HotOrNot: Tracking company attractiveness and profitability from financial news streams

Euan O'Neill

Motivation

- Create a tool to aide financial traders

- Use sentiment of articles to calculate an attractiveness score of a company

Background

Academic Review

- Machine learning
  - Stock Prediction
- Named Entity Recognition
- Sentiment Analysis
- Sentence Extraction

Competitors

- Investing.com
- Trading View
- Stock Analysis

Requirements

Requirements Gathering

- User Studies
- Backend & Frontend
- MoSCoW
  - Must
  - Should
  - Could
  - Won’t Have

Backend Requirements

Must

- Article Collector
- Data storage
- Context Identification
- Sentence Extraction
- Sentiment Analysis
- Stock Prediction

Should

- Accurate context analysis
- Accurate Sentiment analysis
- Accurate Prediction
- Variety of Companies

Could

- Prediction Options
- Deployed and accessible

Frontend Requirements

Must

- Home Page
- Company Page
- Search Bar
- Filters and Tags
- Prediction Results
- Stock Graph

Should

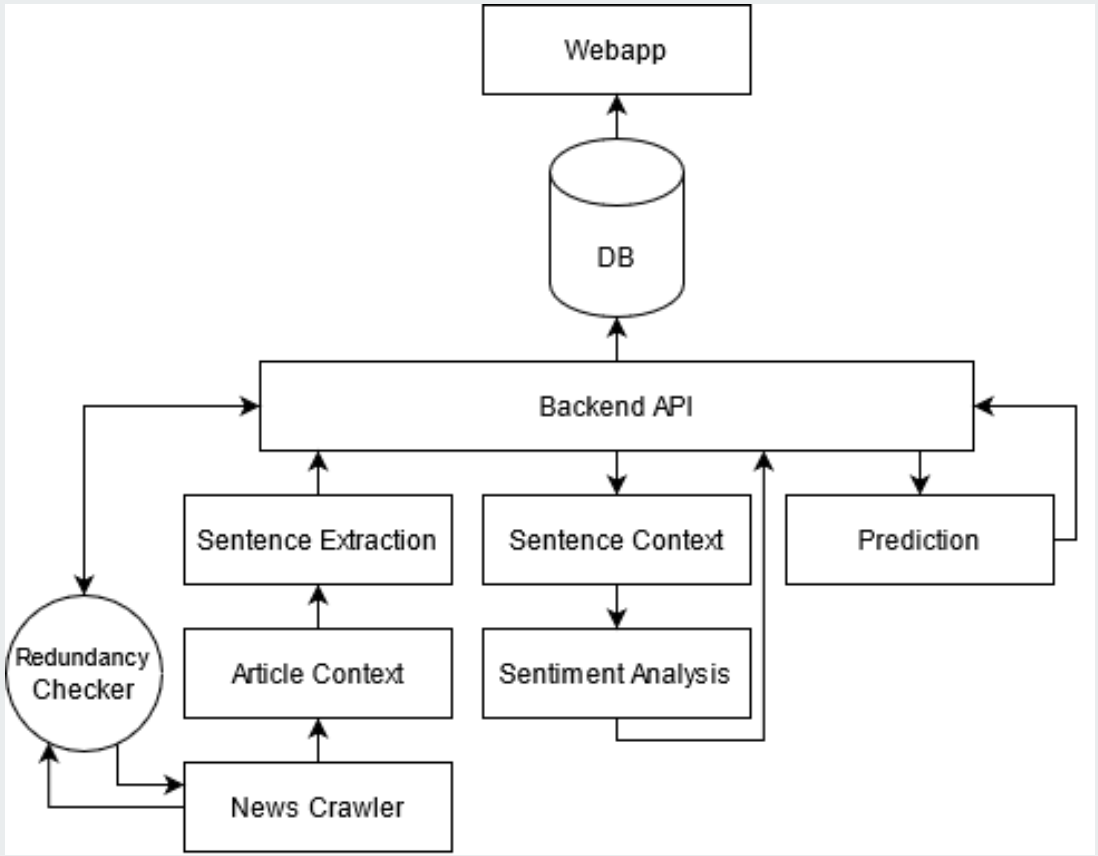
- Related Companies
- Website Design

Could

- Additional Company information
- Personalization
- Article Snippets

Design

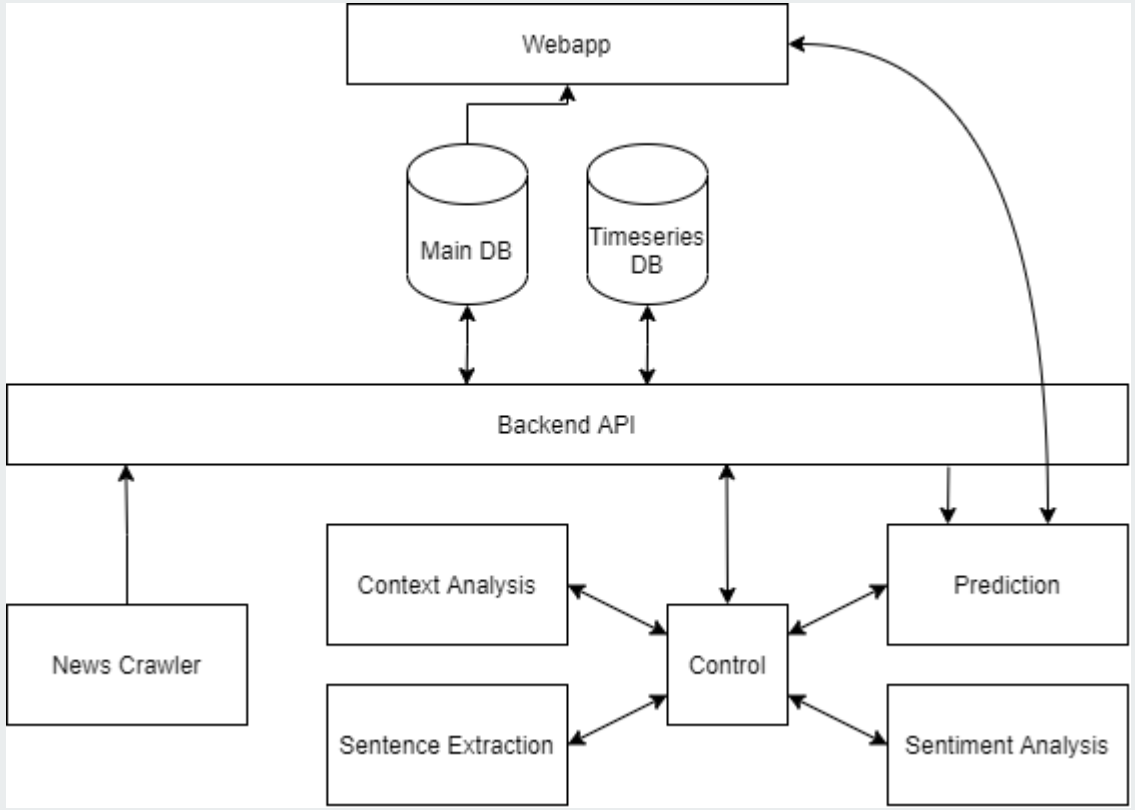
System Architecture Original



System Architecture Revised

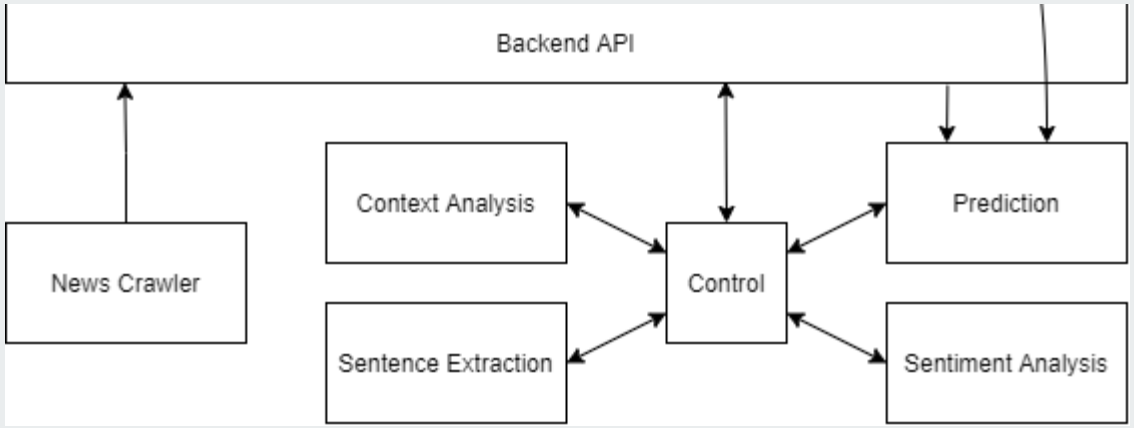
Components

- Backend API
- Context Analysis
- Control
- Main DB
- News Crawler
- Prediction
- Sentence Extraction
- Sentiment Analysis
- Time series DB
- Webapp



Backend

- News Cralwer
- Sentence Extraction
- Context Analysis
- Sentiment Analysis
- Prediction
- Backend API
  - Swagger Documentation
- Control Component
  - Management logic



New Crawler

- Collects Articles from seven well established news sources
  - Wall Street Journal
  - TIME
  - BBC
  - Economic Times
  - CNBC
  - Financial Times
  - Fortune
- Inputs Articles directly into the database

Sentence Extraction

- Extracts sentences from Articles Transcript

Input: Transcript of Article

Output: List of sentences

Context Analysis

- Analyses text of Articles and sentences to understand what companies are being discussed

Input: Transcript of Article

Output: List of potential Companies

Input: Text of Sentence

Output: Stock code of company

Sentiment Analysis

- Awards sentiment score ranging from 1 to -1

Input: Text of Sentence

Output: Sentiment polarity score

Stock Price Prediction

- Predicts future price stock price for a company

Input: Stock code of company

Output: Price predictions and verdict

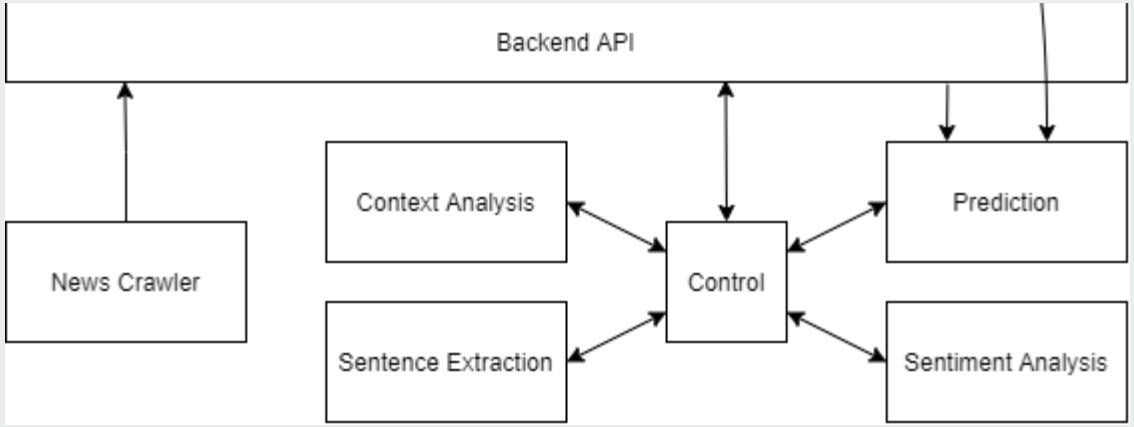
Input: Stock code of company and time frame

Output: Price predictions for given time frame

Control Component & Backend API

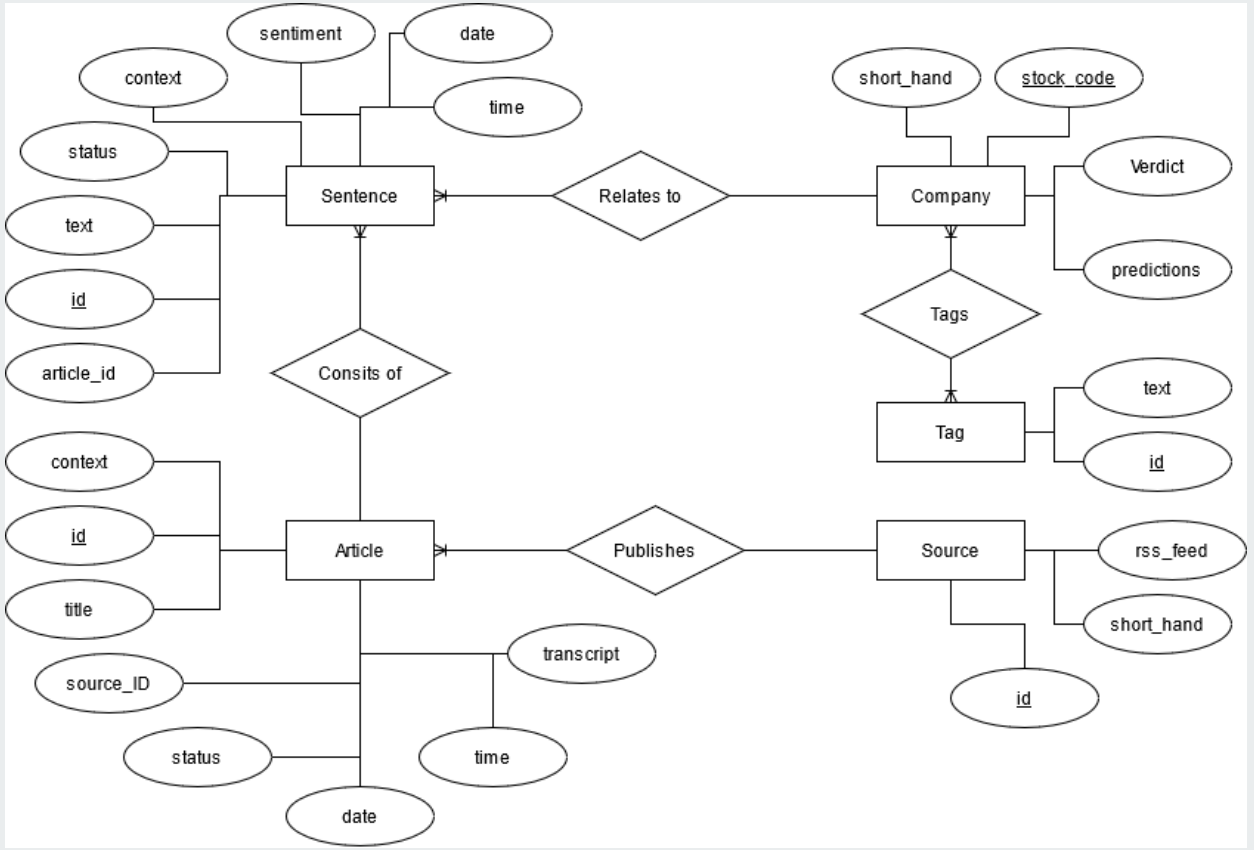
- Backend API acts as gateway to database

- Control component communicates with components



Database

- Main Database
  - Source
  - Article
  - Sentence
  - Company
  - Tag
- Time series Database
  - Prediction Datapoints

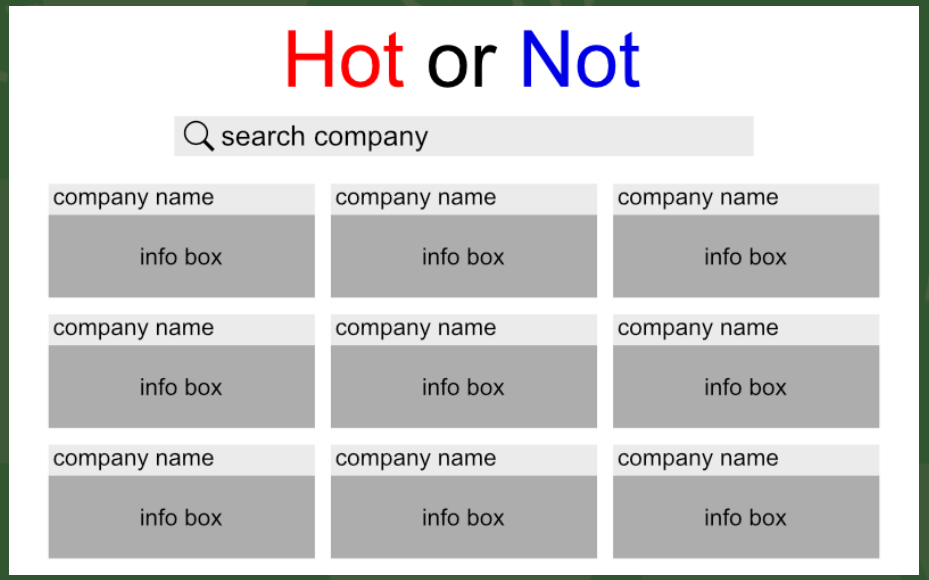


Webapp

- Home page
- Company page
- Iterative Wireframe designs

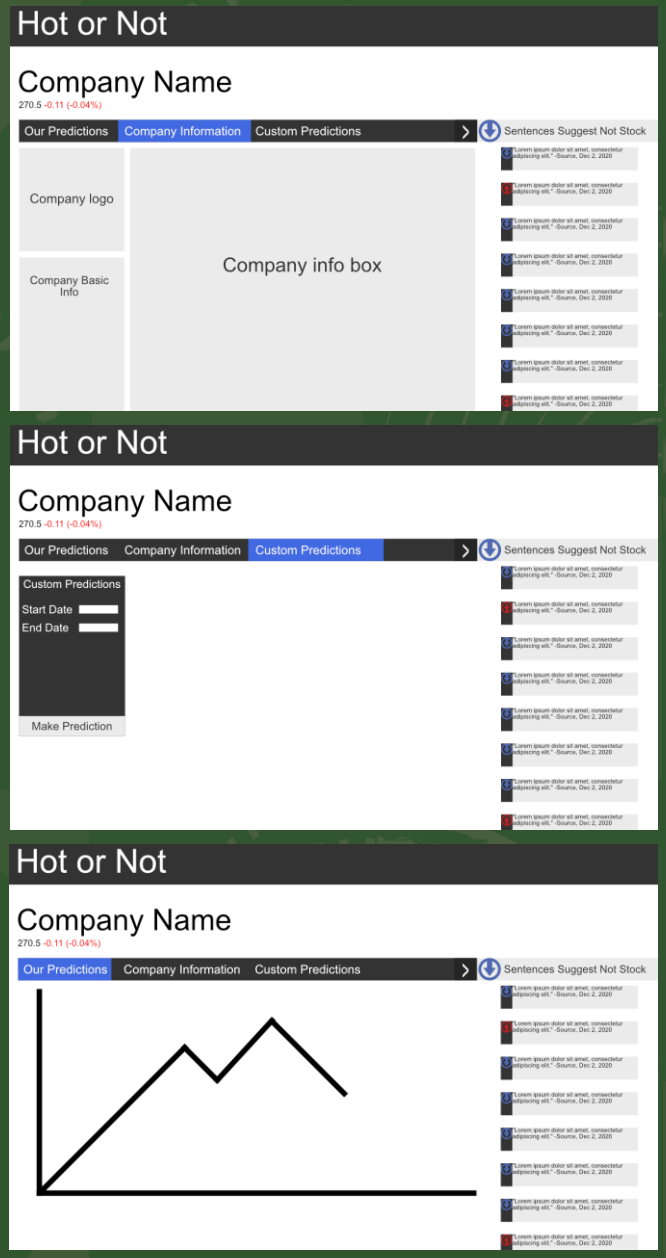
Frontend: Home Page

- Displays list of companies to select from
- Hot, Not or Hold verdict



Frontend: Company Page

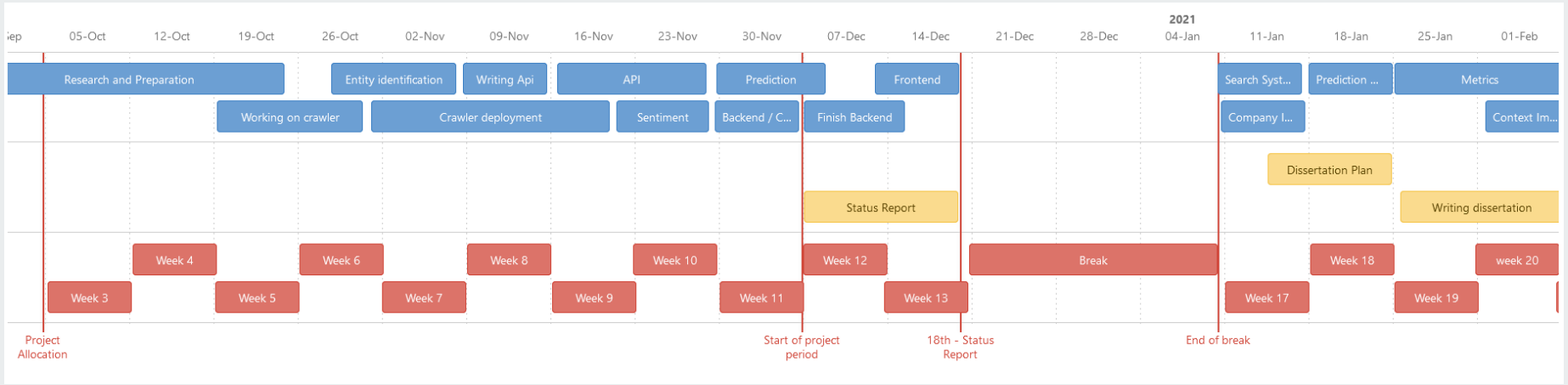
- Our predictions tab
- Company information tab
- Custom predictions tab
- Sentences relating to company



Implementation

Software Engineering

- Github
- Confluence



Backend

- Flask
  - SQLAlchemy
- NLTK
- Spacy
- Vader Sentiment
- Scikit-learn
- Yfinance

Database

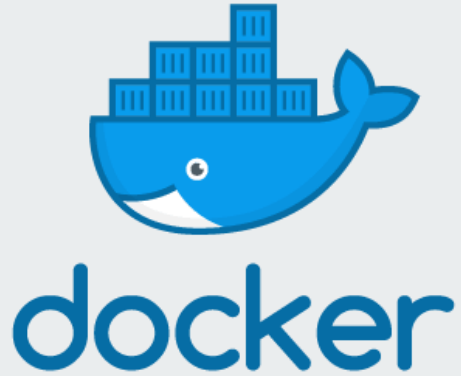
- Database implanted using PostgreSQL
- pgAdmin allowed for easy implementation
- Time Scale for Timeseries database

Frontend

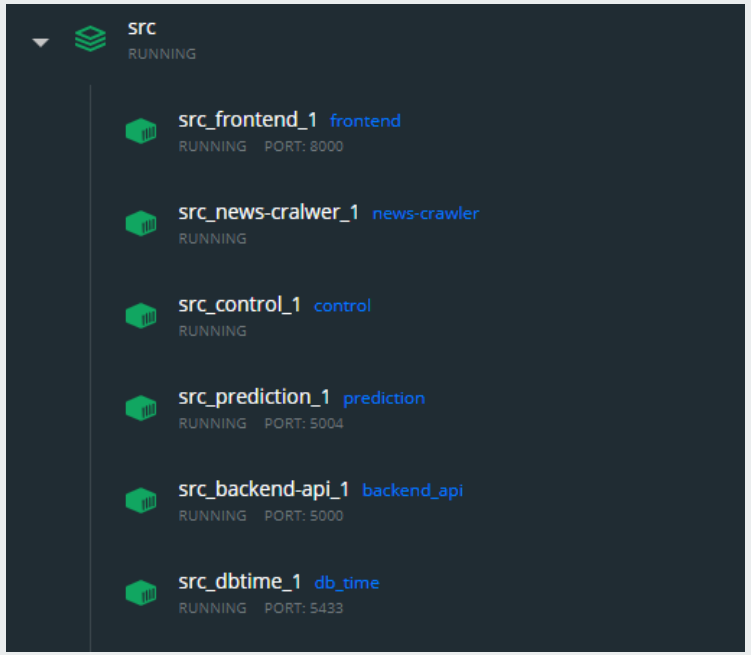
- Implemented using Django Framework

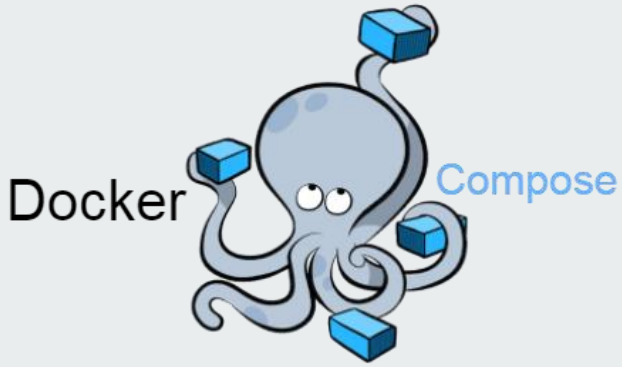
- Time taken to keep consistent and simple design

Containerization



- Docker used to turn component into containers
- Docker-compose used to turn containers into single application





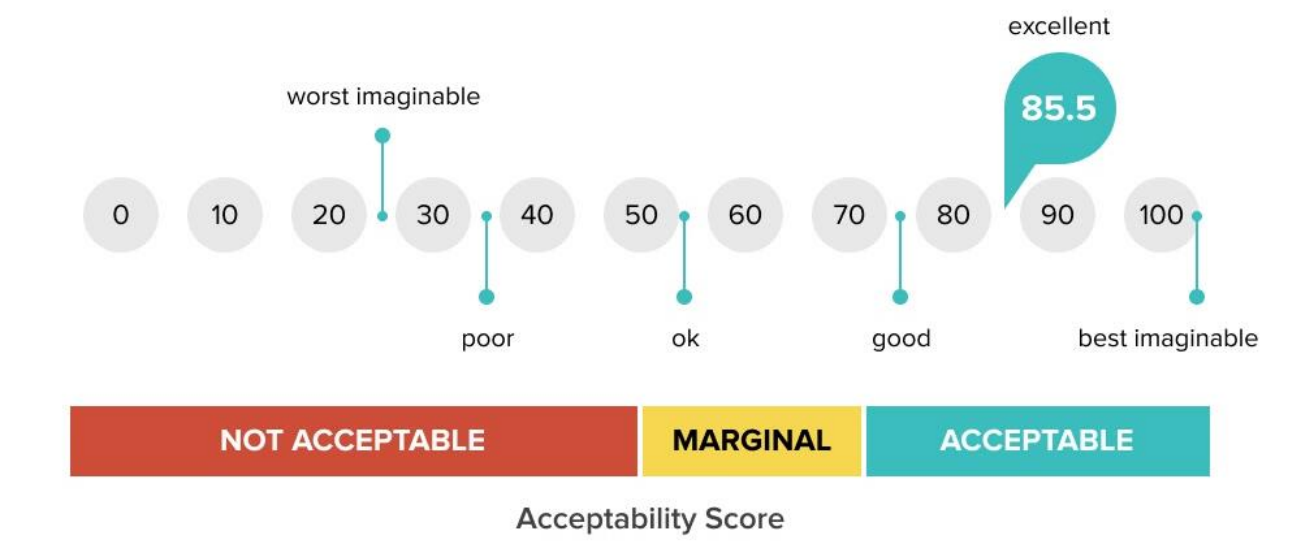
Evaluation

Backend

- Context Analysis
  - System switched from Spacy to Stanza NER
  - Minor overall prediction improvement
- Sentiment Analysis
  - System switched from Vader to Text Blob
  - Minor overall prediction improvement
- Stock Prediction
  - System switched from SVM model to Decision Tree Model
  - Major overall prediction improvement

Frontend User study

- System Usability Scale used as questionnaire
- Six participants took the test
- Average score: 88.75



Demonstration

Conclusion

Thanks for watching