

Requirements Doc MS-CS-Insights

Date: 2024-10-09

Sponsor Signature:

A handwritten signature in black ink, consisting of a large, stylized 'N' followed by a series of loops and a long horizontal stroke.

Time-frame for this requirements doc

- First semester
- A new requirements doc will be created for the second semester

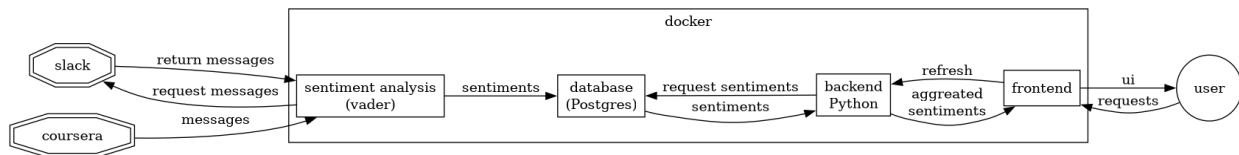
User story - how would a user interact with the product

- User Story 1
 - A course coordinator will add the app to a workspace they wish to have sentiment analysis for.
 - This will allow the app to query and listen for messages and send them to the server for processing.
- User Story 2
 - A user will access the web interface and see a indicator of the positive/negative 'sentiment' for each course arranged in a grid
 - The user will click a "refresh" button and the page will be updated with the most recent sentiments from all the respective messages in each course.
- User Story 3
 - A user will access the web interface, click a button(or buttons) and can see a "word cloud" for each class(or all)

- This word cloud will not include irrelevant words such as “the”, “and”, “is”, etc.
- This will allow the user to determine which topics are being discussed frequently.

How the product should work (MVP)

- The user views the home page and should see relevant courses and their sentiment analysis data from the last sentiment analysis refresh.
- The frontend service should request this data from the backend API.
- The backend API will then query the sentiment analysis data from the database and serve it to the frontend
- The sentiment analysis service will queue messages from slack/other sources
- Upon user clicking a 'refresh sentiment' button, the sentiment analysis service will analyze queued messages and add results to the database.



These kinds of people should be able to use the product

- Course coordinators (Non-tech people)
 - The product should be easy and intuitive to use
 - The product should show the status of each course to inform decisions