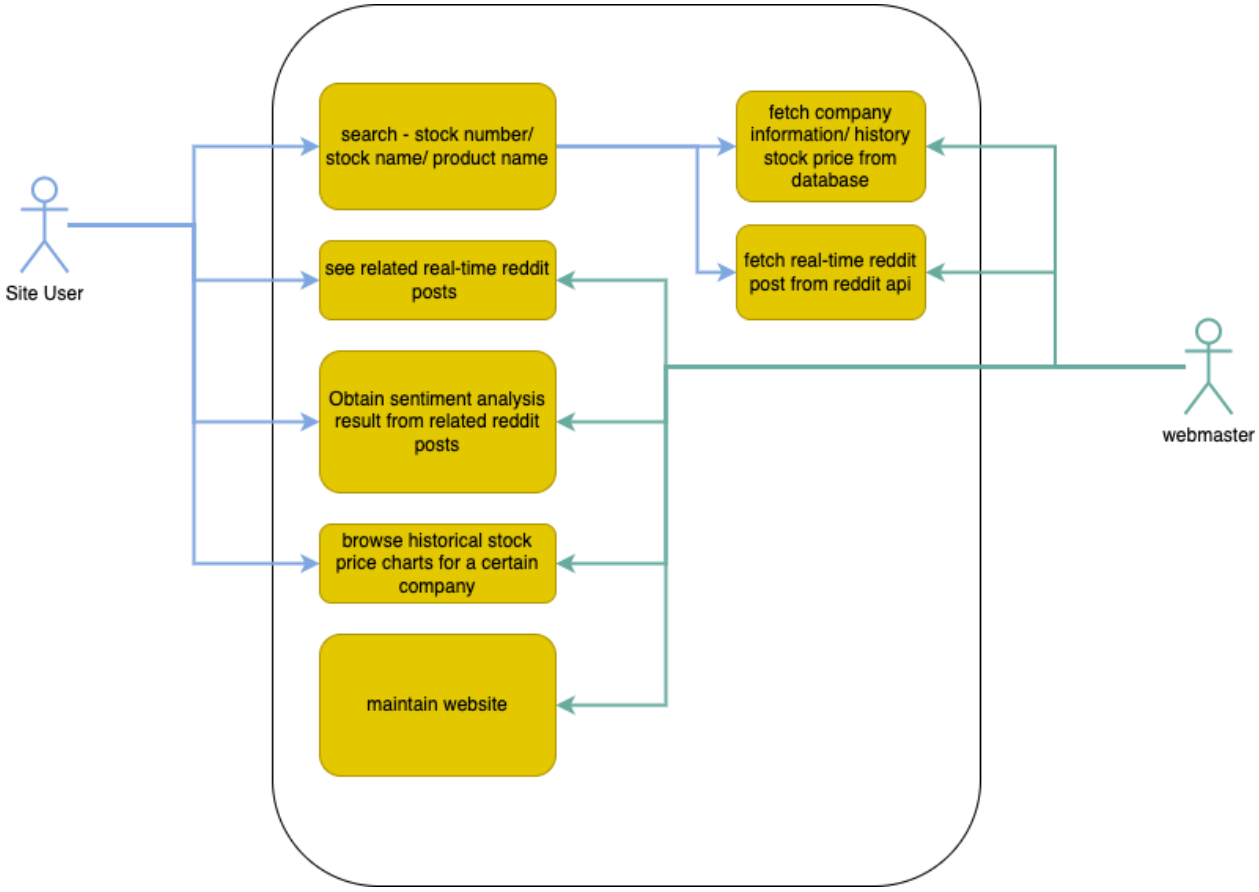
Sprint 1

I. Use Case Diagram



II. User Personas:

20 Years Old | Senior In University

Hi there, my name is John and I am a senior at Virginia Tech. I am excited to begin my investing journey, even though I won't be investing real money just yet. Instead, I plan to focus on gaining knowledge and experience by studying the stock market and tracking its movements.

Goals: Learn stock related knowledge

26 Years Old | Software Develop

My name is Joanna and I work as a software developer at a large tech company. I have been investing in the stock market since I received my first paycheck after graduating from college. My typical approach involves researching various stock platforms and browsing social media groups related to stocks to gauge market sentiment. Additionally, I stay up-to-date with news and financial reports relevant to the companies I am interested in investing in.

Goals : save time on browsing through stock related informations on social media

30 Years Old | Day Trader

My name is Diego and I work as a trader at a trading company, where I primarily invest in tech companies. With the increasing popularity of social media, its impact on the stock market is becoming more significant. As a result, I not only rely on traditional company information but also collect social media posts to gain insight into which topics are generating excitement or concern among people. This helps me make more informed investment decisions.

Goals: collect more information to formed investment decisions including historical stock market price and social media stock related posts

45 years old | Social science researcher

I'm Julia, a social science researcher at Stanford University. My area of interest is public psychology and I'm fascinated by the impact of technology, especially social media, on human behavior. I'm planning to conduct multiple research projects on this topic to gain a deeper understanding of how it affects individuals and communities.

Goals: Observe social media impact and find useful resources for research purposes

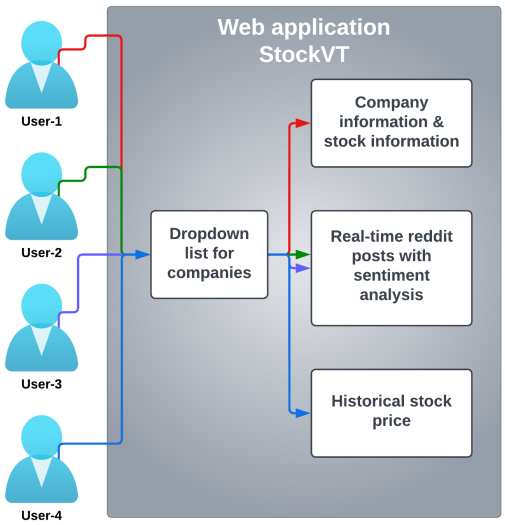
III. User Stories

A user story is a short, simple, and concise statement that captures the needs and requirements of a user or a group of users. It is typically used in agile software development to define a feature or functionality of a product from the perspective of the user. Below is the descriptions of the four user stories for the stock web application:

- User-1
  - As a novice interested in stocks, I want to search for stock-related information so that I can learn and improve my stock investment knowledge and skills.

- User-2
  - As a stock investor, I want to be able to see real-time Reddit posts for a certain company, so that I can stay updated on the latest stock market news and make more informed investment decisions.
- User-3
  - As a stock market researcher, I want to obtain sentiment analysis data from articles on Reddit for a certain company, so that I can discover the stock with the perspective of social media, and understand market trends and potential investment opportunities.
- User-4
  - As a stock trader, I want to be able to browse historical stock price charts for a certain company, so that I can analyze stock price trends and formulate trading strategies.

The diagram below illustrates how those users can use the system according to the user stories.



IV. Scrum

In this section, it includes all the tasks and tests for four user stories. This project has **Priority 1- 4** with their corresponding user stories. The user story with less features has a higher priority, and vice versa. In this way we can implement the minimum viable product (MVP) first and gradually add more features.

Priority-1

User Story:

As a novice interested in stocks, I search for stock-related information from time to time so I can learn more knowledge in stock investment.

Test	Expected result
Search functionality: valid stock symbols	Company information displayed w.r.t. selected stock
Search functionality: invalid stock symbols	No results found
Stock related information	The results are aligned with latest information

Tasks:

1. Design a user interface for displaying basic stock-related company information for a specific company. (24 hrs)
2. Create an automated script using the yfinance API to periodically fetch the latest stock-related data from Yahoo Finance and store it in MySQL database. (24 hrs)
3. Set up Maxwell to capture data changes in the MySQL database and stream them to Kafka. (48 hrs)
4. Develop a Reddit data producer for Kafka topics. Add Spark code to process the streamed stock-related data between Kafka topics. (96 hrs)
5. Write Spring Boot code to serve as the API endpoint for retrieving the processed stock-related data from the Kafka topics and sending the information to the webpage. (72 hrs)
6. Develop a section to display the history of stock price in a clear and organized manner. (72 hrs)

Priority-2

User Story:

As a stock investor, I want to be able to see real-time Reddit posts for a certain company, so that I can stay updated on the latest stock market news and make more informed investment decisions.

Test	Expected results
Reddit posts info: title, body, link, timestamp	Posts are the same as those shown on subreddits.
Real-time post feed	Posts are the same as those shown on subreddits
Feed's filtering option	Customized content displayed on specific keywords

Tasks:

1. Design a real-time Reddit post feed interface for displaying the latest posts from specific stock-related subreddits. (24 hrs)
2. Create an automated script using the Reddit API to periodically fetch the latest Reddit posts from specific subreddits and send it directly to Kafka. (48 hrs)
3. Add Spark code to process the streamed data from Kafka. (48 hrs)
4. Write Spring Boot code to serve as the API endpoint for retrieving the processed Reddit posts data from the Kafka topics and sending the information to the webpage. (72 hrs)
5. Develop the Reddit posts display section on the webpage. (72 hrs)
6. Implement a live update feature to continuously refresh the Reddit post feed with new posts. (72 hrs)
7. Add filtering options to the feed, allowing users to customize the displayed content based on specific keywords. (48 hrs)

Priority-3

User Story:

As a stock market researcher, I want to obtain sentiment analysis data from articles on Reddit for a certain company, so that I can understand market trends and potential investment opportunities, and provide them as decision-making references to traders and investors.

Test	Expected result
Sentiment analysis	Manually check if the classified results are semantically the same as the title and body on the reddit post.
Historical price data	Chart is displayed on the webpage.

Tasks:

1. Design an interface to display sentiment analysis results for Reddit articles related to the stock market. (24 hrs)
2. Use Kafka producer with Reddit API to periodically fetch the latest reddit posts from specific subreddits, and send data directly to designated Kafka topics. (48 hrs)
3. Create Spark application to access data from kafka topics, do sentiment analysis and store the result in Elasticsearch (48 hrs)
4. Write API to fetch data from Elasticsearch (48 hrs)
5. Develop the sentiment resort pie chart on webpage (24 hrs)

Priority-4
User story:

As a stock trader, I want to be able to browse historical stock price charts for a certain company, so that I can analyze stock price trends and formulate trading strategies.

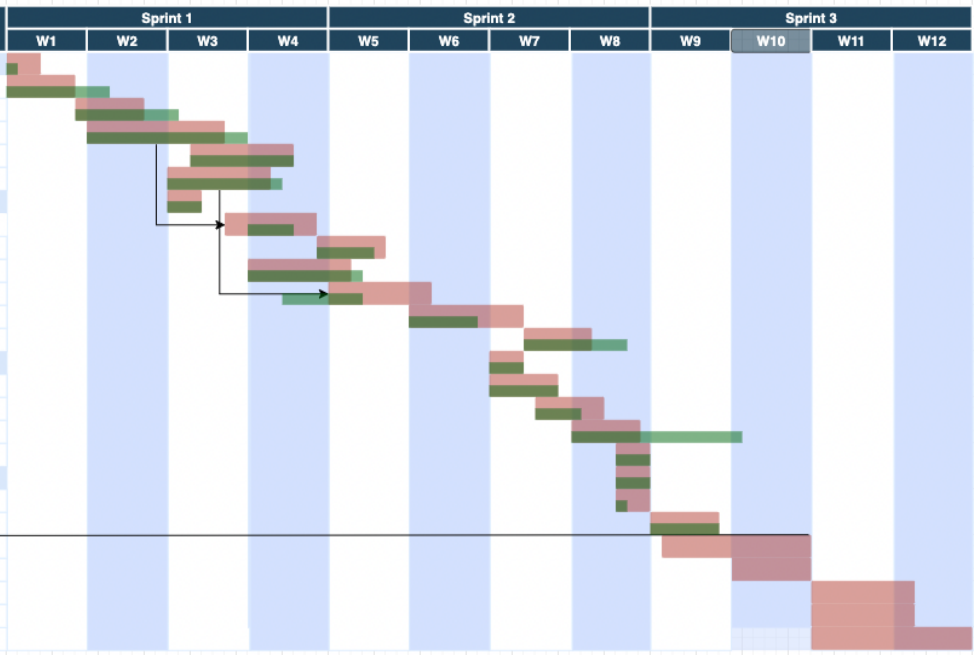
Test	Expected result
Search functionality	Price chart, history
Search functionality - error message	Error messages or "no results found" messages are displayed in place of the chart.
Historical stock price	accurately shows the price data w.r.t. the specified time range. Also make sure that it is regularly updated with the latest price data.

Tasks:

1. Design a user interface for displaying historical stock price charts for a specific company. (24 hrs)
2. Create an automated script using the yfinance API to periodically fetch the latest stock price data from Yahoo Finance and store it in MySQL database. (48 hrs)
3. Set up Maxwell to capture data changes in the MySQL database and stream them to Kafka. (48 hrs)
4. Implement a Spark application to process the streamed stock price data between Kafka topics. (96 hrs)
5. Create a Spring Boot application and Elasticsearch that serves as the API endpoint for retrieving the processed stock price data from the Kafka topics and sending the information to the webpage. (72 hrs)
6. Implement an API endpoint in the Spring Boot application to send the stock price data to the webpage. (72hrs)
7. Finalize all the charts (line, pie) to display on the webpage. (72 hrs)

Product Roadmap

	Task Name	Ideal	actual
1	Priority-1-1 (basic stock info)	24 hrs	10 hrs
2	Priority-1-2 (yfinance API, MySQL)	48 hrs	70 hrs
3	Priority-1-3 (Maxwell)	48 hrs	72 hrs
4	Priority-1-4 (Reddit PRAW, Kafka, Spark)	96 hrs	110 hrs
5	Priority-1-5 (SpringBoot: producer, service, mapper)	72 hrs	70 hrs
6	Priority-1-6 (front end: history stock price)	72 hrs	80 hrs
7	Priority-2-1 (front end: search bar, filter)	24 hrs	24 hrs
8	Priority-2-2 (reddit post Kafka producer, early split topics)	48 hrs	30 hrs
9	Priority-2-3 (Spark Dstream with Scala)	48 hrs	45 hrs
10	Priority-2-4 (SpringBoot: Restful API)	72 hrs	90 hrs
11	Priority-2-5 (front end: reddit post display section)	72 hrs	60 hrs
12	Priority-2-6 (live update of reddit posts)	72 hrs	50 hrs
13	Priority-2-7 (keyword filter)	48 hrs	70 hrs
14	Priority-3-1 (sentiment analysis result for each posts)	24 hrs	24 hrs
15	Priority-3-2 (Kafka producer send to aggregated topics)	48 hrs	48 hrs
16	Priority-3-3 (Data cleaning, sentiment analysis)	48 hrs	40 hrs
17	Priority-3-4 (Elasticsearch APIs)	48 hrs	120 hrs
18	Priority-3-5 (front end: Pie chart for analyzed results)	24 hrs	24 hrs
19	Priority-4-1 (front end: historical data)	24 hrs	24 hrs
20	Priority-4-2 (yfinance polling to MySQL)	24 hrs	10 hrs
21	Priority-4-3 (Read/Write splitting with Maxwell)	48 hrs	48 hrs
22	Priority-4-4 (Data cleaning, sentiment analysis)	96 hrs	
23	Priority-4-5 (Finalize the SpringBoot - Elasticsearch)	72 hrs	
24	Priority-4-6 (Finalize the API endpoints)	72 hrs	
25	Priority-4-7 (Finalize all the charts: line, pie)	72 hrs	
26	Buffer	112 hrs	



Product Burndown Chart