Specification draft

Input

- Data requirements:
  - should be a valid signal set in IVIS.
  - signal set should countain the following to be valid for the anomaly detection framework fo choice (Time-series Anomaly Detection Framework)
    - > should the actual framework even be mentioned or do we want to make the selection of the framework a part of the project itself
    - timestamp signal in `Date/Time` data type and with a constant frequency
    - one other signal representing the data with values valid in its data type
    - (further requirements for data might come up due to the anomaly detection framework)

Visualization

To add support for anomaly detection to IVIS framework there will be two loosely connected components:

- anomaly detection component and
- anomaly report component.

The connection will be relized by a job/task.

Anomaly detection component

User oriented component used to help in setting up the anomaly detecting framework - helps with the input connected to seasonality, boundry selection and final job creation.

It should consist of at least two main parts - first for user input and second for vizualization on a subset of data from a selected valid signal set.

In the user input section there should be a drop down menu to select the seasonality or use auto-detect. There should be a field to input boundries for what is/is not deemed an anomaly or an option to go through iterative boundry selection.

Visualisation section should consist of a graph showing a subset of selected data and its state with respect to what has been selected.
In the graph there should be lines representing the boundries.
The anomalies and non-anomalies should be visually different.

> Maybe a colour, similarly to the actual app/detection framework

> What might be missing is exact description of the iterative mode.

There should be a button to create a task/job after all the settings are deemed final.

Anomaly detection job

The job will be created based on the input made in the anomaly detection component to calculate the anomalies on the entire signal set as appose to just a subset used for visualization in the previous step.

Once a run of the job is completed - new signals are created in the signal set where we are computing the anomalies.

- boolean signal representing `isAnomaly` indicator.
- two signals representing the 25th and 75th percentille related to the calculation of the anomalies serving as a boundry

This effectively takes the pair (time, value) and turns it into a tuple of 5 values (time, value, isAnomaly, lowerBound, upperBound).

Anomaly report component

This component should be used mostly by the end users of whatever the user is running.

It will consist of multiple parts:

- scatterplot with the detected anomalies
  - in the scatterplot there will be a visible are of where the non-anomalies were present
- bar chart with the count of anomalies per (selected) interval of the timestamps

The scatterplot will show point from the original signal based on the `isAnomaly` signal.
The scatterplot will also show the boundaries of where the non-anomalies lie.

The bar chart will show the number of anomalies in an interval selected in a filter.

User stories

> I am not sure but I think the visualization and user stories are blended together a bit too much so I would appreciate some feedback

- As a user I can create a template using the anomaly detection component, use it in a panel and where it provides visible I/O on a subset of the selected signalset to detect anomalies in it

- As a user I can input seasonality or select “autodetect seasonality” so it can be used to determine anomalies.
- As a user I can select min and max boundry for anomalies or select to use the “iterative autodetect” feature.

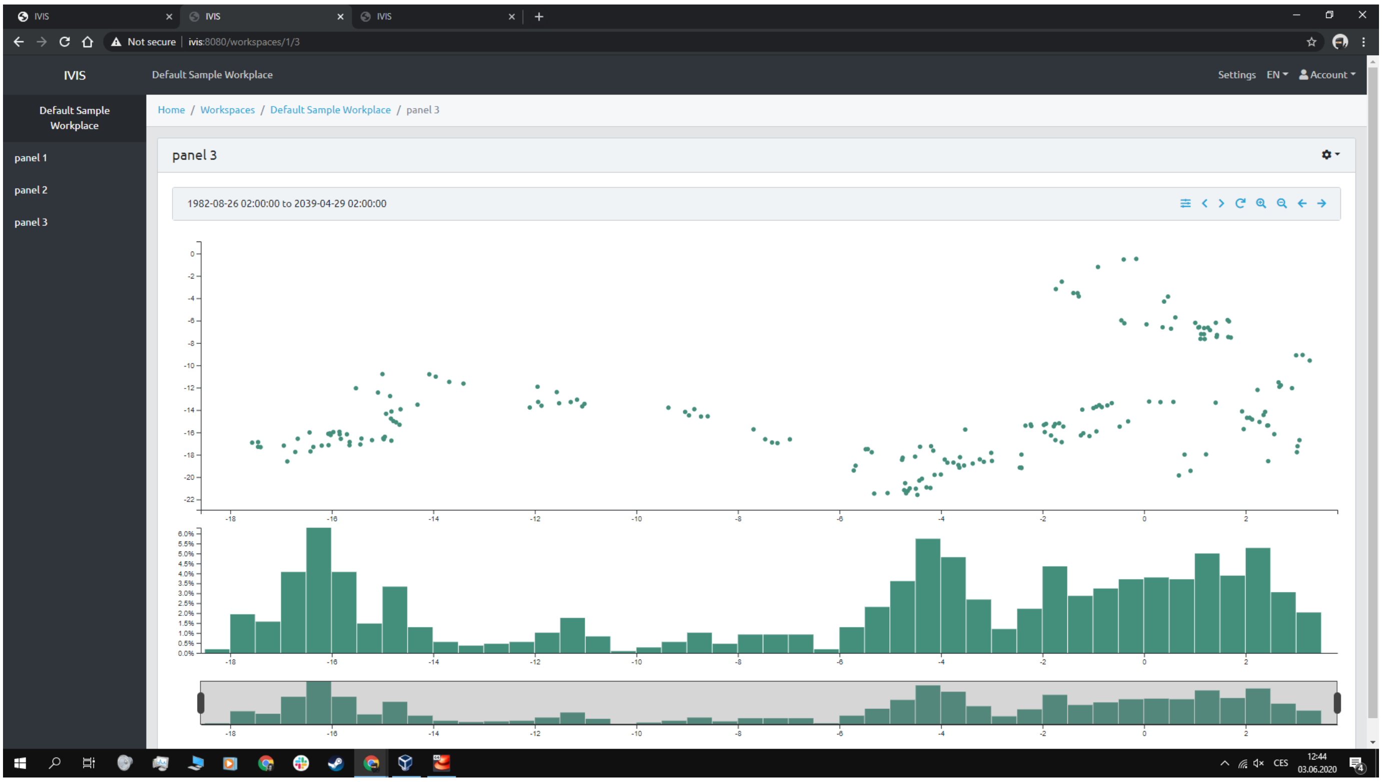
- As a user I can create a task and a related job that runs in the background and according to what I selected in the anomaly detection component it calculates anomalies on the rest of the data set.
- As a user I can create anomaly report component (chart) where
  - the anomalies of a data set are visible alongside their count per selected time interval. (count ~ separate linechart; anomalies ~ dots in the chart, avg ~ line and the limits as an area around it)

Mock-ups

The following images represent the potential designs of the user interface that accomodates the features and functionalities discussed above. It is based on the existing UI features of the IVIS framework.

TODO - add imiges

> after the inital input
> this is the base that will be used in the mockups



1982-08-26 02:00:00 to 2039-04-29 02:00:00