Milestone 2 Report

Engicoders

H. Bird, B. Karacelik, J. Peters, J. Ropotar, A. Rybka

Department of Computer Science, University of British Columbia

COSC310: Software Engineering

Dr. Gema Rodriguez-Perez

February 16, 2024

Project

The project is an IoT system to query, cleanse, filter, visualize, alert, and analyze manufacturing data. The project will function to empower engineers with a web application for the visualization and encapsulation of sensor data from IoT devices. Additionally, the system will use machine learning techniques to draw insight into the future states of sensors.

Project Description

The project is meant to take IoT data based on user preferences from an API. API data, of any variety, will be pulled from the IoT cloud platform "Thinkspeak" using its GET HTTP API. The data provided from the API can come in many forms such as pressure, thermometer, speedometer, height, and many more, as long as they are against time. The user demographic is manufacturing engineers seeking to utilize IoT data from sensors on a plant floor or that provide manufacturing data. This data is then cleansed, the cleansing process uses a combination of filters and outlier removal. Following this sk-learn ML algorithms, based on user desires, are used to predict potential future values and trends in the data. Also, high and low limit alarms can be added to the sensor to inform the user when the inputs go over, or under their desired value. Cleansed and analyzed data is then displayed in standard formats to provide insight into manufacturing operations. These visualizations may be exported and shared to empower manufacturing engineers. Finally, an alerting system will be implemented to allow the specification and identification of discrete sensor values that are dangerous and need to be acted on.

User Requirements

- Common users will start by signing into their account
  - If the user doesn't have an account, they will be prompted to make one
  - Users are also offered to upgrade their account for a price
- Users can select a dataset from an API
- Users can filter the dataset they chose as they see necessary
- Users will select the sensor that they want to visualize from the API
- Users select the type of visualization they want for the sensor
- Users will be able to create alarms based on historical data
- Users set the axis parameters for the data they chose
  - The user can set high and low limits to be alerted if they are breached
  - The user can set alarm management options to avoid alarm issues
  - The user can select an advanced Machine Learning Algorithm to predict future values of a sensor using other sensor values which is accompanied by a confidence interval for all predicted data
- Results will be displayed for the user in Visualization

        - The user can export the visualization
    - The user will be able to see the alarm log if any went off
        - The user can export the alarm log
    - Users can Sign out at any time, which kills their current data
    - Premium users will have all the functionality of the common user but be able to upload their dataset

Requirements

Functional	Non-Functional
The program will provide the user with data analysis and the data set selected	The transition between sending a pull request of data to the API and accessing the data will be seamless
The program will predict future trends within the data the user provides	The webpage should run on any modern device, including mobile.
The program will visualize data pre- and post-analysis for the user	The prediction algorithm should not take more than 1 minute to process future data
	User input data must be securely stored, as it could include confidential information
	The webpage must be able to handle multiple user interactions at the same time
	The UI will be intuitive and easy for the user to interact with

Table 1. Functional and Non-Functional Requirements

Use-Case Diagram

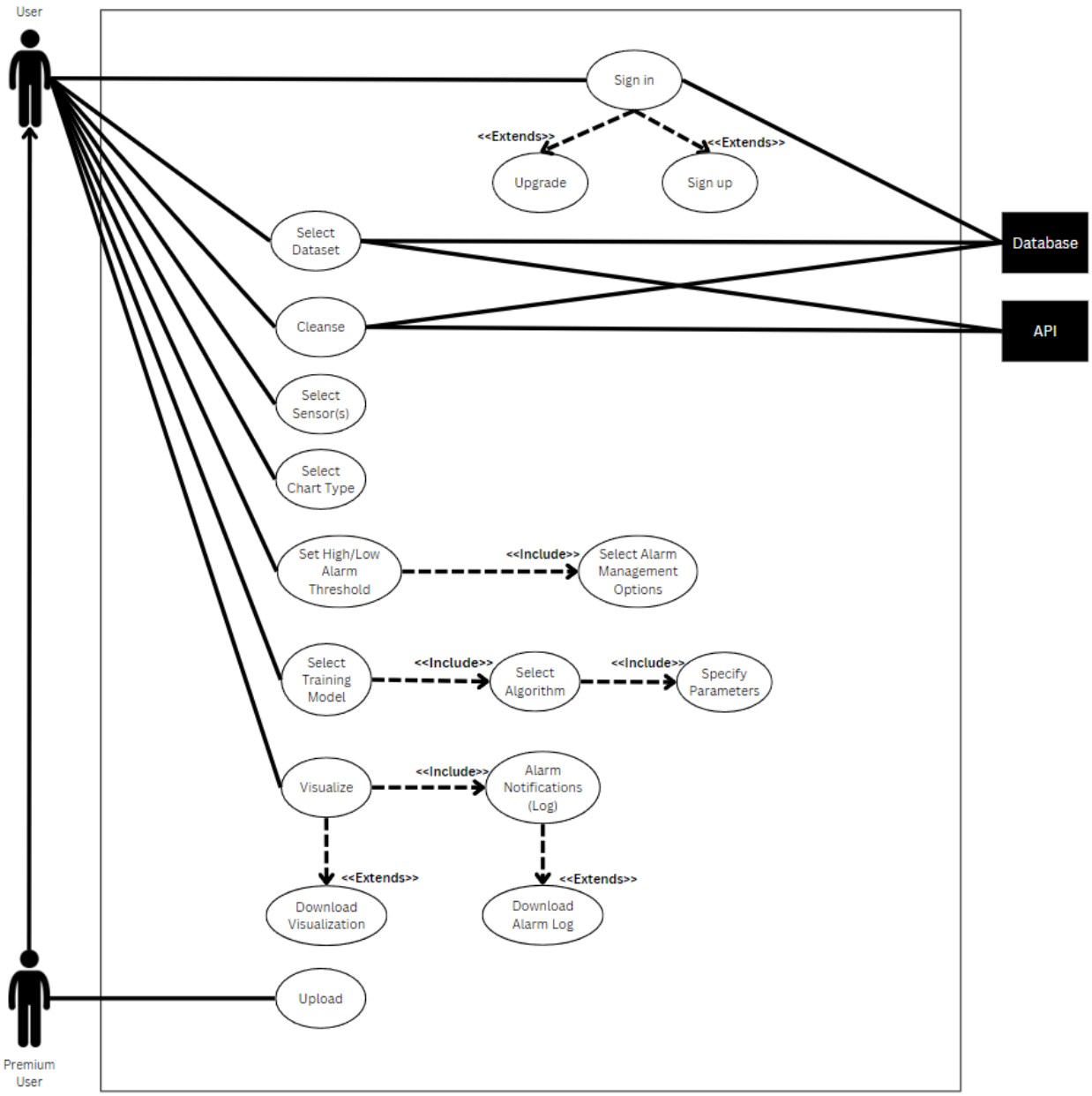

Figure 1. Visualize Data Use-Case Diagram

Use Case: Visualize a Dataset
Primary Actor: User
Description: Describe the process of how a user visualizes a dataset.
Precondition: The user must have successfully accessed the website.
Postcondition: If successful, the user will have a visualization of the data that they chose.

Main scenario:

1. The user signs into their account.
2. The user's credentials are verified.
3. The user selects a dataset to use from a drop-down list.
4. The user manipulates the data by filtering and cleansing.
5. The user selects the sensor(s) from the dataset that they want to use.
6. The user chooses the chart type for their sensor(s)
7. The user can add alarms and alarm management to their system
8. The user can add a training model for data prediction to their model which includes selecting an algorithm and parameters
9. Once all choices are made, the user can visualize the data they chose along with the alarm log.

Extensions:

1 a. User doesn't exist

1 a 1. System prompts the user to sign up

1 b. User selects to upgrade to premium user

1 b 1. System brings the user to a page to pay and upgrade their account.

9 a. The user can download the visualization if they choose.

9 a 1. The system brings up a screen for the user to define the save path.

9 b. The user can download the alarm log if they choose.

9 b 1. The system brings up a screen for the user to define the save path.

Premium user scenario

1. The premium user signs into their account.
2. The credentials are verified.
3. The premium user can upload their own dataset to analyze which brings them to the select dataset page with the uploaded dataset (they can still select a default dataset).
4. The premium user follows the same path as the main scenario steps 3-9.

User Stories

As a regular user, I want to test the software for free so that I can decide if it is worth it to purchase.

Acceptance Criteria: Users can select a preset database from a drop-down menu to perform the software functionalities.

As a regular user, I want to be able to filter and cleanse my data to remove any irregularities as I see fit so that I can produce the best visualization possible without any incorrect data.

Acceptance Criteria: The user can manipulate the selected database in a table after selecting it to fit their expectations.

As a regular user, I want to be able to use different visualization types so that I can adjust the visualization to the data I use.

Acceptance Criteria: The user can select what type of graph from a drop-down menu to visualize their data on.

As a regular user, I want to be able to personalize my visualization if necessary so that it can be more useful and easier for my use.

Acceptance Criteria: The user can manipulate the axis' on the graph if they do not want to use the recommended/preset axis'.

As a regular user, I want to be able to plot the direction of my data to be able to predict the future results

Acceptance Criteria: The user will select from a dropdown a subset of prediction algorithms, this will then display the line or bar on the current sensor plot that shows the direction of the data in the future based on past data

As a regular user, I want to visualize multiple sensors at once so I can compare them

Acceptance Criteria: After the user uploads or selects data the user can select multiple sensors, these sensors are then compared with visualizations, and these visualizations are displayed next to one another for display.

As a regular user, I want to be able to set alerts so that I know when sensors have reached dangerous values

Acceptance Criteria: For each sensor, a user can create a threshold so that a notification is sent when the sensor value exceeds that threshold in the past or the future when prediction algorithms are used.

As a regular user, I want to be able to set an alarm management system so that I can prevent common alarm problems such as chattering.

Acceptance Criteria: The user can implement delay timers and deadbands into the alarm selection screen with buttons and input parameters.

As a regular user, I want to be confident that what I am seeing is actually what the data looks like so my decisions are data-driven

Acceptance Criteria: When a user predicts data using an algorithm there will be a range of accuracy as time goes forward to indicate uncertainty in the prediction

As a regular user, I want to be able to download the visualizations that I have made so that I can save them for future use.

Acceptance Criteria: Once visualized, there will be a button to download the visualization and alarm log to the local device which, when clicked, will bring up a screen to select the save location.

As a premium user, I want my uploaded data to be exclusive to me so that I can ensure my data is not distributed.

Acceptance Criteria: Uploaded data will be temporarily stored in the database until it is killed when the user signs out.

As a premium user, I want to be able to upload my data so that I can visualize it

Acceptance Criteria: The user clicks on a box that says upload, a secondary window pops up with their file explorer, and the user can select only an XLSX or CSV file to upload.

As a premium user, I want to be able to select my data online from Thinkspeak so that I don't have to upload it

Acceptance Criteria: The user will click on a box that says upload data, this box will allow the user to add a channel ID from "Thinkspeak" and pull data from the "Thinkspeak platform"