SYSTEM INTERACTION DIAGRAMS

for

“Titan Trading”

Report 2: Part 1

By

Group #7

Steven Adler

Kristene Aguinaldo

Timothy Liu

Nicholas Lurski

Avanish Mishra

Safa Shaikh

Brooks Tawil

Kristian Wu

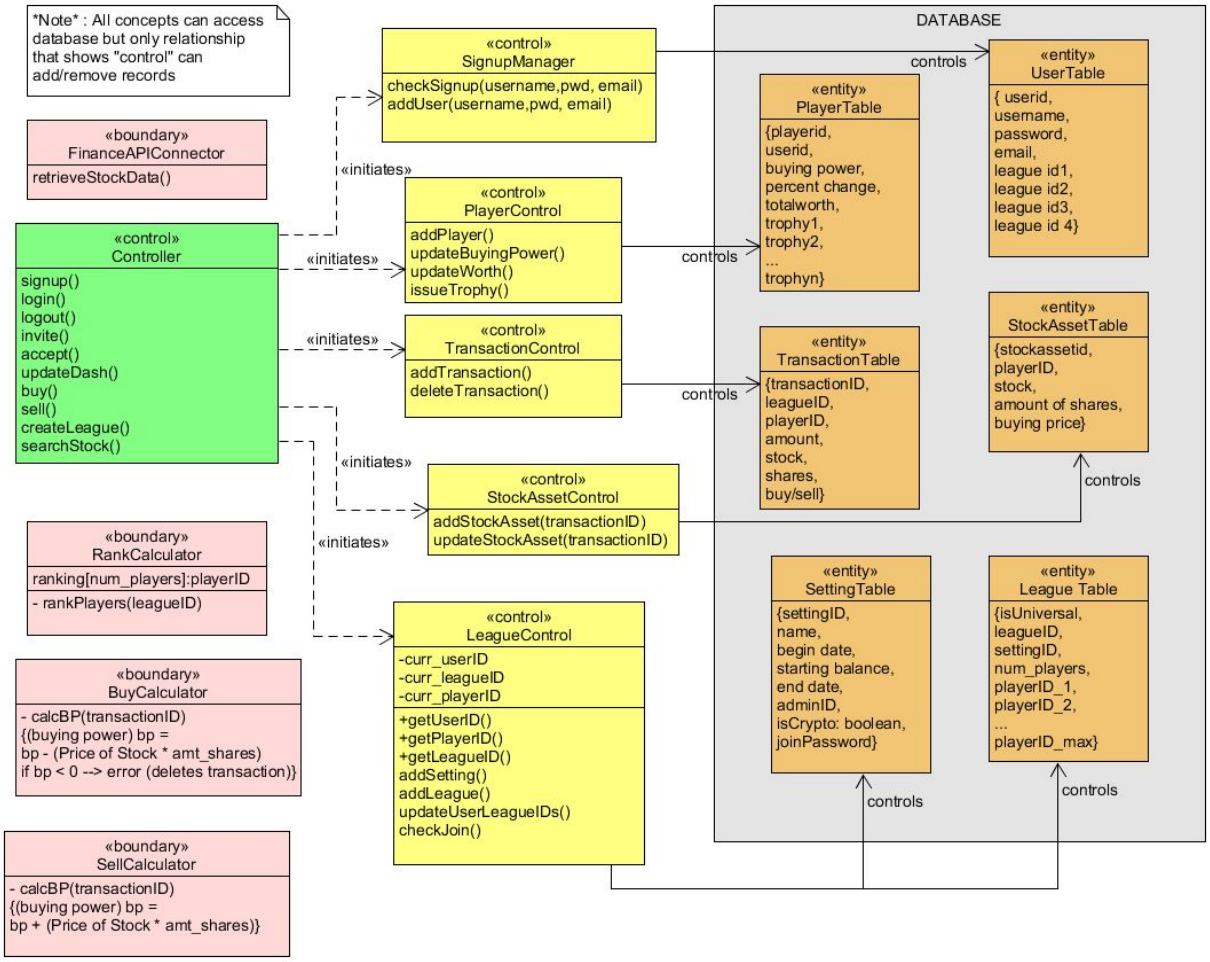
SYSTEM INTERACTION DIAGRAMS

Introduction

Titan Trading is a web app, built with Django and Python and will interface with a PostgreSQL database. The Financial API we will be using is AlphaVantage, which can obtain both stock and cryptocurrency data. Following is an in-depth analysis of use cases and system interaction between domain concepts.

Domain Model

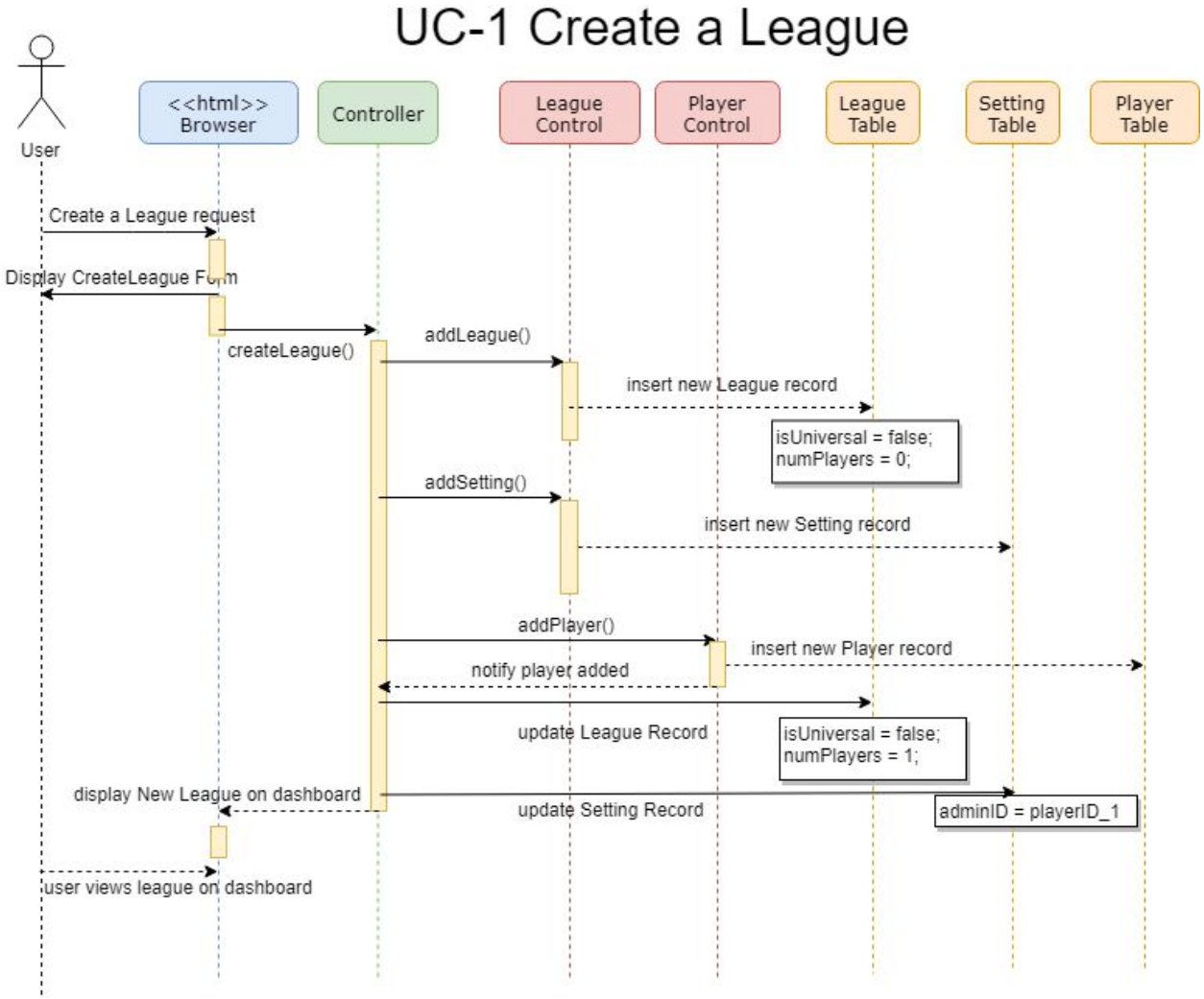
The following is our domain model. The controller is the main interface between the system and the browser with which the user interacts with. The controller allocates responsibilities to the various controls, which respectively update the database.



System Sequence Diagrams

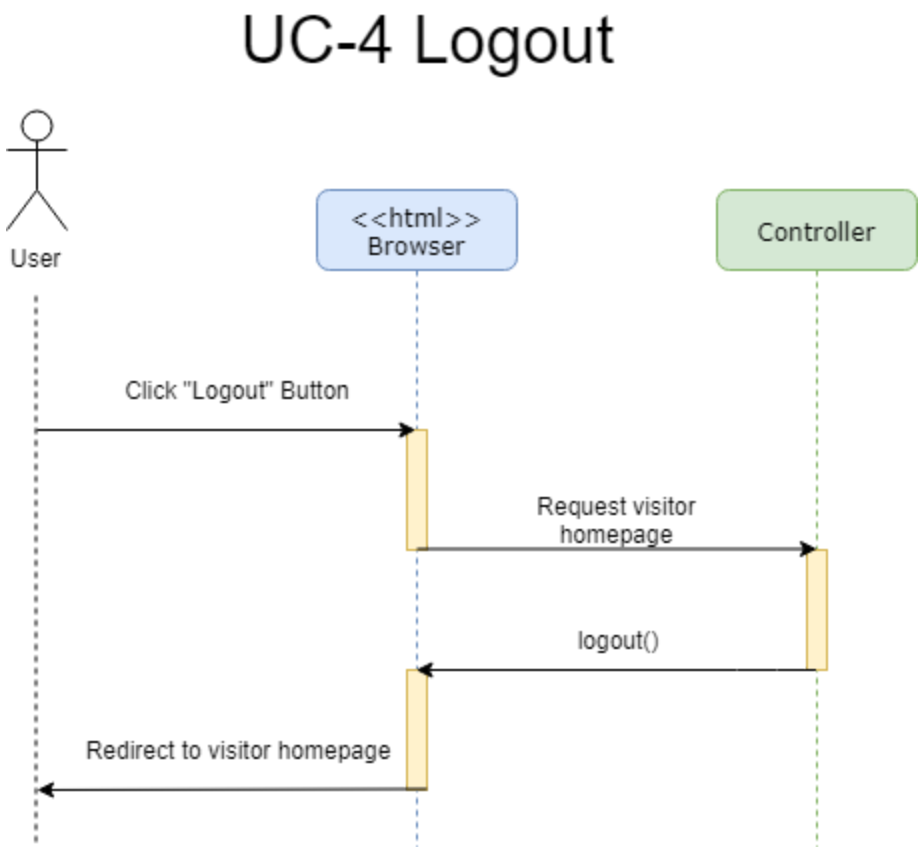
The following diagrams show the system processes of each fully dressed use case. These diagrams showcase the interaction of our domain concepts and how all user data is managed in a database.

UC-1 Create a League



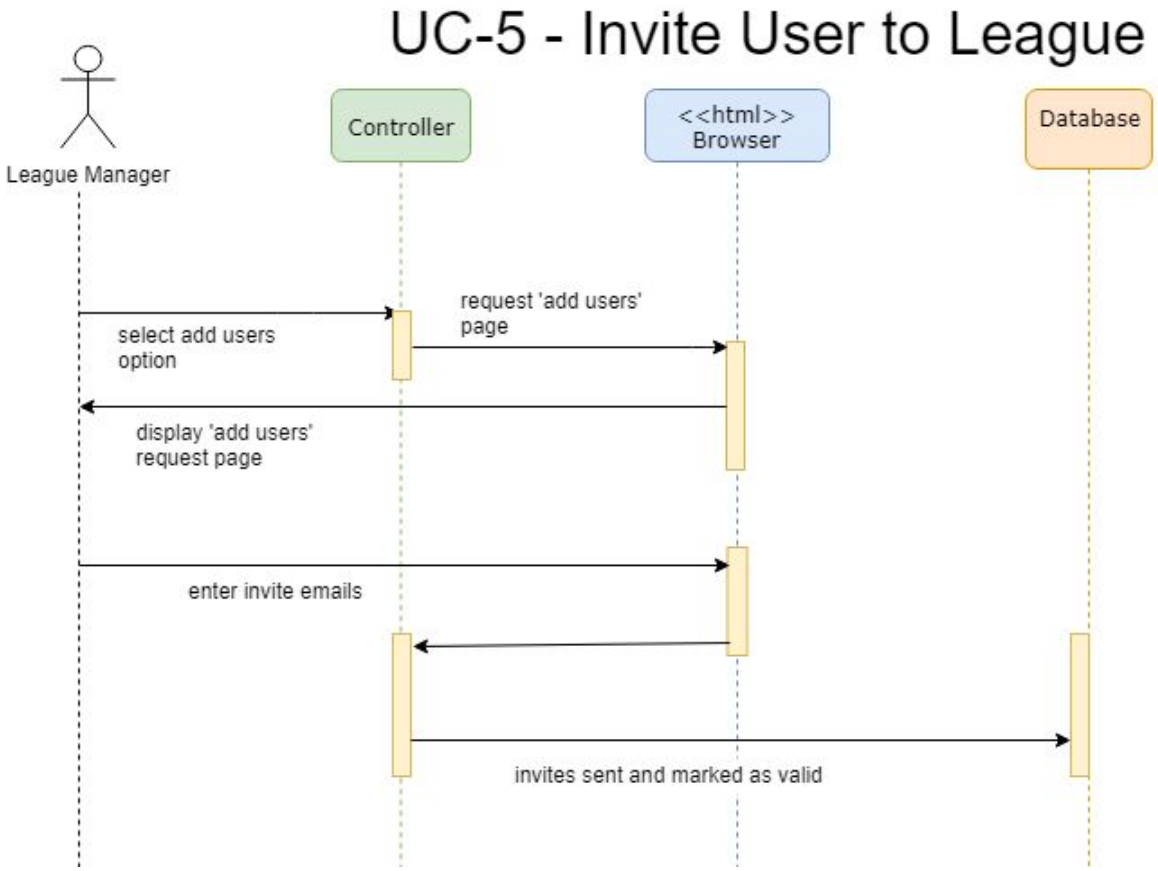
To create a league, a user enters all required data into a form. The controller keeps track of this data and sends it to the League Control and the Player Control. The League Control takes all data associated with the new league and new settings and updates the two league and setting tables in the database. The admin is now the first player, so that data needs to be updated in the player table as well. The player control will create a new player record. This record will now be associated with the prior created league and settings records.

UC-4 - Logout



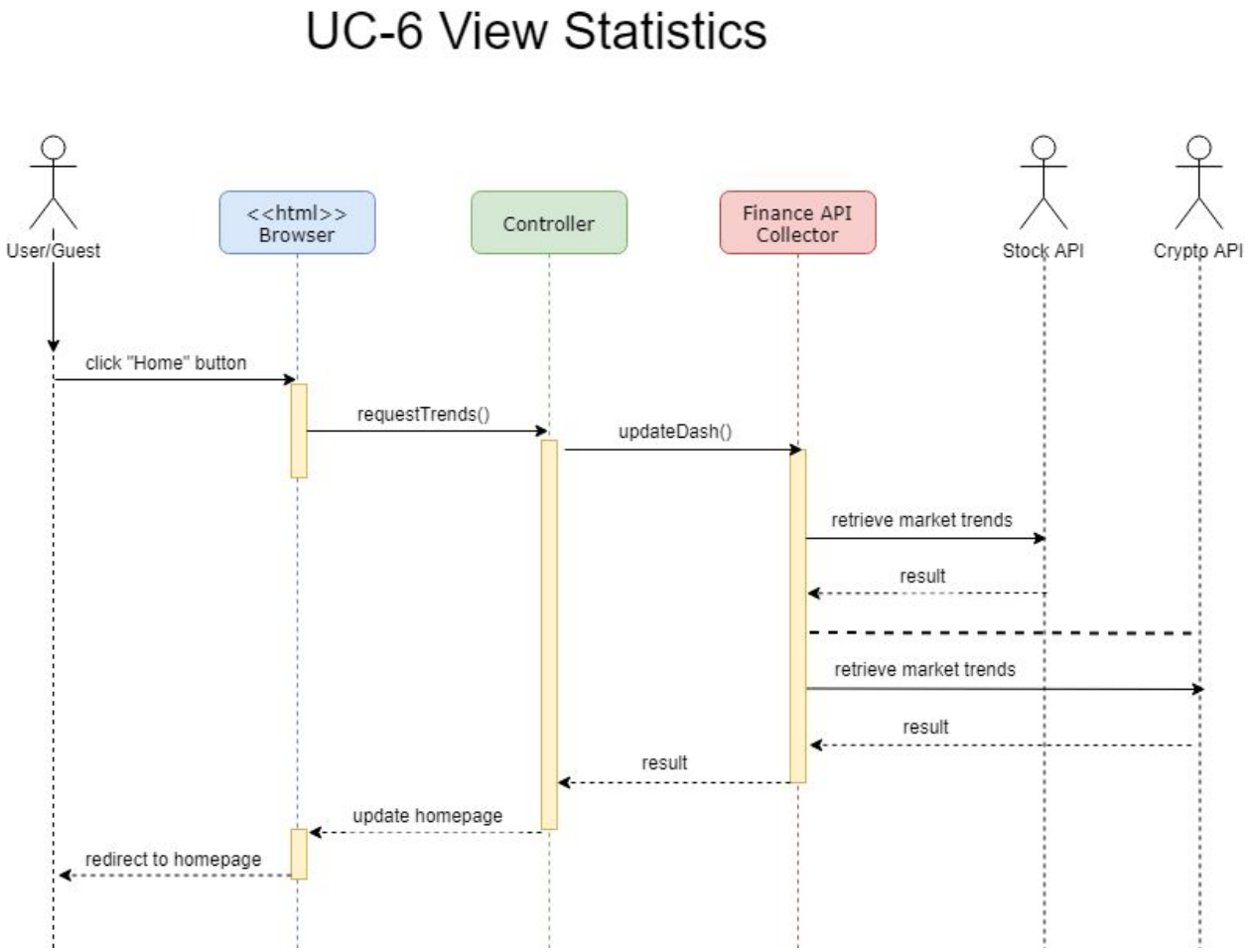
An essential feature is to distinguish visitors from registered users. Users become visitors by logging out and may proceed to initiate this process by clicking the logout button. The framework of the website allows all user data to be saved in the database when the user logs out and that state will be preserved until the user logs in again.

UC-5 - Invite User to League



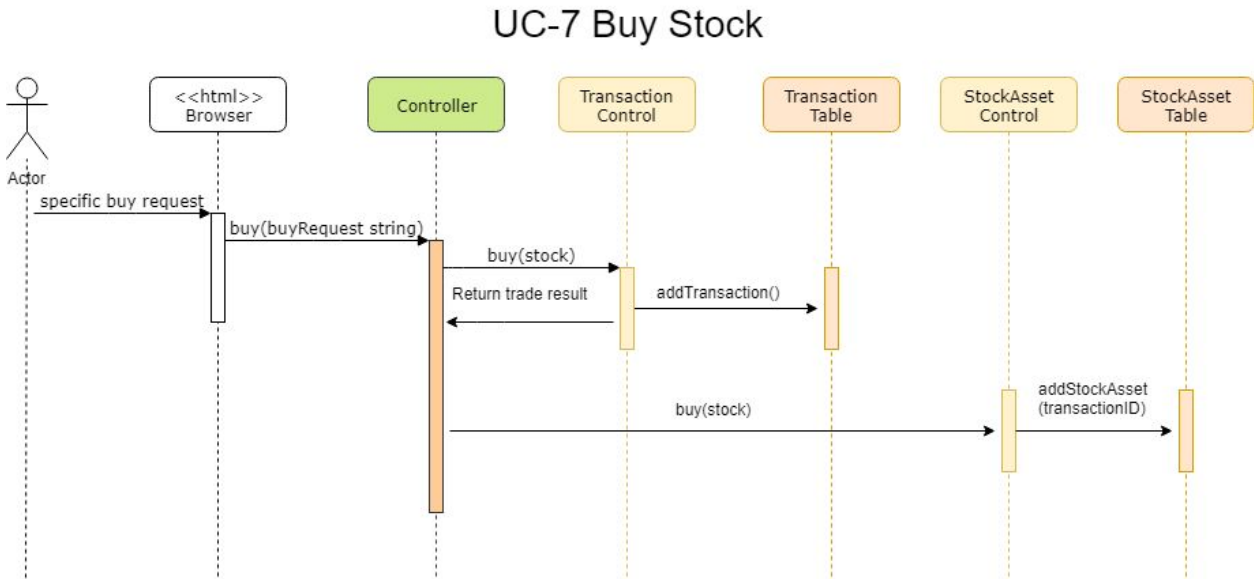
A league manager may decide to invite a user to join a league in order to participate in private competition with that user. The league manager will add users, using the “Add Users” option within the manager’s private league. The Controller will process this request and display the “Add Users” web page to which the league manager will fill in invitees’ emails. The Controller will process and send emails to the invited users. It will then interact with the Database, confirming that invitation emails were delivered.

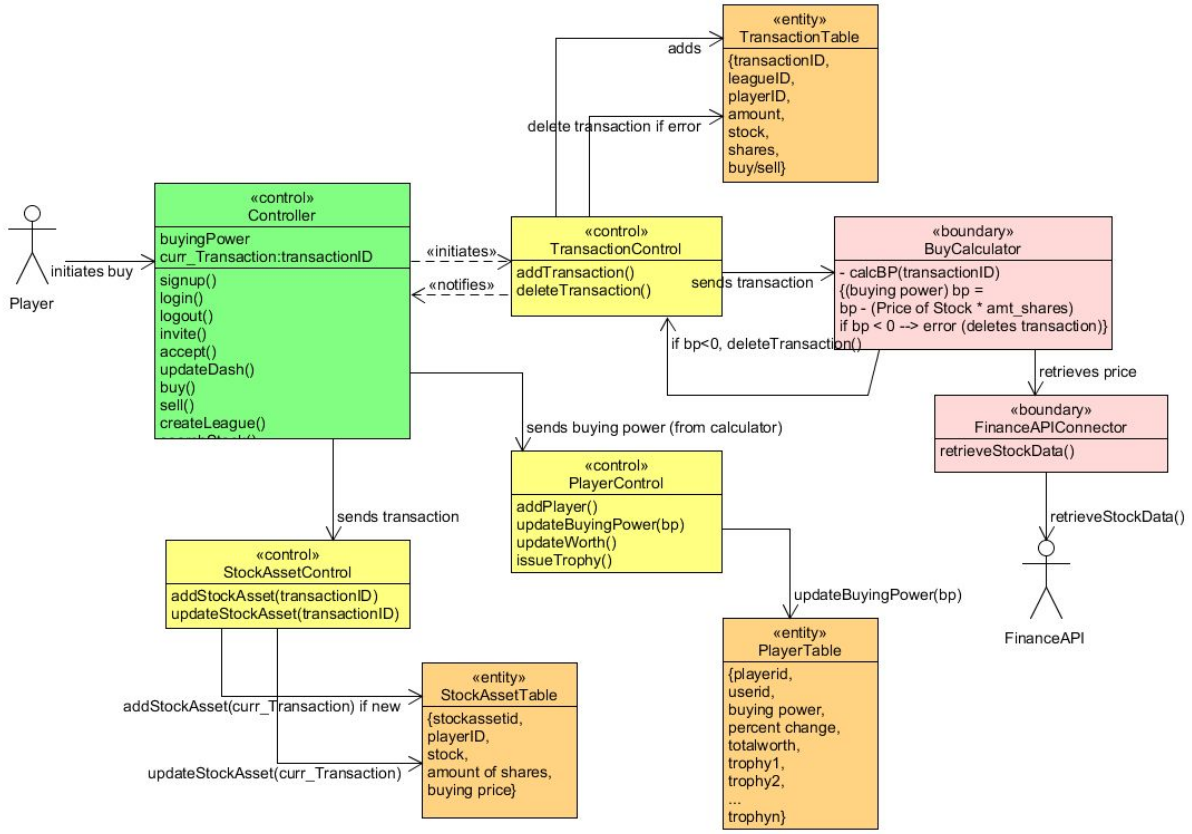
UC-6 - View Market Statistics



Both visitors and users can view market statistics in Titan Trading. The homepage will have updated information on current market trends and this data will be updated any time the user refreshes of clicks the homepage. Clicking the “home” button will retrieve updated data from the stock and cryptocurrency APIs.

UC-7 - Buying a Stock

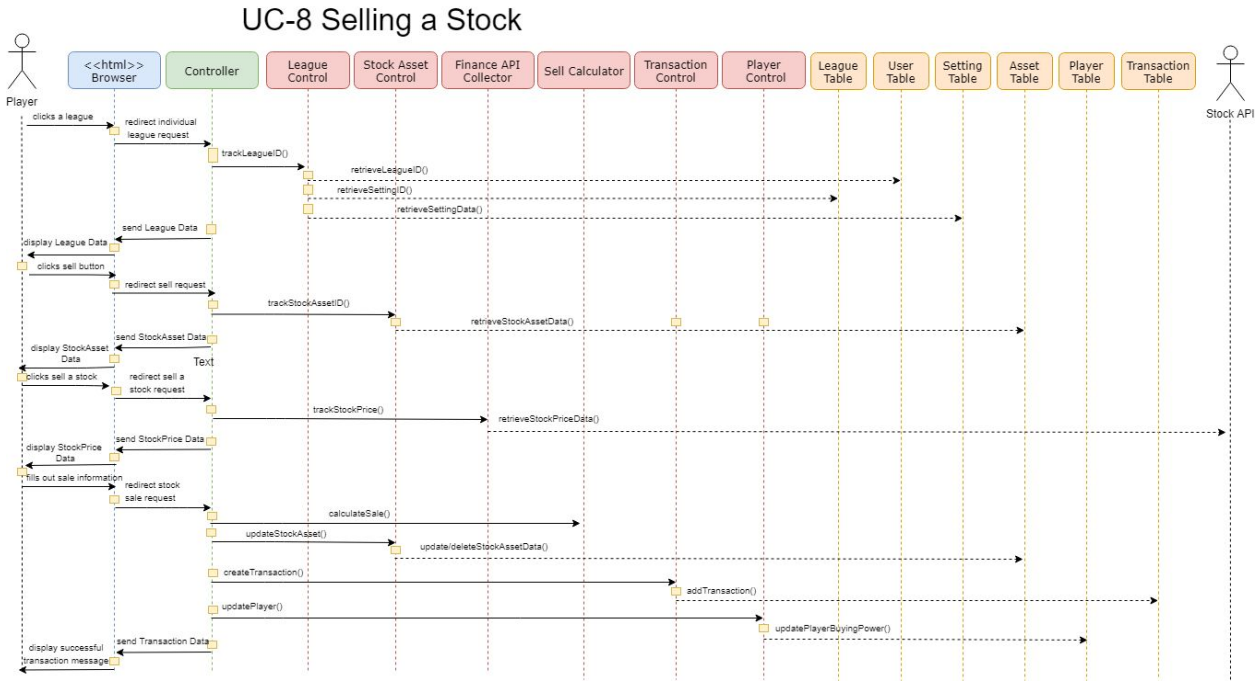


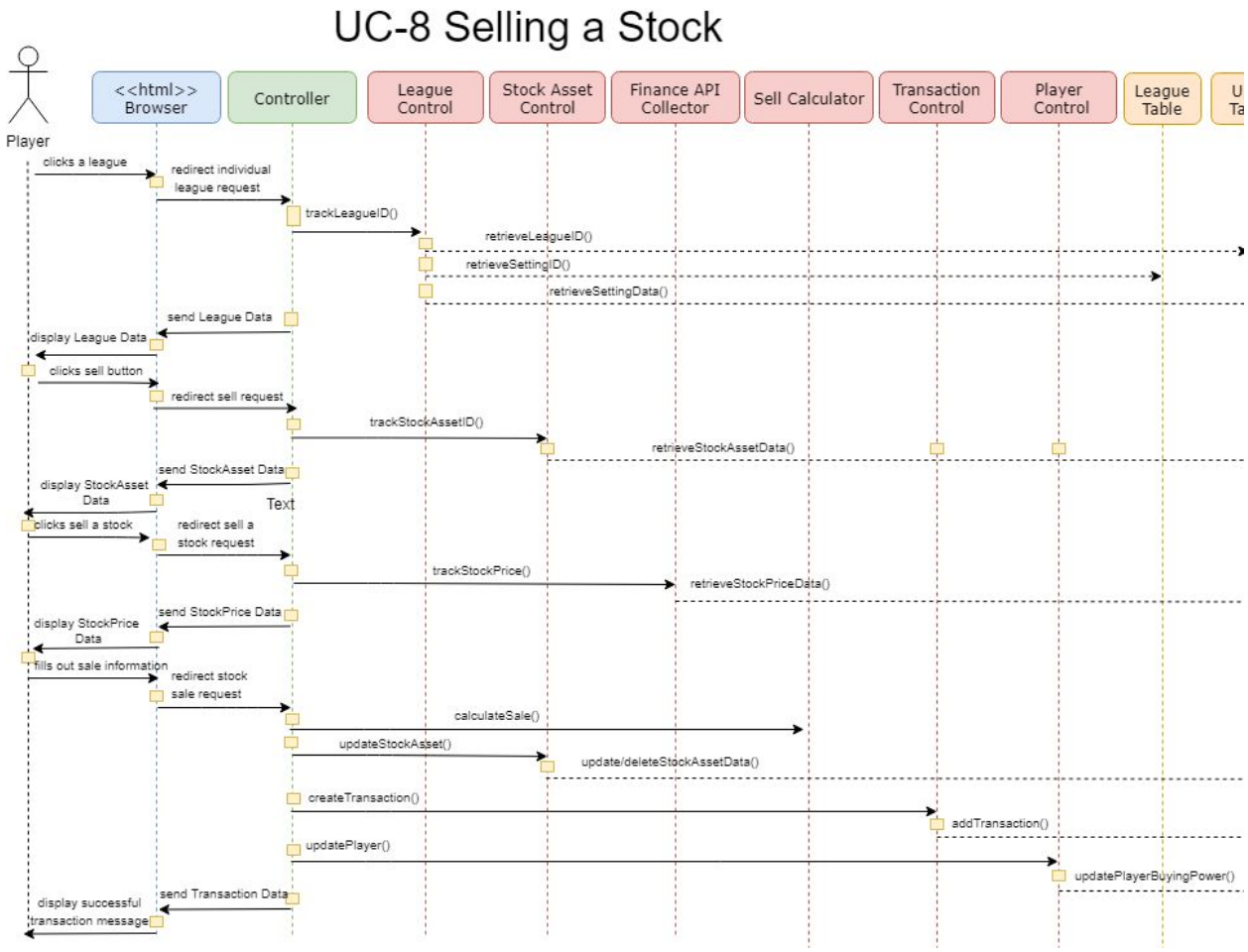


In order to buy a stock, a player must select which stock and how many shares of the stock he/she would like to buy. The controller will take this user input and send it to the Transaction Control, which will create a transaction record with updated prices from the Stock API (or Crypto API if in a

Crypto League). After the transaction is recorded (if successful), the Player Control will update player data, including buying power and player worth (total value of all assets). Stock Asset Control will also add a StockAsset record with the player's ID to keep track of player assets.

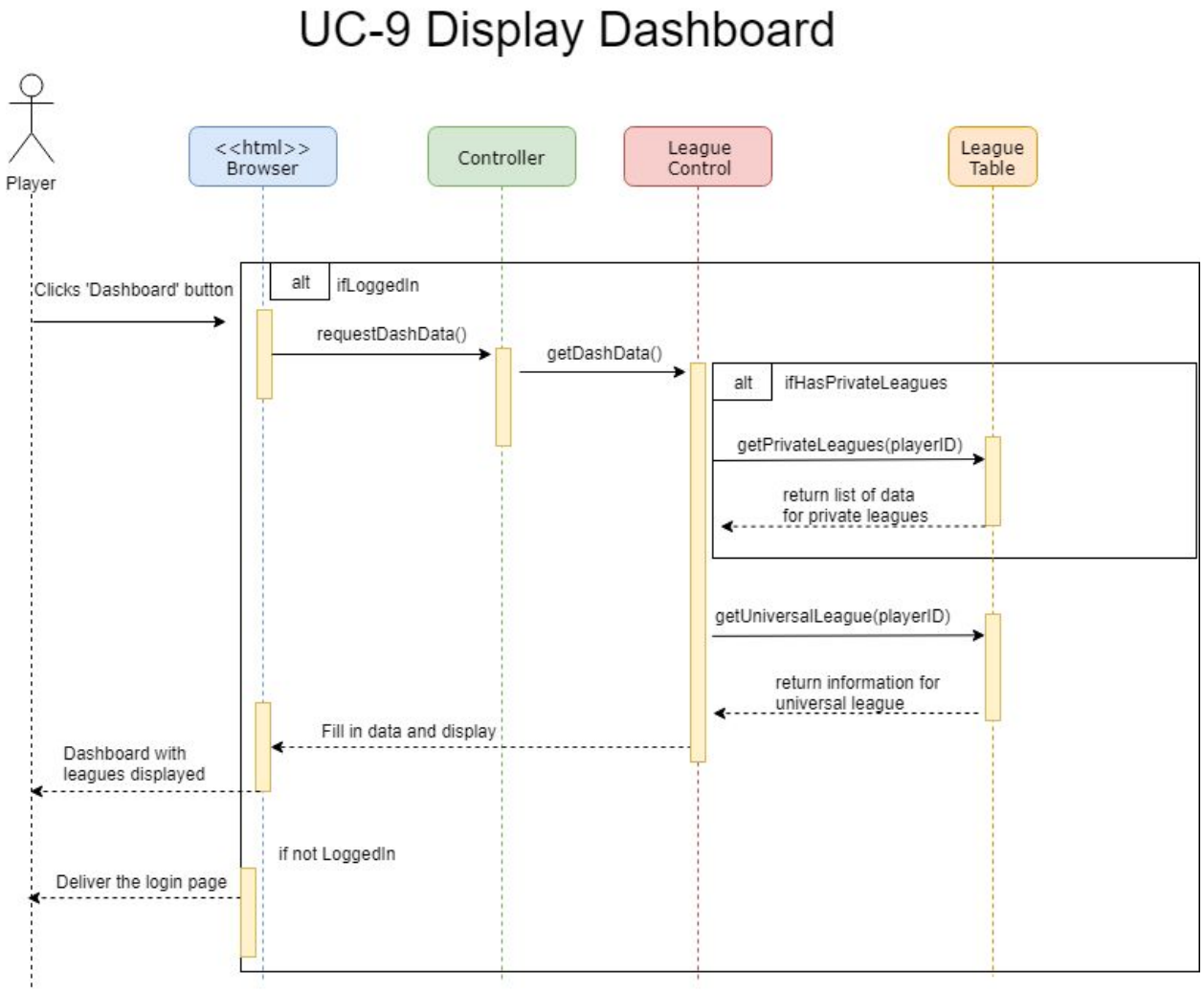
UC-8 - Selling a Stock





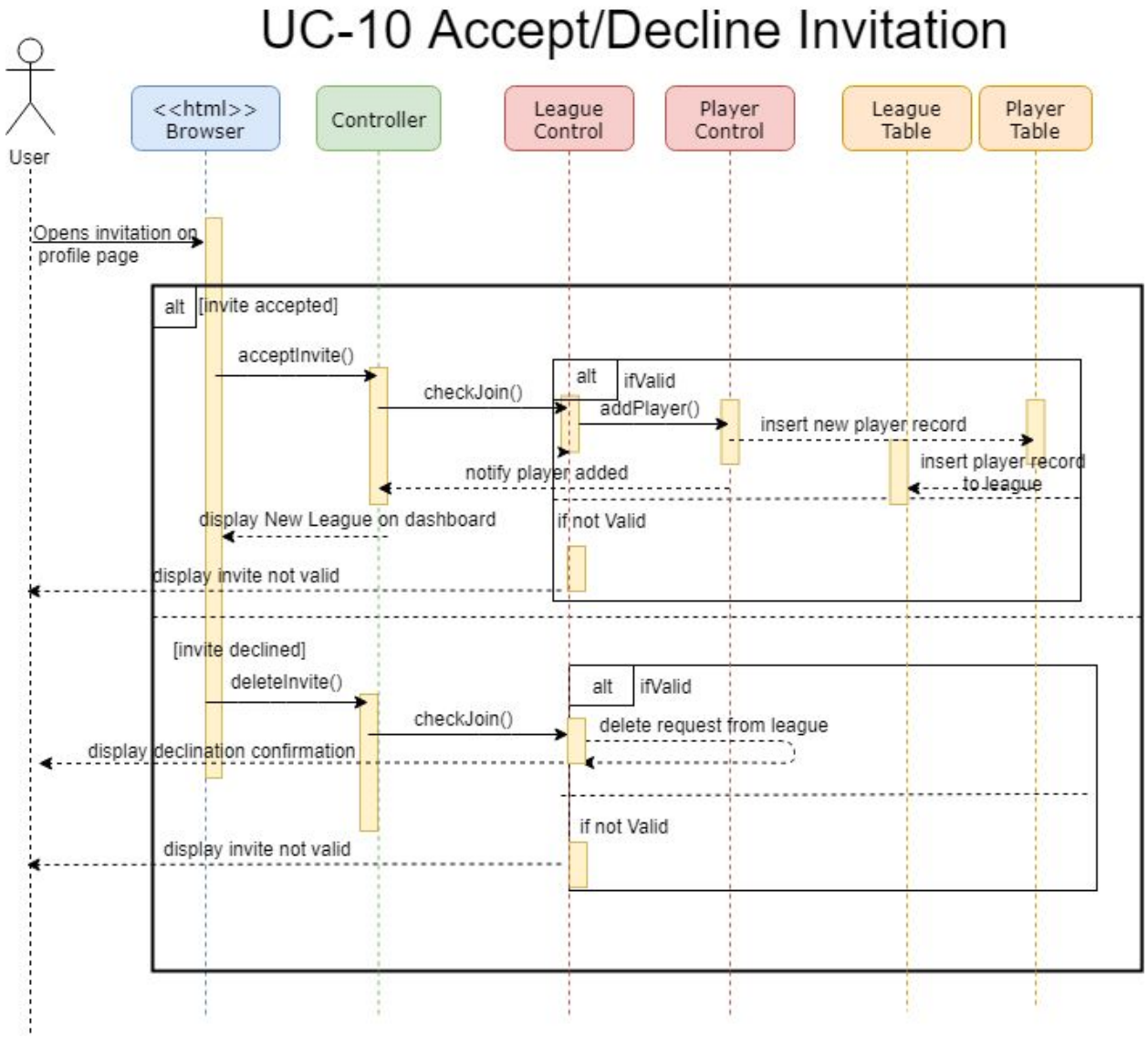
The above system sequence diagram shows the interactions of a player with the system to sell a stock. The player will start at the league dashboard page and will have to click a league where the system will redirect them to the individual league page of the league the player clicked. The redirection process requires the system controller to obtain all the necessary information about the individual league from the League Control. After being redirected to the individual league page with the displayed League Data, the player will have to click the sell button to declare that he wants to sell a stock from the current individual league. When the player clicks the sell button, the system controller will communicate to the stock asset control to obtain all the stocks the current player has within the individual league. The player will click a stock to sell and the system will retrieve the current market price of the stock from the Stock API and prompt the player with a sales form. The player will fill out the sales form with necessary transaction information and the system controller will have to communicate with the Sell Calculator to calculate the sales transaction, Stock Asset Control to updateStockAsset, Transaction Control to create a transaction record, and the player control to update the player's buying power.

UC-9 - Viewing Dashboard and League Leaderboards



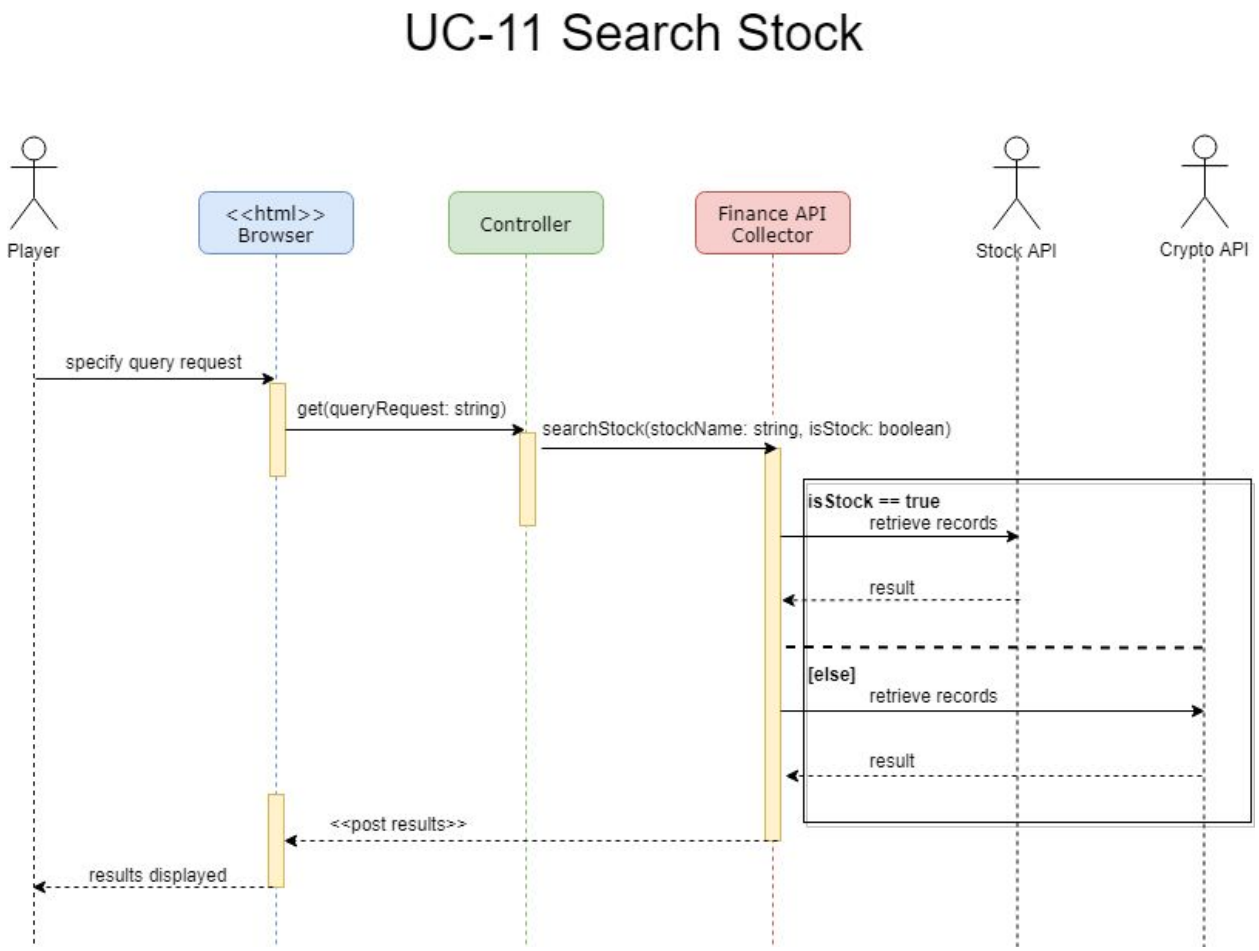
In order to view a user dashboard the user will click the dashboard button. The controller will then display all of the user's active leagues, which are associated with the leagueIDs stored in the user record. The league control will then identify which playerID corresponds to the current user and retrieve the data for that player. The league control will then transfer data to the browser which will display the user's player data in each league.

UC 10 - Accept/Decline Invitation



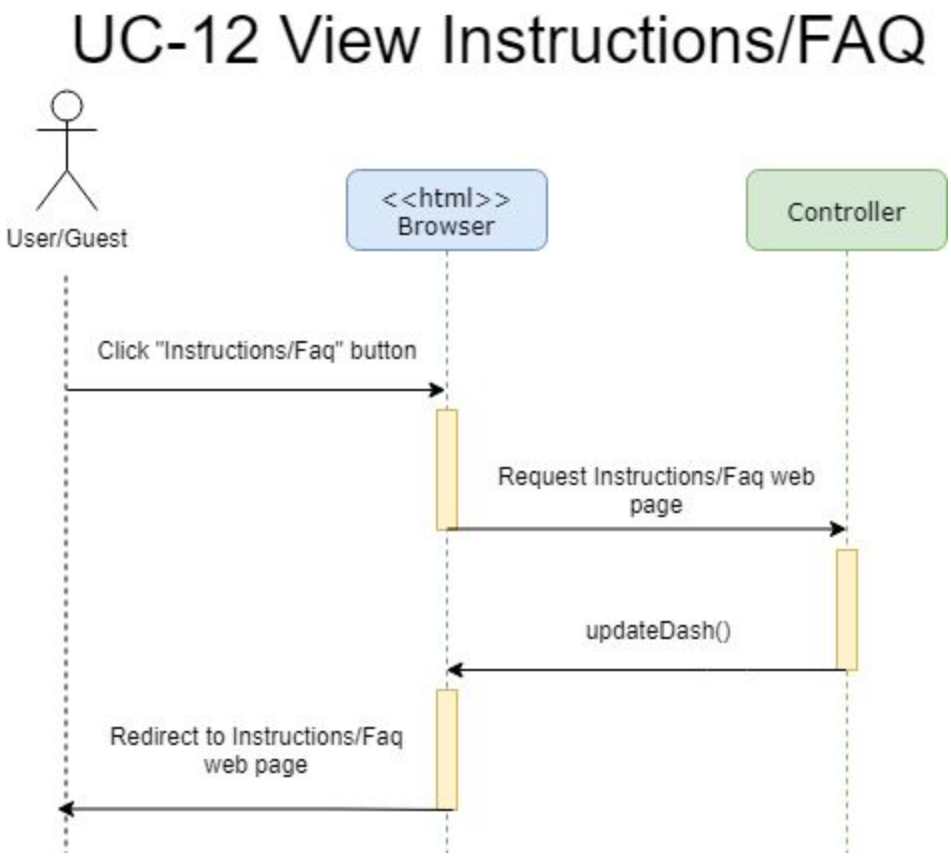
This use case controls whether or not a user invited to a private league accepts or declines the invitation. If they accept the invitation, the controller will communicate with the league control, league table, player control and player tables to create new player profiles and store the new data into the system. If they decline the invitation, the controller will communicate with the league controller to delete this request from the system. In both cases, the league control must check that the invitation request is valid.

UC-11 - Search Stock



A user may want to search for a stock or cryptocurrency in order to find information about its current market value before deciding to buy or sell their assets. When a player is searching for a stock, they initiate a query request which is sent to the Browser. The controller then processes the query request and redirects it to the Finance API Collector to search for most updated market information, which is updated by the minute. A boolean isStock determines if the Finance API Collector will retrieve records for the Stock API or the Crypto API. The result is then passed back to the Browser and displayed to the Player.

UC-12 - View Instructions/FAQ



In this use case, a user or guest may decide to view the instructions or FAQ page to learn more about Titan Trading and making investments. A user or guest initiates this use case by clicking the “Instructions/FAQ” button, which is processed by the Browser. The Controller processes this requests and subsequently updates the current webpage to the corresponding web page.

Plan of Work

The plan of work is for milestones and plans for after the submission of report #1.

The plan of work is divided into “Development Milestones” which serve as checkpoints for development and will determine our course of action as the semester continues. Each of these milestones are broken down into a list of tasks that need to be accomplished to accomplish the milestone.

Every milestone is classified as either “Major” or “Minor” depending on the feature or design that is laid out. Major milestones **must** be completed by the next demo date. Minor milestones are not **required** but they do merit working on should time allow it.

NOTE: The initial milestones are more detailed than the later milestones. Tese milestones and the plan of work will be adjusted and given more detail as certain developmental milestones are reached and demos are constructed.

Milestone #1 - Formalised Front End Wireframes for main pages - Major
Due 2/23/18

As part of the second report, wireframes and sketches of the web pages have to be formalised and put together. These are to be done not only for the report our self, but for the use of front end development. To begin we will set the first milestone to be only the most important pages so that implementation and programming on the front end can begin:

- Homepage(Not logged in)
- Homepage(Logged in)
- Create a League form
- League Dashboard

Milestone #2 - Front End implementation of pages- Major
Due 2/25/18

The report does not call for implementation, but the pages that are wireframed should nonetheless be implemented in some front end capacity. There may be no real data that is drawn, but have a working front end is the first step to using Django and bringing our concepts into a reality. The pages that should be implemented are as follows:

- Homepage(Not logged in)
- Homepage(Logged in)
- Create a League form
- League Dashboard

Milestone #3 - Formalised Front End Wireframes for all pages - Major

Due 2/25/18

The remaining pages have to be wireframed and designed for the report. However, they do not have to be implemented before the first report deadline. These are the remaining pages that should be wireframed:

- League Dashboard
- Individual league page
- Login Page
- Account Page
- Buy page
- Sell Page

First Demo Plan

Demo Due 3/27/18

Below are the goals that we have for the first demo. The first demo is due on March 27th and requires a working base with which to present with. The minimum needed for this demo is:

- Functioning front end UI that allows for navigation to different pages on the webpage.
- Functioning basic account interface. User authentication and preferences are a core function of the demo, and the demo must be able to demonstrate the core functions and preferences.

- A working global league interface that can demonstrate the concept of a general league. Private leagues and the invite system are non-critical at this point and will be implemented in the next parts of the project.

Demo Milestone #1 - User authentication and logging in/logging off - Major

Due 3/3/18

User authentication is directly tied to the backend and getting data from the back end. User authentication is an important first step in constructing a shared infrastructure. Milestone #5 is closely tied to this milestone. See **ST-1, ST-2** and the associated acceptance tests **AT-1, AT-2** for the tasks that must be completed and tests**.**

Demo Milestone #2 - Database setup and interface - Major

Due 3/3/18

The database backend serve as the key factor in the shared infrastructure that all aspects of the site will use. To that end the shared infrastructure must be developed and constructed to allow for tasks and stacks to be spread out and assigned.

Demo Milestone #3 - Writing up a FAQ and About Titan Page - Minor

Due 3/3/18

The front end pages such as the FAQ are simple and only serve to enhance the presentation of the demo. They are not a major requirement as they present no real functional aspects, but they are needed for good presentation.

Demo Milestone #4 - Connecting account page front end and user data - Major

Due 3/10/18

A user should be able to go to their account page and change the values for basic preferences and their profile picture. This must be accomplished after the shared database infrastructure is completed. The changes to profile picture, username, and other basic front-facing account information should be able to be changed, with the results being displayed on the front end.

Demo Milestone #5 - Connecting the buy page to the finance API - Major
Due 3/15/18

Displaying real data on the buy page, in a simple non-visual context, is the first step to getting our product to getting real-time information. This can visually be simple but so long as the API can properly be used to get correct numbers, the milestone can be accomplished.

Demo Milestone #6 - Connecting the Universal League to a user's portfolio - Major
Due 3/17/18

A given user's data should also include the their portfolio. In other words the list of assets that are owned by a user should be showing on the dashboard page for that particular user. At this point there will be no individual leagues implemented yet, so these will be a single portfolio in the single universal league.

Demo Milestone #7 - Connecting the Sell page to a user's portfolio - Major
Due 3/17/18

In the opposite direction, a user should be able to sell their stocks back to the market and have money move into their buying power. This is the dual of the buy page and moves money and assets in the opposite direction. This is a key functionality that will be extended to any league and is a key feature.

Demo Milestone #8 - Create a League form connects to back end and saves values - Major

Due 3/20/18

Once the buy and sell functionality is built in, users now have the features needed to form their own leagues. This starts with ensuring that the form that users fill out is not just a basic UI form. It should save the preferences that a user wants set to the back end. These won't have any effect yet, but the values for that league should be kept in the back end.

Demo Milestone #9 - League invitation interface over email - Major
Due 3/20/18

As user must be able to send out invites to their friends in order for them to join the league. The first milestone related with this would involve sending an email to the users that the league administrator (user who created the league) wants to invite. This milestone involves connecting our backend to some email service/API, which is important for other aspects later and merits its own milstone.

Demo Milestone #10 - Creation of new league is reflected on the front and back end - Major
Due 3/24/18

In creating a new league, the user is now able to have a good basis for using the application and the buy and its features and operations. The transaction pages that were implemented for the universal setup in previous milestones should be extended to also be able to be done in these private leagues. Once the user is able to create their own league, invite users, and buy/sell stocks. Then the first demo can be called a somewhat complete product. It may not be visually pleasing, there may be room for more features and visualization, but it's a start.

Demo Milestone #11 - Choreograph a demo - Major
Due 3/27/18

Any demo needs a script describing what needs to be done to fully demonstrate the product in its current state. This involves stepping through a list of steps and ensuring that ANY member of the team can explain what is going on in the product on any of its features.

Second Demo Plan
Due 5/2/18

Once the first demo is complete, the product should have basic functionality when it comes to sign up, authentication, description, transactions, league invitation and creation. The next demo is due on May 2nd and will focus on the enhanced functionality of our application. This is also the time we can extend to cryptocurrency and other features that we feel are appropriate to add. The goals for the 2nd demo are below:

- Adjust user interface of dashboard and other front facing pages to ensure that users have an easier time navigating and understanding the system.
- Adding data visualization and graphs to various pages on the site to enhance the user experience.
- Incorporate cryptocurrency and the ability to use cryptocurrency in a new league in addition to the classical stock market fantasy game.
- Create a full tutorial or training module so that users with minimal trading experience can get started and use out project as an education tool.
- Brainstorm new features and additions that can be made and implement them.

Final Milestone #1 - Revisit the User Interface and main Navigation - Major
Due 4/2/18

After receiving input from the demo it's important to evaluate and adjust the core construction of the project. If anything about the user experience is clunky or needs reworking, this is a good time to fix before more features are layered on.

Final Milestone #2 - Include Data visualization using static graphs - Major
Due 4/9/18

Markets are confusing and hard to understand at a quick glance. So a stack graph can deliver a better idea of trends and information, more so than a bunch of numbers set in rows. Static graphs are easier to implement and present the first step to having dynamic graphs which can adjust to certain options.

Final Milestone #3 - Connect all preferences on an account level to their respective uses - Major
Due 4/16/18

At this point the preferences should all have an actual effect on the user experience. This includes the league options, league creation options, and single user preferences. Without the ability to see settings change the product, the product suffers greatly. So it is important to ensure that everything is connected before adding non-critical features, which in turn will give more settings and preferences to add.

Final Milestone #4 - Implement a basic achievement system on a per user basis - Minor
Due 4/19/18

An achievement system is not a mission critical feature, but its implementation serves to enhance the user experience and give incentive to continue using the product and exploring its features. This can be done in a similar fashion to Snapchat, where the criteria is hidden and acquisition results in obtaining some award that can be displayed to other users.

Final Milestone #5 - Add cryptocurrency data with “Crypto-leagues” - Major
Due 4/23/18

An important addition that was desired was the addition of cryptocurrency to the typical stock trading game. At this point the project already has transaction infrastructure and the ability to well display information. The addition of cryptocurrencies complicates the product more horizontal than vertically. This means that current systems simply have to be expanded to include this new dynamic, rather than producing entire new systems from scratch.

Final Milestone #6 - Construct dynamic graphs that react to user input - Major

Due 4/29/18

The previously created static graphs can be enhanced to react to mousing over, option setting, and use different graph styles so that traders of all skill levels can get a better sense of the market, without leaving the application to go to third parties.

Final Milestone #7 - Construct a tutorial page for beginners - Minor

Due 4/29/18

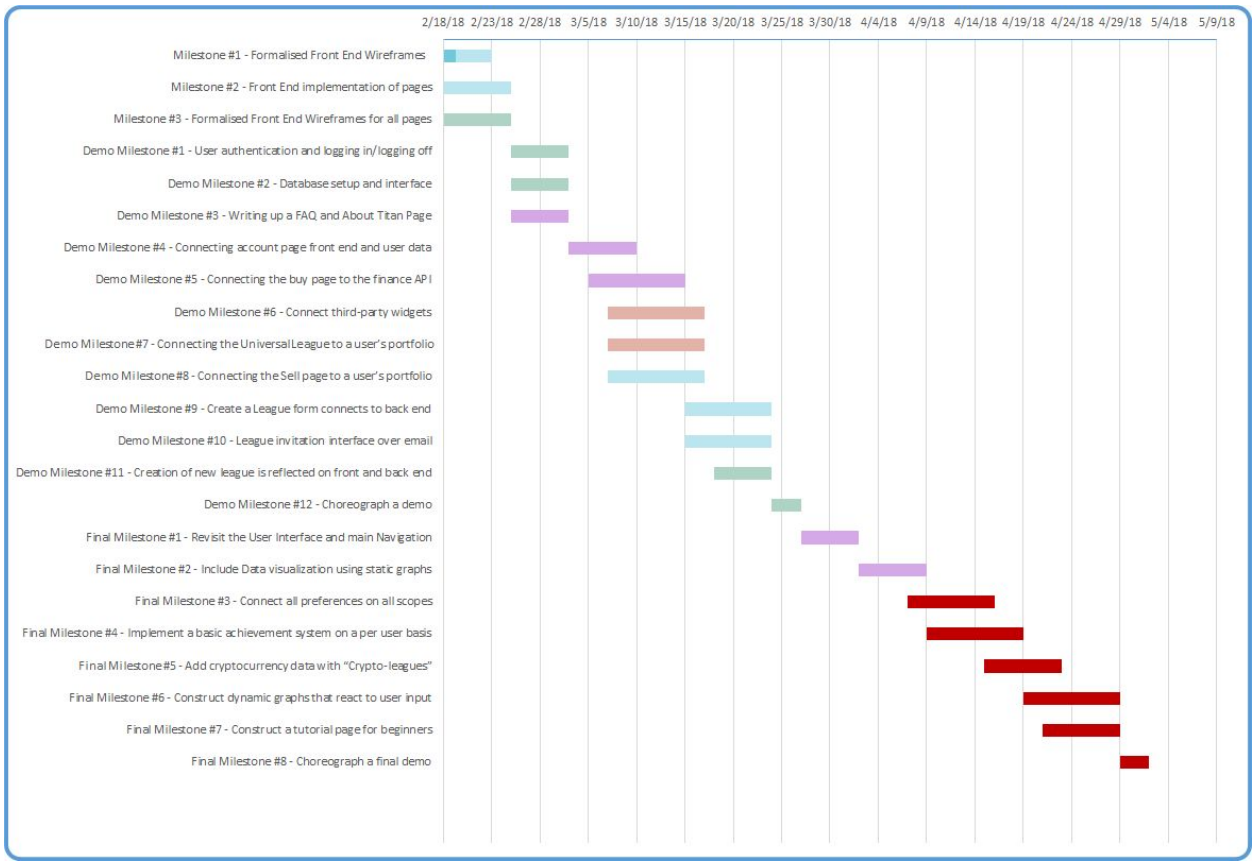
At this point in production the product should be mostly complete in functionality and in ability to fulfill the goals set out. Another goal for the product is to give it an educational use. The creation of a tutorial on top of what is already constructed can give a new UX opportunity and serves as another enhancement to the product.

Final Milestone #8 - Choreograph a final demo - Major

Due 5/2/18

This demo is the final part of the project, and making sure all is smooth is important when showing off in our final demonstration. Using the lessons that were learned in the first demo, we can showcase all of the implemented features of Titan Trading.

Gantt chart


2/18/18 2/23/18 2/28/18 3/5/18 3/10/18 3/15/18 3/20/18 3/25/18 3/30/18 4/4/18 4/9/18 4/14/18 4/19/18 4/24/18 4/29/18 5/4/18 5/9/18
Milestone #1 - Formalised Front End Wireframes
Milestone #2 - Front End implementation of pages
Milestone #3 - Formalised Front End Wireframes for all pages
Demo Milestone #1 - User authentication and logging in/logging off
Demo Milestone #2 - Database setup and interface
Demo Milestone #3 - Writing up a FAQ and About Titan Page
Demo Milestone #4 - Connecting account page front end and user data
Demo Milestone #5 - Connecting the buy page to the finance API
Demo Milestone #6 - Connect third-party widgets
Demo Milestone #7 - Connecting the Universal League to a user's portfolio
Demo Milestone #8 - Connecting the Sell page to a user's portfolio
Demo Milestone #9 - Create a League form connects to back end
Demo Milestone #10 - League invitation interface over email
Demo Milestone #11 - Creation of new league is reflected on front and back end
Demo Milestone #12 - Choreograph a demo
Final Milestone #1 - Revisit the User Interface and main Navigation
Final Milestone #2 - Include Data visualization using static graphs
Final Milestone #3 - Connect all preferences on all scopes
Final Milestone #4 - Implement a basic achievement system on a per user basis
Final Milestone #5 - Add cryptocurrency data with "Crypto-leagues"
Final Milestone #6 - Construct dynamic graphs that react to user input
Final Milestone #7 - Construct a tutorial page for beginners
Final Milestone #8 - Choreograph a final demo

Task Name	Start Date	End Date	Duration (Days)	Days Complete	Days Remaining	Percent Complete
Milestone #1 - Formalised Front End Wireframes	2/18/2018	2/23/2018	5	1.25	3.75	25%
Milestone #2 - Front End implementation of pages	2/18/2018	2/25/2018	7	0.00	7.00	0%
Milestone #3 - Formalised Front End Wireframes for all pages	2/18/2018	2/25/2018	7	0.00	7.00	0%
Demo Milestone #1 - User authentication and logging in/logging off	2/25/2018	3/3/2018	6	0.00	6.00	0%
Demo Milestone #2 - Database setup and interface	2/25/2018	3/3/2018	6	0.00	6.00	0%
Demo Milestone #3 - Writing up a FAQ and About Titan Page	2/25/2018	3/3/2018	6	0.00	6.00	0%
Demo Milestone #4 - Connecting account page front end and user data	3/3/2018	3/10/2018	7	0.00	7.00	0%
Demo Milestone #5 - Connecting the buy page to the finance API	3/5/2018	3/15/2018	10	0.00	10.00	0%
Demo Milestone #6 - Connect third-party widgets	3/7/2018	3/17/2018	10	0.00	10.00	0%
Demo Milestone #7 - Connecting the Universal League to a user's portfolio	3/7/2018	3/17/2018	10	0.00	10.00	0%
Demo Milestone #8 - Connecting the Sell page to a user's portfolio	3/7/2018	3/17/2018	10	0.00	10.00	0%
Demo Milestone #9 - Create a League form connects to back end	3/15/2018	3/24/2018	9	0.00	9.00	0%
Demo Milestone #10 - League invitation interface over email	3/15/2018	3/24/2018	9	0.00	9.00	0%
Demo Milestone #11 - Creation of new league is reflected on front and back end	3/18/2018	3/24/2018	6	0.00	6.00	0%
Demo Milestone #12 - Choreograph a demo	3/24/2018	3/27/2018	3	0.00	3.00	0%
Final Milestone #1 - Revisit the User Interface and main Navigation	3/27/2018	4/2/2018	6	0.00	6.00	0%
Final Milestone #2 - Include Data visualization using static graphs	4/2/2018	4/9/2018	7	0.00	7.00	0%
Final Milestone #3 - Connect all preferences on all scopes	4/7/2018	4/16/2018	9	0.00	9.00	
Final Milestone #4 - Implement a basic achievement system on a per user basis	4/9/2018	4/19/2018	10	0.00	10.00	
Final Milestone #5 - Add cryptocurrency data with "Crypto-leagues"	4/15/2018	4/23/2018	8	0.00	8.00	
Final Milestone #6 - Construct dynamic graphs that react to user input	4/19/2018	4/29/2018	10	0.00	10.00	
Final Milestone #7 - Construct a tutorial page for beginners	4/21/2018	4/29/2018	8	0.00	8.00	
Final Milestone #8 - Choreograph a final demo	4/29/2018	5/2/2018	3	0.00	3.00	

Key:

Calculated Cell	Manual Entry Cell
These cells will be automatically calculated based on the inputs on other cells.	These cells require manual input so the calculated cells have data to work with.

References

Barker, Jonathan Todd. “Why Is Bitcoin's Value So Volatile?” *Investopedia*, 2 Jan. 2018, www.investopedia.com/articles/investing/052014/why-bitcoins-value-so-volatile.asp. [Online, accessed 17 February, 2018]

Fontinelle, Amy. “Perfect Competition.” *Investopedia*, 29 Jan. 2018, www.investopedia.com/terms/p/perfectcompetition.asp. [Online, accessed 17 February, 2018]

I. Marsic, Software Engineering. New Brunswick, USA: Ivan Marsic, 2012

Investopedia. “Efficient Market Hypothesis - EMH.” *Investopedia*, 28 Sept. 2016, www.investopedia.com/terms/e/efficientmarkethypothesis.asp. [Online, accessed 17 February, 2018]

Marr, Bernard. “A Short History Of Bitcoin And Crypto Currency Everyone Should Read.” *Forbes*, Forbes Magazine, 6 Dec. 2017, www.forbes.com/sites/bernardmarr/2017/12/06/a-short-history-of-bitcoin-and-crypto-currency-everyone-should-read/#35cfdb633f27. [Online, accessed 17 February, 2018]

“The Paramount Investments League.” *Rutgers University*, 8 May. 2014, http://www.ece.rutgers.edu/~marsic/books/SE/projects/TradingLeague/2014-g1-report3.pdf. [Online, accessed 17 February, 2018]

“Rutgers University Investing.” *Rutgers University*, 12 May. 2014, http://www.ece.rutgers.edu/~marsic/books/SE/projects/TradingLeague/2014-g2-report3.pdf. [Online, accessed 17 February, 2018]

“User & Design Specifications for “Capital Games.” *Rutgers University*, 3 May. 2013, http://www.ece.rutgers.edu/~marsic/books/SE/projects/TradingLeague/2013-g2-report3.pdf. [Online, accessed 17 February, 2018]