

FantasyStock - CSCI 150 FALL 2022 Project

[FantasyStock - CSCI 150 FALL 2022 Project](#)

[Introduction](#)

- [1.1 Purpose](#)
- [1.2 Document Conventions](#)
- [1.3 Intended Audience and Reading Suggestions](#)
- [1.4 Product Scope](#)
- [1.5 References](#)

[Overall Description](#)

- [2.1 Product Perspective](#)
- [2.2 Product Functions](#)
- [2.3 User Classes and Characteristics](#)
- [2.4 Operating Environment](#)
- [2.5 Design and Implementation Constraints](#)
- [2.6 User Documentation](#)
- [2.7 Assumptions and Dependencies](#)

[External Interface Requirements](#)

- [3.1 User Interfaces](#)
- [3.2 Hardware Interfaces](#)
- [3.3 Software Interfaces](#)
- [3.4 Communications Interfaces](#)

[System Features](#)

[Other Nonfunctional Requirements](#)

- [5.1 Performance Requirements](#)
- [5.2 Safety Requirements](#)
- [5.3 Security Requirements](#)
- [5.4 Software Quality Attributes](#)
- [5.5 Business Rules](#)

[Other Requirements](#)

[Appendix A: Glossary](#)

[Appendix B: Analysis Models](#)

[Use Case Diagram](#)

[4.2 Site Map](#)

[Appendix C: To Be Determined List](#)

Introduction

1.1 Purpose

FantasyStock is a web-based game that allows users to create groups where they can compete against each other by picking stocks in the stock market and seeing which collection of stocks grows the most within a defined amount of time. We call these groups leagues, and in addition to creating them, we will also be joining existing leagues. We'll also allow communication within each league, the ability to add people, and customization of your profile.

1.2 Document Conventions

Every requirement statement is to have its own priority.

1.3 Intended Audience and Reading Suggestions

Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type. This document is intended for developers, project managers, and testers. The rest of this SRS will contain information relevant to this product. Most pertinent sections for developers will have to read the sections 1, 2.1, 2.2, 2.4, 2.5, 2.7, 3, 4, 5. Most pertinent sections for project managers will have to read the entire document. Most pertinent sections for testers will have to read 1, 2.2, 2.3, 4, 5.

1.4 Product Scope

FantasyStock is an application for anyone interested in the stock market; these people can range from people who have never heard the term stock market to professionals who deal with it for a living. We provide the ability for the unsure to test hypothetical situations in a gamified manner. Users will be able to create an account, create or join a league, and be able to communicate with others within leagues.

1.5 References

See section appendix A, B, C for references.

Overall Description

2.1 Product Perspective

This product is a new, self-contained product.

2.2 Product Functions

The major functions for the product are logging in, logging out, creating leagues, joining leagues, finding leagues, commenting on leagues, adding friends, deleting friends, messaging friends, and buying items.

2.3 User Classes and Characteristics

The main two user classes will be people with low experience in the stock market and people with high experience in the stock market. The requirements for both of these classes will be the same, because the purpose of FantasyStock is to provide an easy-to-use, gameified version of the stock market.

2.4 Operating Environment

For the server, we will likely be deploying on a free low resource server running some flavour of linux likely ubuntu or on the cloud. For the client, we will likely be targeting any computer that is capable of running a modern web-browser, such as chrome. This will exclude ultra small devices such as smart watches.

2.5 Design and Implementation Constraints

This project will be created using Mongodb, Express, React, and Node. Additionally, this product will have to run on very few resources.

2.6 User Documentation

No user documentation is planned with this product.

2.7 Assumptions and Dependencies

Assumptions and dependencies for the frontend on this project are axios, react, react-dom, react-redux, react-router-dom, react-scripts, and web-vitals. Assumptions

and dependencies for the backend on this project are body-parser, connect-mongo, dotenv, express, express-session, mongoose, mongoose-findorcreate, node-schedule, passport, passport-google-oauth20, and request.

External Interface Requirements

3.1 User Interfaces

The software components for which a user interface is needed will start with a way to navigate through all the components. Next will be a way to create leagues, search for leagues, view leagues, and join leagues. After this we will need a way to create users, sign in, sign out, edit profile, add friend, delete friend, and view friend.

3.2 Hardware Interfaces

This product is designed to be used on modern computers and mobile devices capable of running a modern web browser, such as google chrome. This does not include ultra small devices such as smart watches.

3.3 Software Interfaces

Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

Software interfaces for react on this project are axios, react, react-dom, react-redux, react-router-dom, react-scripts, and web-vitals. Software interfaces for node on this project are body-parser, connect-mongo, dotenv, express, express-session, mongoose, mongoose-findOrCreate, node-schedule, passport, passport-google-oauth20, and request. Software interfaces for mongodb would be mongoose. Software interfaces for the stock API would be through alpha vantage.

3.4 Communications Interfaces

Communication functions associated with this product would be through a google account and Gmail most likely though a browser. The communication standards will be HTTP and WebSocket.

System Features

Title	Description	Priority	Stimulus	Functional Requirements	ID
User Registration	If a user intends to use FantasyStock , they must register with us, and this will be done using their google account.	High	1. User clicks the register/sign-in button	N/A	FR1
User Login	If a user has registered, the user should be able to log in.	High	1. User clicks the register/sign-in button	FR1	FR2
User Logout	Let the user safely logout of their profile.	High	1. User logs in 2. User clicks the sign-out button	FR1	FR3
Create League	Let users create leagues so that they can participate in games.	High	1. User logs in 2. User clicks create league button	FR1	FR4

View League	Let users view a league so that they can view a game.	High	1. User logs in 2. User views a league	- FR1 - FR4	FR5
Join League	Let users join leagues so that they can participate in a game.	High	1. User logs in 2. User views a league 3. User clicks join league button	- FR1 - FR4 - FR5	FR6
View Registered Leagues	Let users view a league so that they can view a game.	Medium	1. User logs in 2. User views registered league	- FR1 - FR4	FR7
Search Active Leagues	Let users search for an active league so they can view.	Low	1. User logs in 2. User searches leagues	- FR1 - FR4	FR8

Post Comment on League	Let users comment on a league so they can communicate .	Low	1. User logs in 2. User Views a league 3. Posts comment on league	- FR1 - FR4	FR9
Edit Comment on League	Let users edit a comment on a league.	Low	1. User logs in 2. User Views a league 3. Posts comment on league 4. Edit comment on league	- FR1 - FR4 - FR9	FR10
Delete Comment on League	Let users delete a comment on a league.	Low	1. User logs in 2. User Views a league 3. Posts comment on league 4. Delete comment on league	- FR1 - FR4 - FR9	FR11

Reply to Comment on League	Let users reply to a comment on a league.	Low	1. User logs in 2. User Views a league 3. Posts comment on league 4. Reply to a comment on league	- FR1 - FR4 - FR9	FR12
View Profile	Let users view their profile.	Medium	1. User logs in 2. User Views their profile	- FR1	FR13
Edit Profile	Let users edit their profile.	Low	1. User logs in 2. User Views their profile 3. User Edits their profile	- FR1 - FR13	FR14
View Other People's Profile	Let users view other people's profile.	Low	1. User logs in 2. User views other person's profile	- FR1	FR15

Manage Friends	Let users view, and delete any friend. Allow users to add friend by ID. Allows users to accept or decline friend requests.	Medium	1. User logs in 2. User clicks manage friends	- FR1 - FR20 - FR22	FR16
Send Friend Request	Let users send friend request to other people.	Low	1. User logs in 2. User views other person's profile 3. User clicks add friend button (Alt) 2. User clicks manage friends (Alt) 3. User adds friend with friend ID	- FR1 - FR15 - FR16	FR17

Accept Friend Request	Let users accept friend request.	Low	1. User logs in 2. User clicks manage friends 3. User clicks accept a friend request	- FR1 - FR16 - FR17	FR18
Decline Friend Request	Let users decline a friend request.	Low	1. User logs in 2. User clicks manage friends 3. User clicks decline friend request	- FR1 - FR16 - FR17	FR19
Add Friend	Let users add friend by ID.	Low	1. User logs in 2. User clicks manage friends 3. User clicks add friend by ID	- FR1 - FR16	FR20

Delete Friend	Let users delete a friend.	Low	1. User logs in 2. User clicks manage friends 3. User clicks delete friend	- FR1 - FR16	FR21
View Store	Let users view the store.	Low	1. User logs in 2. User clicks store	- FR1	FR22
Purchase From Store	Let users purchase from store.	Low	1. User logs in 2. User clicks store 3. User clicks purchase item	- FR1 - FR22	FR23
Toggle Dark Mode	Let users toggle dark mode.	Low	1. User logs in	- FR1	FR24

Other Nonfunctional Requirements

5.1 Performance Requirements

The product should be able to perform on most devices capable of running a modern browser, so this product on the client side would need to perform as quickly as possible.

5.2 Safety Requirements

No safety requirements are needed for this product.

5.3 Security Requirements

Due to the limited information this product uses, there should not be any privacy requirements. Due to the limited information this product uses, the security for this product will be minimal. The user identity authentication requirement will come OAuth20.

5.4 Software Quality Attributes

Quality characteristic for this product include reliability, robustness, testability, and usability. Ideally this product will be reliable and robust when it comes to usability. Additionally, this product should be highly testable to ensure reliability and robustness.

5.5 Business Rules

Any roles may perform any functions.

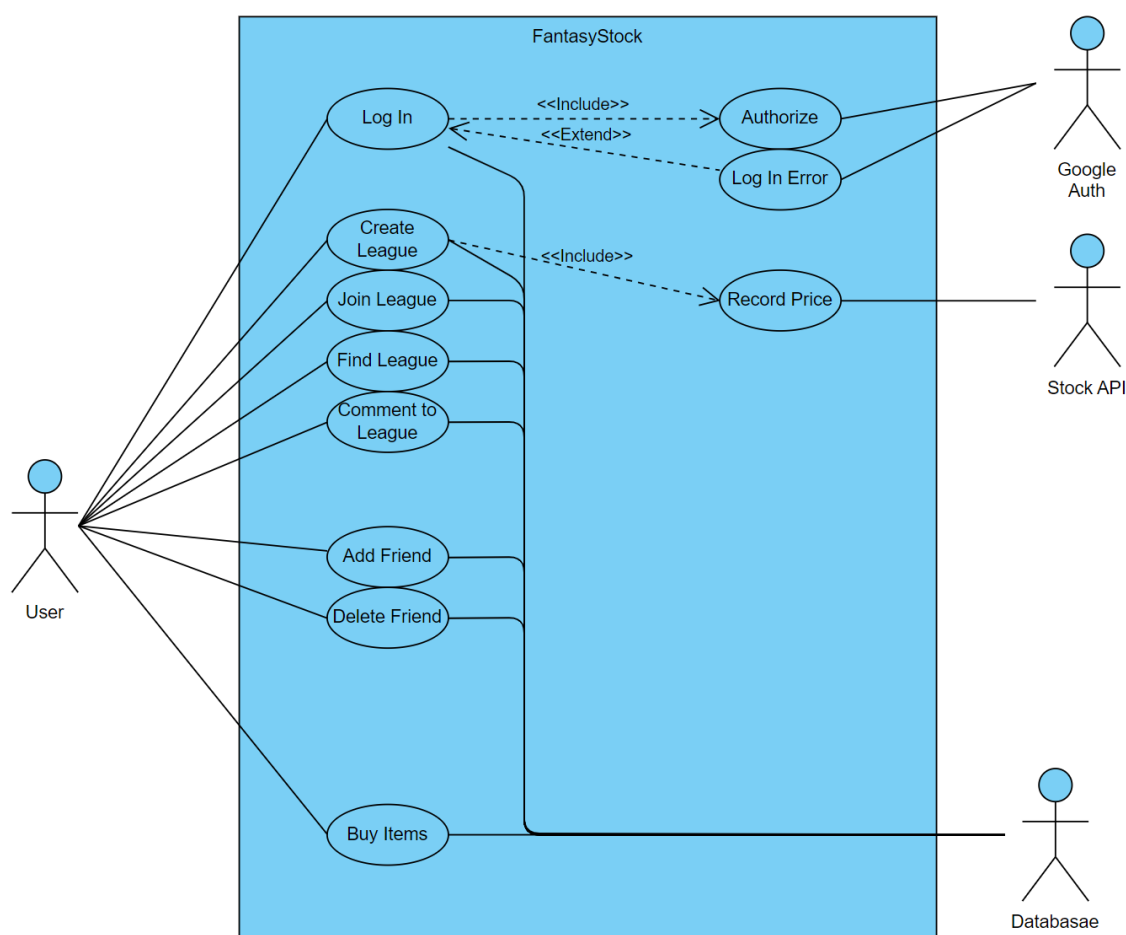
Other Requirements

Appendix A: Glossary

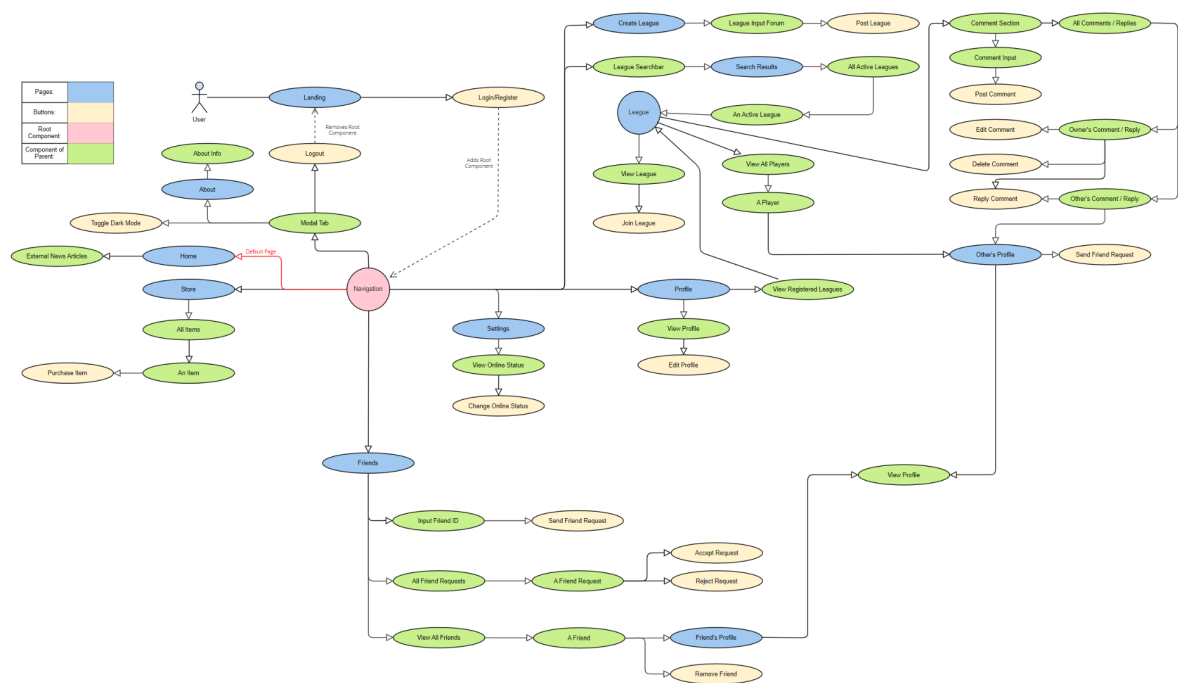
No glossary terms.

Appendix B: Analysis Models

Use Case Diagram



4.2 Site Map



Appendix C: To Be Determined List

TBD