Our team consists of William Skedd (wskedd2), Sewoong Lee (called Sam) (samuel27), Ritik Kulkarni (rk30), and Jeremy Bao (jbao8). William is the captain.

Our free topic is to create a search engine for StockTwits. StockTwits is a social media platform in which investors can post about their opinions regarding the stocks of various companies. It is fairly similar to Twitter, but also displays the price of and its users' feeling about various stocks over time (though this is based off of its users explicitly stating if they are "bullish" or "bearish" about the stock they are posting about and not on any sophisticated text mining algorithms). Unlike Twitter, StockTwits offers no searching functionality, so retrieving old posts is very difficult and inconvenient.

We plan to remedy this problem. We will create a web crawler to gather information about the posts within StockTwits, create an inverted index to store information about these posts, design a search algorithm capable of taking a query and retrieving relevant posts, and implement a web application capable of performing searches on the indexed posts using our algorithm. To tune the parameters of the text retrieval algorithm (perhaps Okapi BM25) that will be used for our search engine, we will create a test collection of posts from StockTwits, example queries, and relevance judgements, then try out different algorithms and combinations of parameters and see which yields the best F-score or mAP.

To do this, we will get our documents (both the entire corpus and the sample set for our test collection) from Stocktwits.com, our sample queries by making them up, and our relevance judgements by looking through the sample set of documents and deciding which are relevant to each sample query. The BeautifulSoup and Selenium libraries will be used for our web crawler. Metapy will be used to implement our text retrieval algorithm. Google Firebase will be used for storage, hosting and cloud functionality. Flask will be used to create our web application.

Designing a search engine for StockTwits posts would be very useful, as it would help investors figure out what others think about particular stocks. This could help them make better decisions about when stocks should be bought and sold and thus earn more money.

The expected outcome is us creating a web application that can, when given a query, return posts from Stocktwits.com that are relevant to that query. We will evaluate our work by evaluating its F-Score or mAP on the previously mentioned test collection.

We plan to use Python for our web crawler and our interactions with Google Firebase. Python will also be used for implementing our text retrieval algorithm. A mixture of Python and JavaScript will be used to create our web application. HTML will be used to design our application's user interface, and CSS will be used to specify its appearance.

Our project will probably take at least 80 hours to finish. Creating the code needed to crawl posts from Stocktwits and store them will take around 20 hours. Creating, tuning the parameters for, and evaluating our text retrieval algorithm will probably take 20 more hours. Developing, styling, testing, implementing cloud functionality for, and performing database management for our web app will take no less than 40 hours. Our project involves enough work to occupy each of the four members of our team for at least twenty hours.