Youshi * Shelby * Rework primary key in accounts table to a user\_id with enumeration * Setup watchlist and portfolio table to keep track of user’s stock information. * Account table changes:   + Add user\_id as the primary key   + Email is now normal attribute   + Add capital as new attribute * Watchlist table attributes:   + Change user\_email foreign key to user\_id from account table   + Foreign key referencing company\_id from company table   + Primary key is composite of both foreign keys * Company table attributes:   + Primary key company\_id that is enumerated   + Text/String attribute company\_code * Portfolio table attributes (Weak Entity):   + Foreign key referencing user\_id   + Foreign key referencing company\_id   + “total\_quantity” attribute to store number of stock owned   + “avg\_price” attribute to store price at point of purchase   + Primary key is composite of both foreign keys * **TODOS (2.5):**   + Separate database.py to watchlist.py   + Fix up some of the stuff in retrieveWatchlist in flaskApi.py like removing unnecessary username retrieval.   + Implementing add to watchlist functionalities * **TODOS (2.6):**   + Setup portfolio table in database   + Work on buy and sell functionality   + Setup frontend once buy and sell are done |
| Retro A | * Added “pair programming” approach to writing reports in the to to try section.   + Youshi in charge of overseeing * Completed Retrospective A |
| Pytest | Note that because we are making changes to backend, our tests will no longer pass because things have changed locations, etc |

**Action Items**

* Dom sends off the retro to tutor
* Finish setting up tables in database
* Complete TODOS in sprint 2 by Tuesday

**Meeting Date**

* W6: 22nd October

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Diary Update
* Assigning new sprint stories to team members
* Assign next scribe
* Merge watchlist and buy\_sell

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Diaries | * Everyone needs to keep it up to date till week 6 * **Progression Check:**   + Dom = Done   + Hazel = Week 5   + Shelbi = Week 5   + Youshi = Week 5   + Jet = Week 5 |
| Sprint Stories Assignment | * **Group Assignment:**   + Betting Predictions:     - Jet     - Dom     - Shelbi   + Investment Advisor:     - Hazel     - Youshi |
| New scribe assignment | * **New Scribe List:**   + Dom (Week 6)   + Shelbi (Week 7)   + Hazel (Week 8)   + Youshi (Week 9)   + Jet (Week 10) |
| Merging watchlist and buy\_sell | * **TODO:**   + Don’t forget to check why this.state.code prop is not working in stock\_detail line 54.   + When you search should say no search results instead of not owning anything.   + Fix Remove button function in watchlist |
| UX | **-**Buy/sell button on portfolio page as its more convenient for user  -When selling stocks from the details page, the user has to remember how many stocks they have from the previous page.  -When buying/selling, user doesn’t see how much money they currently have  - |
| errors | -also account page is broken when the capital value has changed after buying/selling  -Transaction displays a market price of 0: Process to reproduce Navigate to a stock details page Hit ‘back’ in browser Hit ‘forward’ in browser Try to buy sell, the market price is displayed as 0  I think the props is not being passed when this happens |

**Action Items**

* Add modal notification that you need to login into click buttons
* Work on new stories

**Meeting Date**

* W6: 25th October

**Participants**

* Dom
* Youshi
* Jet
* Shelby

**Agenda**

* Add modal notification that you need to login into click buttons
* Work on new stories

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Advisor Buy | Done backend, need to start on frontend  New module:secrets   * better random number generate * 5 companies to buy   Selling  - only tells you to sell stuff 25% of the time. |
| Investment Betting | * Betting button on the search page * Backend   + New table in DB, Stock code, user\_id, bet amount, predicted increase or decrease, startdate, price at the beginning of the bet, reward multiplier   + Change period to a static amount of one week   + Change the period prediction to just a prediction of it increasing or decreasing   + Player gets to choose their reward multiplier. In the event of a loss, the lost amount is double the reward multiplier. |

**Action Items**

* Investment Betting backend done by Thursday
* Advisor Buy frontend done by Tuesday

**Meeting Date**

* W7: 29th October

**Participants**

* Dom
* Youshi
* Hazel
* Jet
* Shelby

**Agenda**

* Investment Betting backend done by Thursday
* Advisor Buy frontend done by Tuesday

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Investment Betting | * Finished backend unit testing * Added a separate betting history table * Implement something that limits the amount of rows in the betting history table for each user |
| Investment Advisor | * FInished backend on Tuesday * Plan to finish frontend today * Add button on main page that takes to advisor page |
| Vlab | * Write a requirements script for packages to install |
| UX for buy/sell | * Buy modal: show how much you have in modal * Sell modal: Show how many stocks you have for that stock   + Tell Youshi/Hazel so they can add to Advisor |
| Diaries | * Dom: need to do * Youshi: Done * Jet: need to do * Hazel: Probably Done * Shelbi: need to do |

**Action Items**

* More comments
* Start Investment Betting frontend
* Test running system on CSE machines
* Discuss the Software Quality Report

**Meeting Date**

* W7:1st November

**Participants**

* Dom
* Youshi
* Jet
* Shelby

**Agenda**

* More comments
* Start Investment Betting frontend
* Test running system on CSE machines
* Discuss the Software Quality Report

|  |  |
| --- | --- |
| **Item** | **Notes** |
| Diaries | * Dom: Week7 * Youshi: done * Jet: Week -6 * Hazel: * Shelbi: Week7 |
| Comments | Going through adding comments while debuging |
| Investment Betting frontend | * Finished frontend * Enter Bet Amount can choose -1, needs to be fixed - Done * Need to check if bet\_amount is null - Done * Checks if user has enough capital for the punishment - * Need a link or button leads to the betting page |
| Test running system on CSE machines | * Done by Youshi * Everyone needs to test it * We can ignore warnings |
| Investment Advisor | * Done * Add extra reasoning to error message * Add some interesting sounding text * Add in capital to modal |
| Round() | * Need to set four decimal places |
| Discuss the Software Quality Report | <https://docs.google.com/document/d/1T3SrPzWIVsLJA_DuzMol0yYO0SEMwNds82qoC49c45M/edit?usp=sharing>   * Show special architecture design * User menu for set up system and functionalities - helper menu * Two weeks to go |
| Demo merge and practise | Wed night |

**Action Items**

* Demo merge and practise
* Dumping stuff into the report
* Add comments every where
* Test betting