

Software Requirements Specifications Document

MLOps enabling anomaly detection in real-time data streams.

- Imami
- Tanmay Goyal
- Soveet Nayak
- Lakshmi Girija

BRIEF PROBLEM STATEMENT:

Sensors (sensing flow and pressure) monitor the water distribution network and generate high-velocity data that must be examined for abnormalities in real-time. Sensor data streams are aggregated into cohorts and evaluated together for aberrant behaviour by DeepAR or TFT. Using machine learning, we are meant to find and categorise abnormalities in high-granularity data streams.

SYSTEM REQUIREMENTS:

- Azure VM
- Python + Flask
- React JS
- Javascript
- MongoDB
- Plotly

USERS PROFILE:

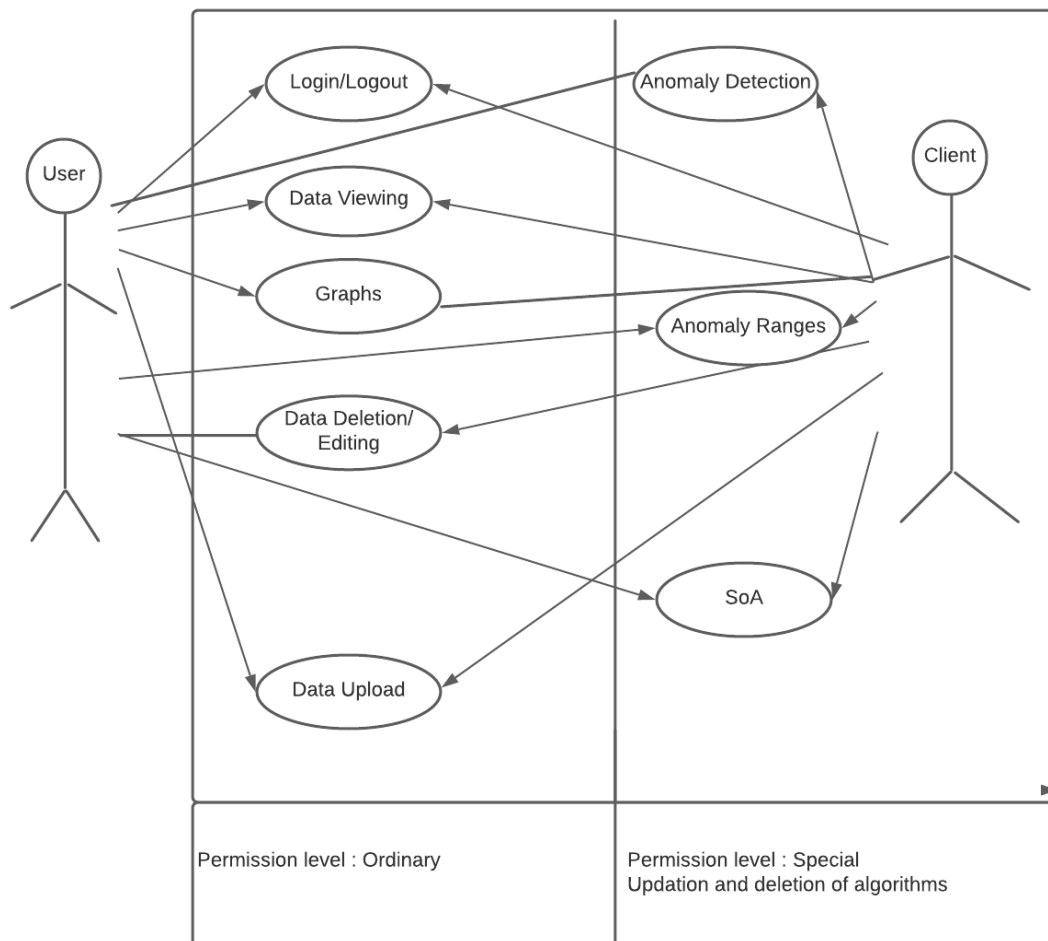
- The Users with water pipelines data can upload and view graphs and processed data signalling errors or anomalies.
- The Client(client here means people at Smartterra) will be able to access the graphs of water flow and pressure of different pipelines and will be able to detect the anomaly analysis of all the water pipelines in the website. Can access to the database of any location without restriction and inform the necessary people of errors and leakages.

FEATURE REQUIREMENTS:

No.	Use case name	Description	Release
1	Login/Logout	The user should be able to Login to/Logout of the website. (will be implemented at the end if required)	R2
2	Data Upload	The user will be able to upload a CSV file to the website.	R1
3	Data Viewing	<ul style="list-style-type: none">• The user can access data of pipelines at any instant.• And we should also store the data of previous months.• Data is presented in the form of graphs.	R1
4	Algorithms	<ul style="list-style-type: none">• User can choose form multiple algorithms to check anomalies.	R2
5	Anomaly Detection	<ul style="list-style-type: none">• Anomalies should be visible to the user• There should be multiple algorithms for anomaly detection and the user should be able to choose the algorithm.	R2
6	Graphs	The data when asked should be displayed in terms of graphs for different months and different algorithms.	R2
7	Anomaly Ranges(AR)	Given some data, we should be able to predict anomaly ranges and take input the ranges from the user too .	R1

8	Severity Of Anomaly(SoA)	Anomalies with different severity should be displayed according to the number of algorithms that predict an anomaly in the data.	R2
9	Data Deletion/Editing	The user will be able to remove or edit the data.	R2

USE CASE DIAGRAM:



USE CASE DESCRIPTION:

Use Case Number	UC - 01
Use case Name	Login/Logout
Overview	Users can login into the website using their credentials and easily logout.
Actors	Users, Clients
Pre Condition	None
Flow	Credentials are checked against a database to authenticate the user.
Post condition	The User is logged in and can access the website depending on his/her access level.

Use Case Number	UC - 02
Use case Name	Data Upload
Overview	Users, Clients can upload the data of pipelines across a period of time
Actors	Users, Clients
Pre Condition	1) The users and the clients must be logged in to upload the data
Flow	1) To upload the actor must click on the upload tab 2) We take the data from CSV in react, and put it into MongoDB through flask
Post condition	None

Use Case Number	UC - 03
Use case Name	Data Viewing
Overview	Users, Clients can access the data of pipelines at any instant
Actors	Users, Clients
Pre Condition	<ol style="list-style-type: none"> 1. The users and the clients must be logged in to view the data 2. The actor must have access to requested data
Flow	<ol style="list-style-type: none"> 1. To view the actor must click on the data tab 2. We fetch data from our database and display it to the actor
Post condition	None

Use Case Number	UC - 04
Use case Name	Algorithms
Overview	Users can choose from multiple algorithms.
Actors	Users
Pre Condition	<ol style="list-style-type: none"> 1. The users and the clients must be logged in to view the data 2. The actor must have access to requested data
Flow	<ol style="list-style-type: none"> 1. To view the actor must click on the algorithm they want to run 2. We fetch data from our database and display it to the actor

Post condition	None
-----------------------	------

Use Case Number	UC - 05
Use case Name	Anomaly Detection
Overview	The data is tested by the ML engine against a trained model.
Actors	None
Pre Condition	Users upload data to the database successfully in a certain format.
Flow	ML engine checks the data for irregularities against previous statistics and suggests possible errors.
Post condition	Feedback

Use Case Number	UC - 06
Use case Name	Graphs
Overview	The data when asked should be displayed in terms of graphs.
Actors	Users, Clients
Pre Condition	<ol style="list-style-type: none"> 1. Users and clients must be logged in 2. The actor must have access to the respective data
Flow	<ol style="list-style-type: none"> 1. To view the user must select graph in the tab 2. We fetch the data and display it in graphs

Post condition	None
-----------------------	------

Use Case Number	UC - 07
Use case Name	Anomaly Ranges
Overview	Given some data, we should be able to predict anomaly ranges and take input the ranges from the user too .
Actors	Users, Clients
Pre Condition	The data should already be present in the database
Flow	The actor will be able to view an anomaly range predicted or would be able to provide an anomaly range.
Post condition	The anomaly ranges can be reset post changes

Use Case Number	UC - 08
Use case Name	Severity of Anomaly (SoA)
Overview	Anomalies with different severity should be displayed according to the number of algorithms that predict an anomaly in the data.
Actors	Users, Clients
Pre Condition	The data is already given or present in the database.
Flow	The actor will be able to preview and

	understand the Severity of an Anomaly and take required actions.
Post condition	The User/Client can see a report of Severity of Anomalies in the data provided by the User through a CSV file.

Use Case Number	UC - 09
Use case Name	Data Deletion/Editing
Overview	The user will be able to remove inconsistencies in the data.
Actors	Clients
Pre Condition	The actor must be logged in
Flow	<ol style="list-style-type: none"> 1. The actor must select delete/edit the data either of the options 2. Then the respective data must be selected 3. The new value of data should be selected in case of editing
Post condition	None

TEST BACKLOG:

No backlog as of yet.