Xomnia Data Engineer Assessment

This document contains the Xomnia Data Engineer Assessment. The results of this assessment will be used to determine your technical skill level, your understanding of the underlying data and your problem solving abilities. In the context of a project for Shipping Technology™, our product owner wants to have insights on GPS information and weather conditions during the travel of our cargo vessels. In this assessment you are given a dataset containing log data that our fleet transmits during its travels and you are asked to deliver a small application that satisfies our product owner's needs

The estimated time is between four and eight hours. We are interested in a fair assessment of your thought process and skills so do not over-polish your results. A shortcut is allowed and sometimes necessary as long as you can explain what a better (more time consuming) approach would be. Because of the nature of the assessment, this is not required to be a completely realistic and "production-ready" app even though some steps into that direction will be appreciated. You can use as many conventions/assumptions for the logic of your application, as long as those are clearly stated. If a question is unclear to you, please proceed with your own interpretation.

Dataset

Assume this is the initial data load of the application. The log data look like this:

Field	***Definition***
Device_id	Ship identifier
datetime	Unix timestamp of the message transmission Hint: Should be used for all time based operations!
address_ip	Ip address of the system on board transmitting the data
address_port	Port of data transmission
original_message_id	Message identifier
raw_message	Raw message with all transmitted information

The raw_message field contains important information as comma separated values. Only alphanumeric characters (and dot for decimals) are valid characters for the contained information, however in some of the rows there is noise due to a malfunction of the transmission device. Example fields contained within a ***raw_message*** record:

Data status:	V=navigation receiver warning, A=ok
Latitude:	51.31831
Latitude Direction:	N or S (North/South)
Longitude:	4.18015
Longitude Direction:	E or W (East/West)
Speed over ground in knots:	0.0
Track made good in degrees True:	1.59
UT date:	150218
Magnetic variation degrees:	0.8
Magnetic variation direction:	E or W

You will need no other GPS format specific information aside those mentioned above. You may not need all of the fields within the raw_message mentioned above. Also, there is no need to use any geo-specific libraries for the needs of this application.

For some of the above questions you will need additional weather data that you can find in the provided weather_data.json file.

ASSESSMENT

1. Initially you will need to bring the dataset to a consistent human readable format according to the above specification, without duplications and other "noise" in the data.

Helper: In case you spend too much time without coming to a solution, the raw_messages_clean.csv is provided with the cleaned and parsed data.

2. Our product owner explained that he would like to look at some metrics he thought necessary for effective monitoring of our fleet's travels. Eventually, your application should have a way to expose the data related to the requested metrics via appropriately configured web endpoints.

Hint: It is up to you if the metrics will be served from a single or multiple endpoints! The API can be only available locally for demonstration purposes, many bonus points if the application is actually deployed somewhere!

Metrics:

a. For how many ships do we have available data?
b. Avg speed for all available ships for each hour of the date 2019-02-13.
c. Max & min wind speed for every available day for ship "st-1a2090" only.
d. A way to visualize full weather conditions (example fields: general description, temperature, wind speed) across route of the ship "st-1a2090" for date 2019-02-13. In case of time pressure, the application could simply return the requested data instead of a visualization.

Hint: The "visualization" could be as simple or complex you want, from a simple table to a using a map layer to visualize the route of the ship, is up to you :)

Tip: The weather_data.json file contains weather information for selected locations (lat, lon) of the ship "st-1a2090" for the date 2019-02-13!

3. Our colleagues from BI would like to collect history of the data during our fleet's travels combined with the weather conditions so that they can perform analysis on the impact of weather to our operations, therefore the application should persist the data in any kind of database system of your choice.
4. (BONUS) Finally, the code should follow clean coding best practices. The deliverable should come bundled with all necessary dependencies so that nothing besides a computer and an internet connection are needed to run the application and reproduce the results.

Good Luck!