

CS 3600 Project Phase 1

Jake Gendreau, Michael Crapse

February 25, 2026

Contents

1	Project Plan	2
2	Introduction	2
2.1	Understanding of the Project	2
2.2	Interpretation of Requirements	2
3	Technology Stack	2
3.1	Backend	2
3.1.1	Language	2
3.1.2	Libraries	3
3.2	Database	3
3.3	Visualization Tools	3
4	ER Diagram	3
5	Timeline of Deliverables	3

1 Project Plan

StockUp is a social and financial app which is a combination of fantasy football and real-world investment. On StockUp, users will be able to join a team, invest in stocks, and earn money for their team.

- Create an account with a starting balance of \$10,000.
- Join a team with other users.
- Invest those \$10,000 as they see fit.
- View current, historic, and predicted stock prices.
- Use the social tools to view their team members, user portfolios, and top performing investors.

2 Introduction

2.1 Understanding of the Project

We understand the project to be something which requires a database, frontend, and backend. Additionally, we understand the project to require a "wow-factor" element. Additionally, the project must include a graph database and a set of queries regarding the graph database.

2.2 Interpretation of Requirements

The project needs to implement, on top of our relational database, either an emulation or actual implementation of a graph database, with the following queries:

1. **Co-Author Network** Our equivalent of a Co-Author network will be the user's teammates. A user will be able to type in their own username, or anyone else's username, and see what team they are on, and who else is on that team.
2. **H-Index Filter** Our equivalent to an h-index filter will be sorting users based on their average portfolio performance. If they are making over a certain H-index amount, on average, per stock, then they will be listed. Essentially, this would be all of the highest performing users on the platform. It will list, under their name, what their top 5 stock values are.
3. **Q1 Journal Influence Network** Our equivalent to a Q1 Journal Influence Network would be to list all of the teams which feature one or more users who have an H-index score greater than a certain value.

3 Technology Stack

3.1 Backend

3.1.1 Language

For our backend language, we will be utilizing Python. We chose python for a few reasons:

1. **Documentation and Support:** Python is an extremely well documented and supported language, and as a result, many StackOverflow forms, Reddit threads, or the actual documentation will have everything that we need to solve any problems we may face.

2. **Libraries:** Extremely well supported libraries make implementing the features that we want to a much easier task than it would be if we had to implement them from scratch. Libraries such as Django or Flask for the web server, and PyTorch or XGBoost for machine learning provide extremely refined and well documented solutions for the features that we wish to add.
3. **API Interfacing:** Since our project will be utilizing the Yahoo Finance API, we wanted to use a language that has easy support for it. Python has the `yfinance` library, which automatically calls the Yahoo Finance API and formats the results.

3.1.2 Libraries

We will be using several libraries in our project. Primarily, we will be using the following:

1. **Django:** We chose to use Django as our web server because of its "batteries included" approach, which means that many features that a website may want to use (e.g. login security, cookies, databases) have first-party developed modules which are well documented and easy to add.
2. **XGBoost + PyTorch:** We chose to use XGBoost and PyTorch for the machine learning aspects of our project because they are extremely well documented, easy to use, and provide powerful tools for our application.
3. **YFinance:** We chose to use YFinance instead of doing manual API calls because in testing, I was frequently limited by the API due to utilizing it too much. As a result, I had to find an alternative which wouldn't limit me (at least as much). YFinance is a well-supported option which uses a reliable source for its information (Yahoo Finance).

3.2 Database

Due to our choice to use Python and Django, we were left with a few natively supported options for our database management software. After some conversation, we decided that SQLite3 would be the best option, as both of us have experience in the software, and the limited installation size and feature set means that it is easier to install, use, and develop with.

3.3 Visualization Tools

We chose to use Chartjs for our database visualization due to its compatibility with Django, nice looking output, and documentation support.

4 ER Diagram

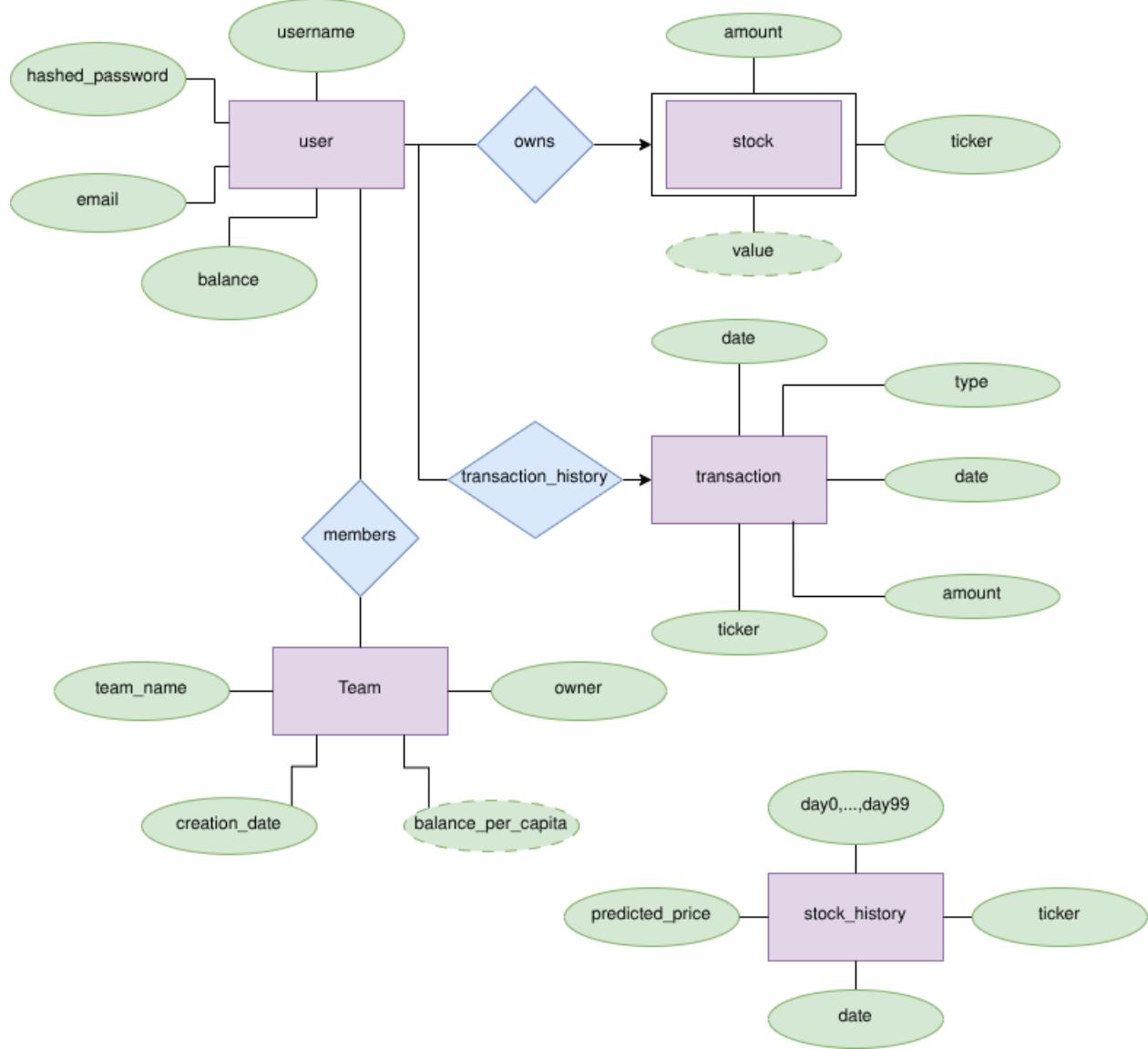


Figure 1: ER Model for StockUp

5 Timeline of Deliverables

- **February 26, 2026:** The initial draft for the project must be ready, along with a presentation about the project.
- **March 17, 2026:** The project must have a finalized database design, all interfaces designed, all functions planned out, and at least one functional interface with backend. Additionally, the stock price predictor model must be functional at this time. It should be done with Phase II at this point.
- **April 15, 2026:** All parts of the project must work in isolation of each other. The website

should be fully functional, the backend and API should be fully functional, and the database should be fully functional. At this point, the project should be done with Phase III.

- **May 1, 2026:** The project should be fully functional at this point. The required queries should be in place, and all aspects of the project should properly interact with minimal bugs. Phase IV should be complete at this point.
- **May 15, 2026:** The project must be complete by this point.