# HIS - Safety Critical Systems:
# Project Report

Students: Mrinal Tyagi (1383988), Nikhil Bajaj (1392872),
Shobhit Tiwari (1387366), Shounak Ozarkar (1386299)
Professor: Dr. Matthias Wagner

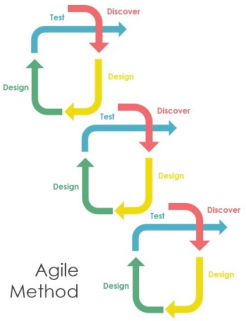
February 4, 2022

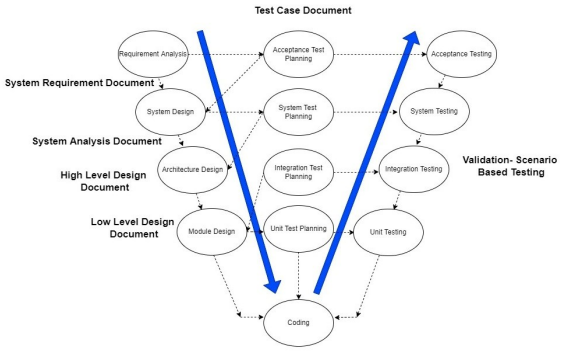
## 1 Project Information:

**Hydraulic Test Rig** consists of a primary working and secondary cooling-filtration circuit which are connected via oil tank. System cyclically repeats constant load cycles of 60s and measures process values of pressures, volume flows and temperatures while the condition of hydraulic components(cooler, valve, pump and accumulator) is quantitatively varied. Predictive maintenance enables the maintenance frequency to be minimal to avoid unexpected maintenance.

## 2 Process Model:

**Agile** scrum software development(Vmodel XT) is often carried out in short and quick cycles which results in very frequent incremental releases, with each release building on previous features.




## 3 Team Organisation

| Serial No. | Project Tasks | Contributions |
|---|---|---|
| 1 | Requirement Gathering and Analysis | Team |
| 2 | Project Domain | Shounak |
| 3 | Project Architecture and Design | Nikhil, Shobhit |
| 4 | Cost Estimations | Mrinal, Shounak |
| 5 | Mathematical Model | Nikhil |
| 6 | Ruleset and Business Logic Modeling | Team |
| 7 | Safety Requirement and Constraints | Nikhil, Shobhit |
| 8 | Safety Plan | Nikhil, Shobhit |
| 9 | Hazard Analysis | Shobhit |
| 10 | Coding and Development | Team |
| 11 | HMI | Mrinal |
| 12 | Testing | Mrinal,Shounak |
| 13 | Documentation | Nikhil, Shobhit |

## 4 Dataset Information:

A timeseries dataset where system cyclically repeats constant load cycles of 60s and measures process values such as pressures, volume flows and temperatures while the condition of 4 hydraulic components (cooler, valve, pump and accumulator) is quantitatively varied.

# 5 Use cases for Industry 4.0

- **Use Case:** We intend to developed an application which can assist in predictive maintenance of **Hydraulic Test Rig**

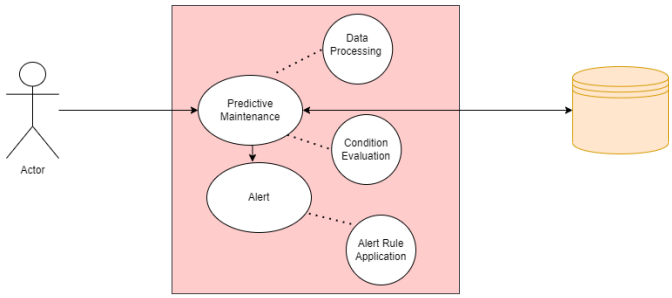
| **Use-Case 1:** |
|---|
| **Title:** Start of Application |
| **Description:** The application will start at the 'Home Page' page. |
| **Primary Actor:** Client/ User |
| **Precondition:** NA |
| **Postcondition:** Application starts successfully. |
| **Main Success Scenario:** Application starts smoothly, and user is on the landing page. |

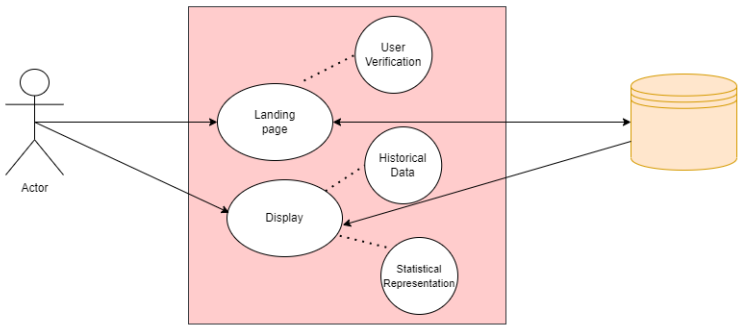
| **Use-Case 2:** |
|---|
| **Title:** Starting 'Hydraulic Suspension Failure Detection and Maintenance' |
| **Description:** Application will convey the current status of Machine as well as various sensors attached. |
| **Primary Actor:** Client |
| **Precondition:** Client has triggered 'Hydraulic Suspension Failure Detection and Maintenance' |
| **Postcondition:** The user gets notified if the suspension or it's parts need maintenance with an alert window. |
| **Main Success Scenario:** The user gets notified if the suspension is healthy / needs maintenance or the current state of various sensors |

| **Use-Case 3:** |
|---|
| **Title:** Historical Data |
| **Description:** The application should display timestamped logs to the user and will be able to see higher level metrics. |
| **Primary Actor:** Database |
| **Precondition:** System has already been triggered by the client at least once |
| **Postcondition:** The application should display relevant logs to the client |
| **Main Success Scenario:** Logs and higher-level metrics are displayed to the client |

| **Use-Case 4:** |
|---|
| **Title:** Alert window for notifying the client |
| **Description:** The application will inform the user about the maintenance status of the suspension with a pop-up. |
| **Primary Actor:** Application |
| **Precondition:** Client has triggered system using the Start button |
| **Postcondition:** The user gets notified with an alert displaying the current status of the suspension and if the machine needs predictive maintenance. |
| **Main Success Scenario:** The user gets notified if the machine is not healthy / needs predictive maintenance. |

| **Use-Case5:** |
|---|
| **Title:** Predicition of Machine Status |
| **Description:** The application should be able to predict the current machine status as well as machine status 15 mins from current time |
| **Primary Actor:** Machine Learning Models |
| **Precondition1:** Trained and tested ML Models |
| **Precondition2:** Transformed data must be sent to ML models. |
| **Postcondition:** The ML models must be able to accurately predict the machine status. |
| **Main Success Scenario:** Higher accuracy has been achived in indentifying overall trend of machine condition. |



# 6 Schedule with milestones

| Activities | Week01 | Week02 | Week03 | Week04 | Week05 | Week06 | Week07 | Week08 | Week09 |
|---|---|---|---|---|---|---|---|---|---|
| *Requirement Analysis and Planning* | | | | | | | | | |
| *Design and Documentation* | | | | | | | | | |
| *Development* | | | | | | | | | |
| *Testing and Bug Fixes* | | | | | | | | | |
| *Release & Deployment* | | | | | | | | | |

# 7 Risks to the Project

- Understanding scope and requirement analysis
- Real-time behaviour of connected devices
- Maintenance and regression testing during working phase
- User Interactions with end system

# 8 Software Configuration Management

- Trello-https://trello.com/b/OkuGNx6F/industry-40
- Github-https://github.com/Nikhil-Bajaj/SafetyCriticalSystem_PredictiveMaintenance.git.

# 9 Requirement Management

- **Functional Requirement**
  1. The application shall display statistical inferences drawn from the data, predict metrices for the status after 15min along with the threshold that will be used to trigger predictive maintenance warning.
  2. The application shall predict whether system is healthy, or maintenance is required.
  3. The application should display machine status based on the sensor values using graphs.
- **Non-functional Requirement**
  1. The application shall continuously log information which can be used to analyse system failures.
  2. Volume of data should not restrict system performance.
  3. The application shall be platform independent and ensure the privacy and security of the data.

# 10 Software Requirement

- Programming Language: Python 3
- Web Technologies: HTML, CSS and Javascript
- Database: Postgres
- Web Server: Grafana Server
- IDE/GUI Tool: Anaconda, Grafana 8
- Operating System: Windows/Linux/Mac

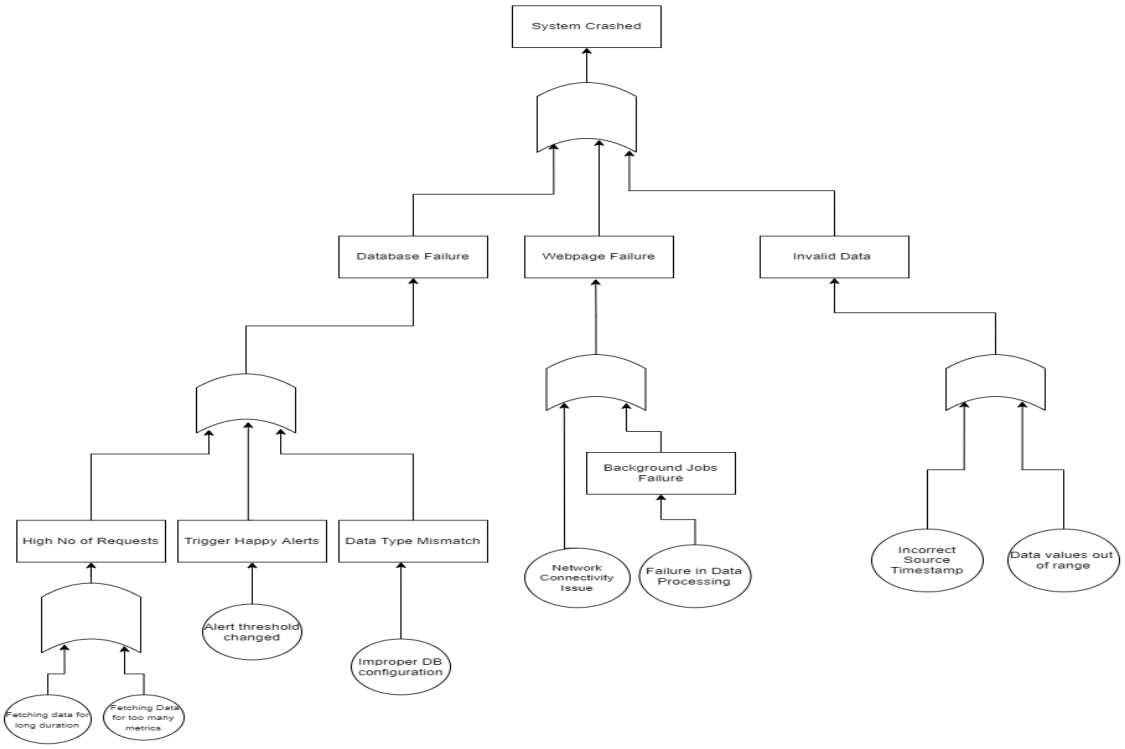
# 11 System Constraints

- Alert Rule needs to be updated over time with more data and sensor´s condition over lifespan
- Connectivity is most important for any system be it Data base or Network
- Data Collection and its processing at Real-time may be a challenge.

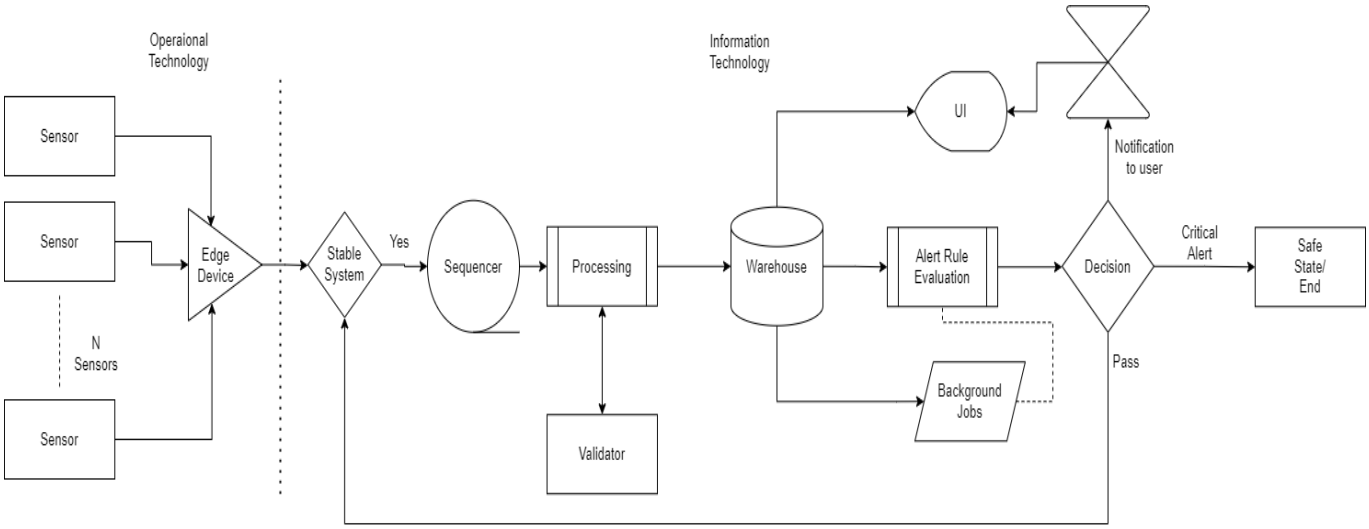
# 12 Safety Constraints

- Ensuring proper Database connectivity.
- Implementing Transaction management techniques
- Handling Webpage failure due to connectivity problems
- Proper error/ exception handling for failed data processing.
- Data validation logic for fine tuning data.
- Scheduler for data pipeline

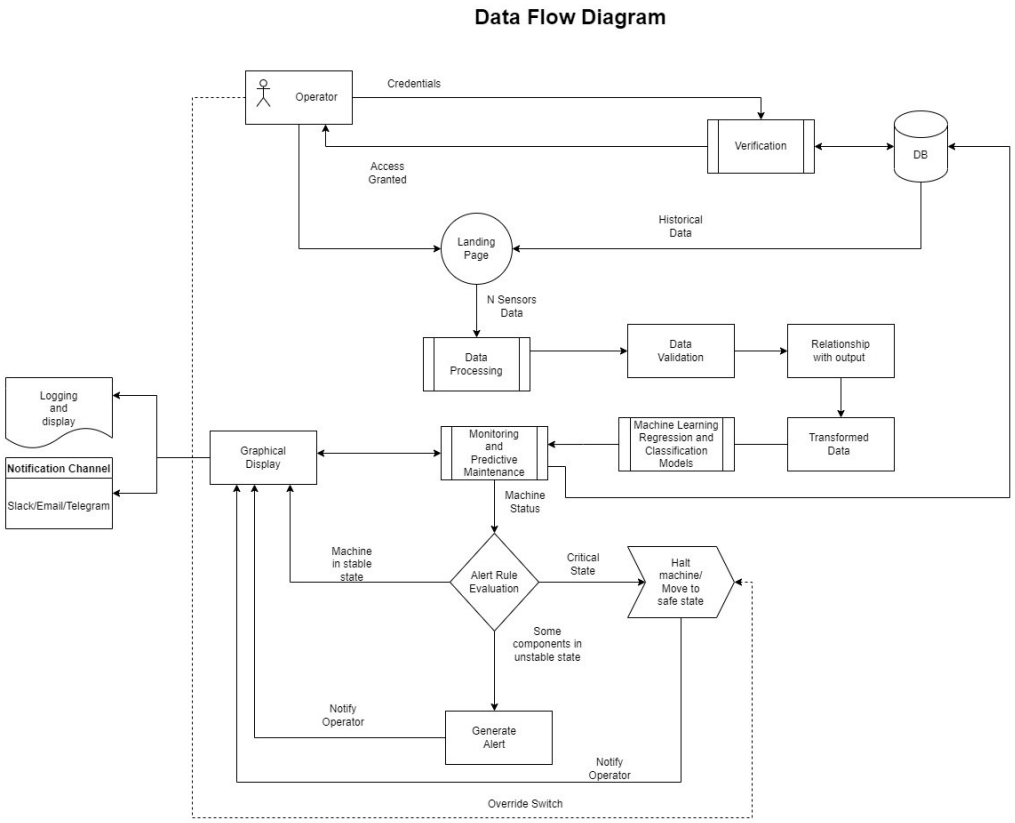
# 13 Hazard Analysis using Fault tree analysis (FTA)



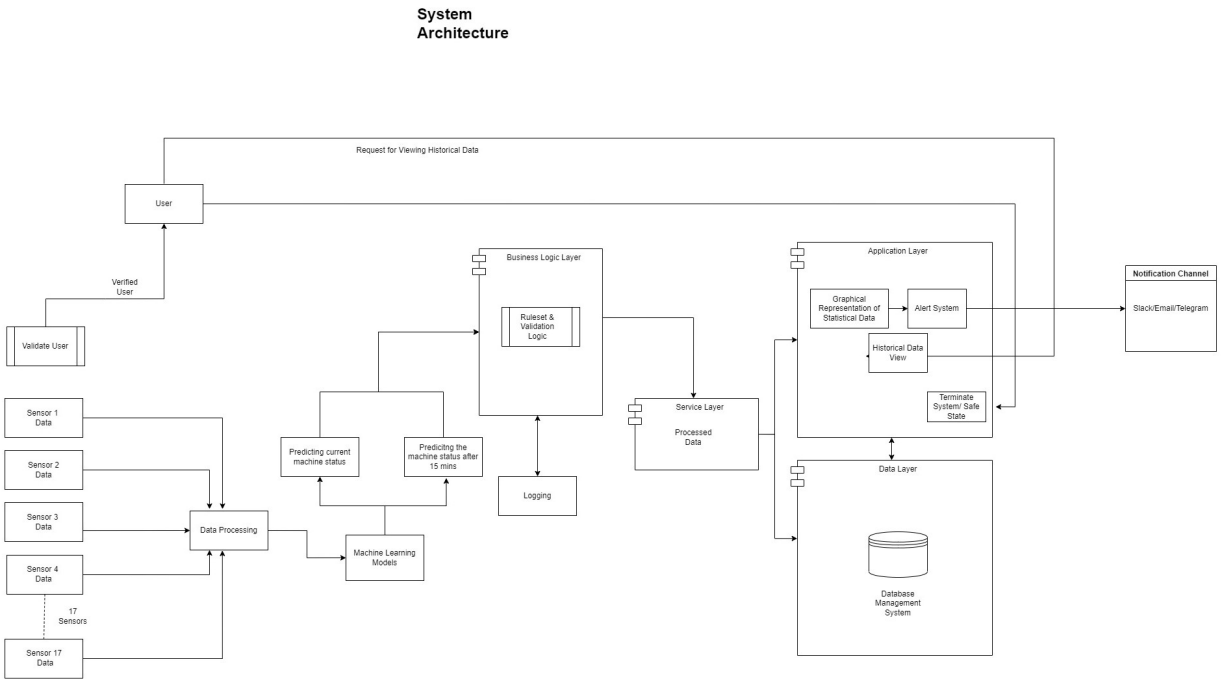

# 14 Control Flow



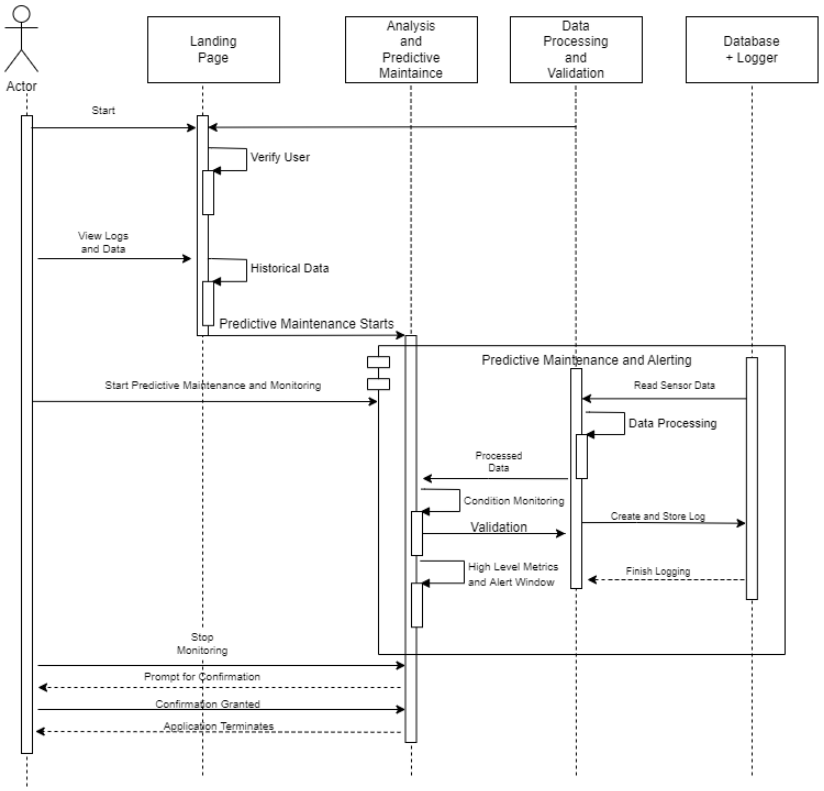
# 15 Data Flow



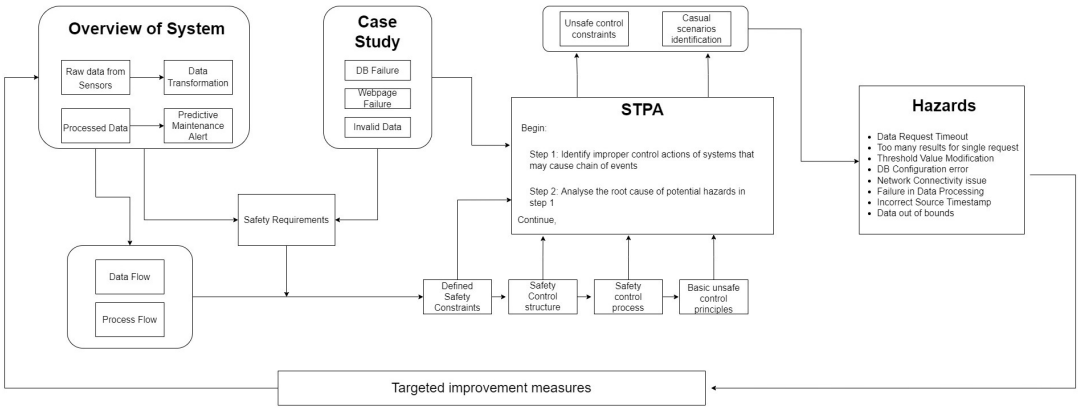

# 16 System Architecture

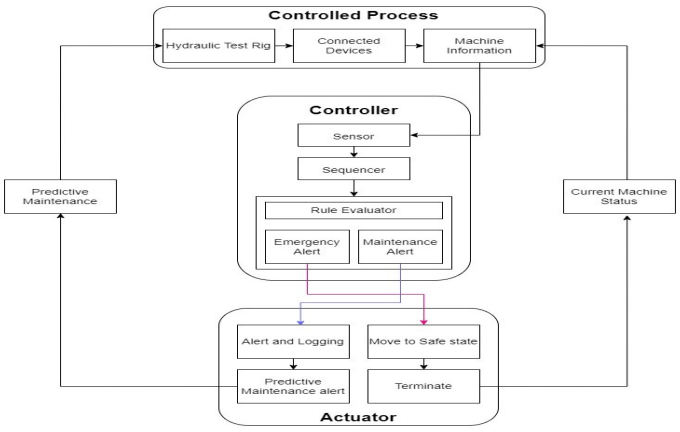


# 17 Sequence Diagram




# 18 STAMP



# 19 COCOMO (Constructive Cost Model) :

Calculating Unadjusted Functional Point

| Function type | Simple | Average | Complex | Considered For Project | Count | Total |
|---|---|---|---|---|---|---|
| External Inputs | 3 | 4 | 6 | 3 | 2 | 6 |
| External Output | 4 | 5 | 7 | 4 | 2 | 8 |
| External Inquiries | 3 | 4 | 6 | 3 | 2 | 6 |
| Internal Logical Files | 7 | 10 | 15 | 10 | 3 | 30 |
| External Interface Files | 5 | 7 | 10 | 5 | 0 | 0 |
| | | | | | | **50** |

UFP = 50

Calculating Complexity Adjustment Factor (CAF)

| Sr. No | Characteristics | (0-5) |
|---|---|---|
| 1 | Data communications | 4 |
| 2 | Distributed data processing | 0 |
| 3 | Performance | 5 |
| 4 | Heavily used configuration | 0 |
| 5 | Transaction rate | 5 |
| 6 | On-Line data entry | 1 |
| 7 | End-user efficiency | 0 |
| 8 | On-Line update | 0 |
| 9 | Complex processing | 4 |
| 10 | Reusability | 4 |
| 11 | Installation ease | 4 |
| 12 | Operational ease | 3 |
| 13 | Facilitate change | 5 |
| 14 | Multiple sites | 0 |
| | | 35 |

CAF = 0.65+ (0.01 * ∑Fi)

CAF = 0.65+ (0.01 * 35) = 1

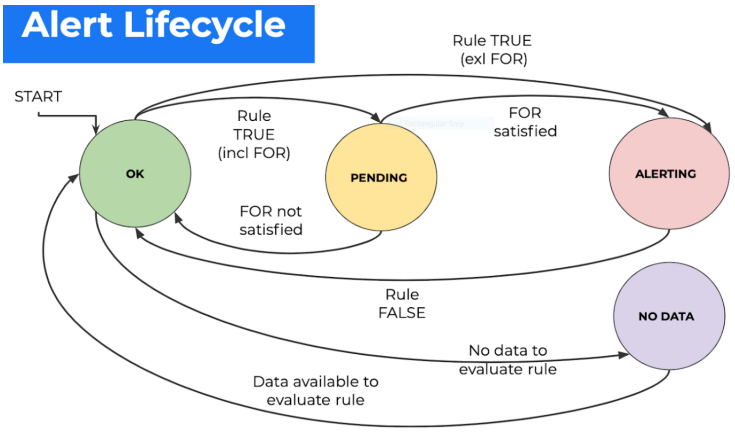
FPC = UFP * CAF

=50*1 = **50.**

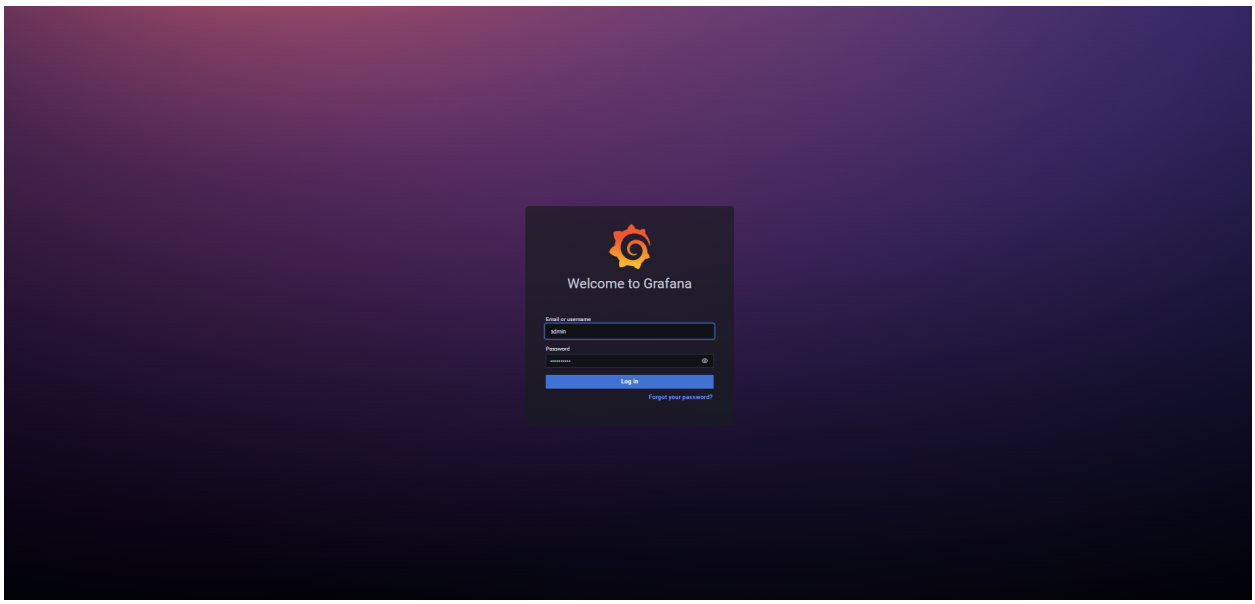
# 20 Safety Plan

| ID | Category | Name | Priority | Description | Impact | Resolution |
|---|---|---|---|---|---|---|
| DB-1 | Database | Fetching data for Long Duration | 5 | This will cause other DB queries in slowing down and even DB failure in case of persistant queries | Restrict request data time range | Transaction Management |
| DB-2 | Database | Fetching data for too many metrics | 4 | This will cause other DB queries in slowing down and even DB failure in case of persistant queries | Restrict number of metrices in the request data | Transaction Management |
| DB-3 | Database | Alert Threshold updated | 2 | This will cause load on Alert rule evaluation engine, and ultimately DB as well | Suppress/ over ride alert criteria | Multiple Sensor Alerts or Continous alerts from same sensors |
| DB-4 | Database | Improper DB Configuration | 2 | This will break the data processing pipeline and eventually put load on the DB the pipeline is continously failing | Suppress data transformation for this particular machine until a proper fix is deployed | |
| W-1 | Frontend | Network Connectivity Issue | 2 | Webpage will fail to load for requested parameters | User would not see requested metrices on web page | Ensuring proper network connectivity and informing user in time about the issue |
| W-2 | Frontend + Backend | Failure in Data Processing | 5 | The webpage may show incorrect data or even fail to load | User would see exceptions/errors on web page | Use exception handling and data validation process. |
| INV-1 | Backend | Data values out of range | 4 | Alert rule will be useless as it may not have any meaning | Alert for sensor maintainance and Data Validation should be checked | Proper Data validation |
| INV-2 | Backend | Data has a time offset | 3 | Wrong interpretation of the data or may even break the transformation pipeline | Rectify time sync and run transformation again. Check for Sensor Maintenance | Sensor Downtime for sometime |

# 21 Alert Lifecycle



# 22 HMI Design





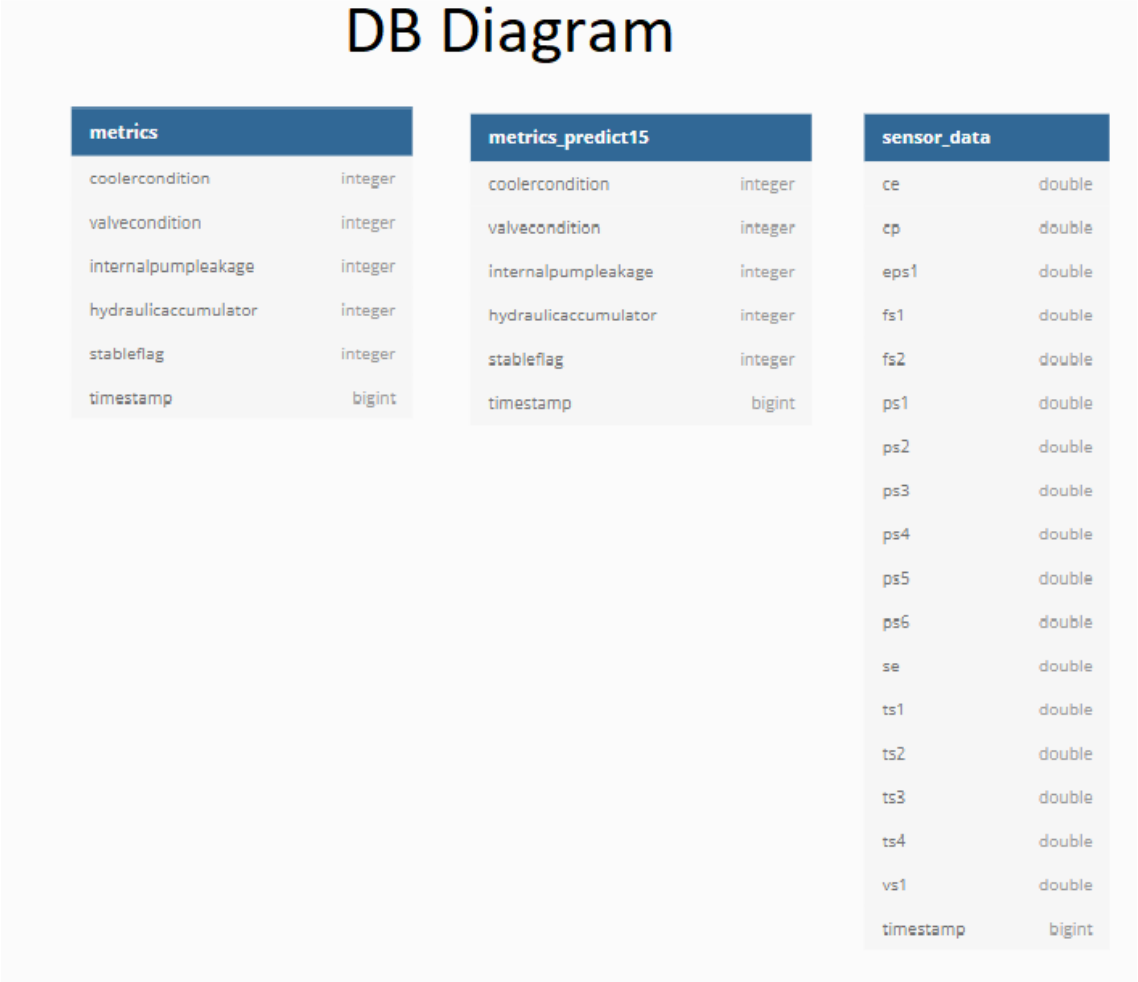
Predicted Panel

| Time | coolercondition | valvecondition | internalpumpleakage | hydraulicaccumulator | stableflag |
|---|---|---|---|---|---|
| 2022-01-31 10:41:01 | 100 | 100 | 0 | 90 | 1 |
| 2022-01-31 10:41:03 | 100 | 100 | 0 | 90 | 1 |
| 2022-01-31 10:41:04 | 100 | 100 | 0 | 90 | 1 |
| 2022-01-31 10:41:05 | 100 | 100 | 0 | 90 | 1 |
| 2022-01-31 10:41:06 | 100 | 100 | 0 | 90 | 1 |

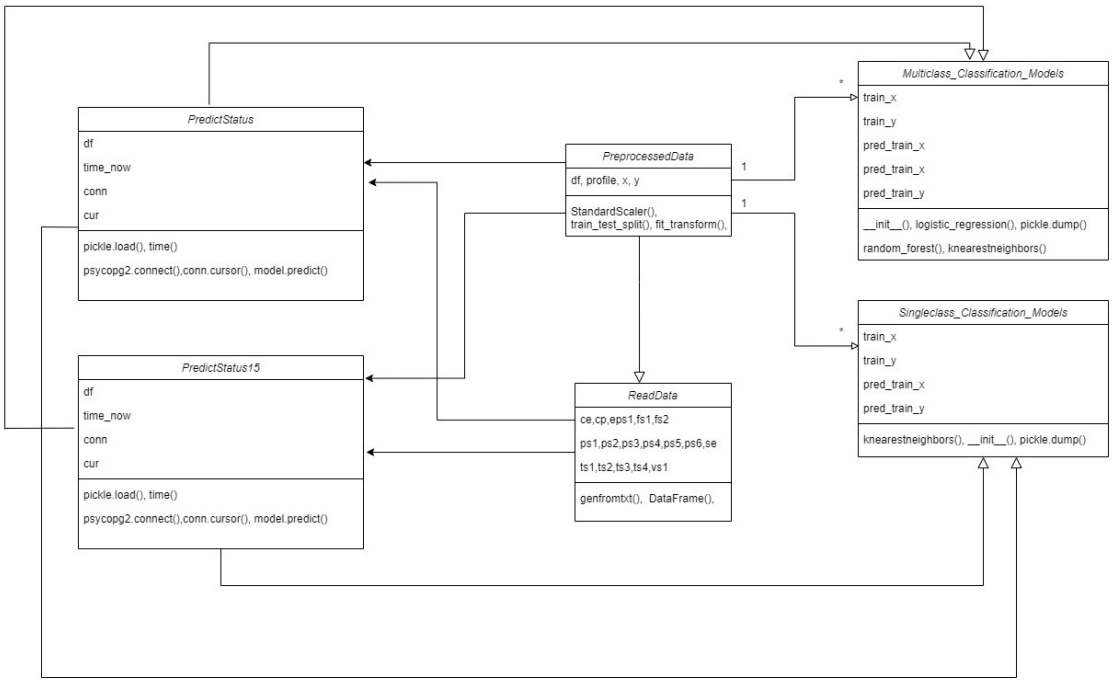
Panel Title

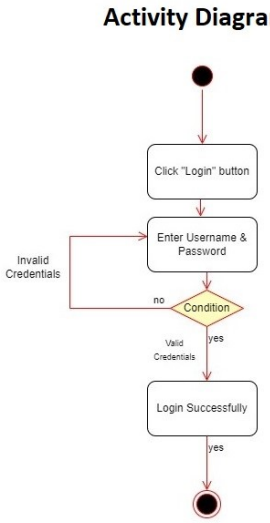
| Time | coolercondition | valvecondition | internalpumpleakage | hydraulicaccumulator | stableflag |
|---|---|---|---|---|---|
| 2022-01-31 10:41:00 | 100 | 100 | 0 | 90 | 1 |
| 2022-01-31 10:41:01 | 100 | 100 | 0 | 90 | 1 |
| 2022-01-31 10:41:03 | 100 | 100 | 0 | 90 | 1 |
| 2022-01-31 10:41:04 | 100 | 100 | 0 | 90 | 1 |
| 2022-01-31 10:41:05 | 100 | 100 | 0 | 90 | 1 |

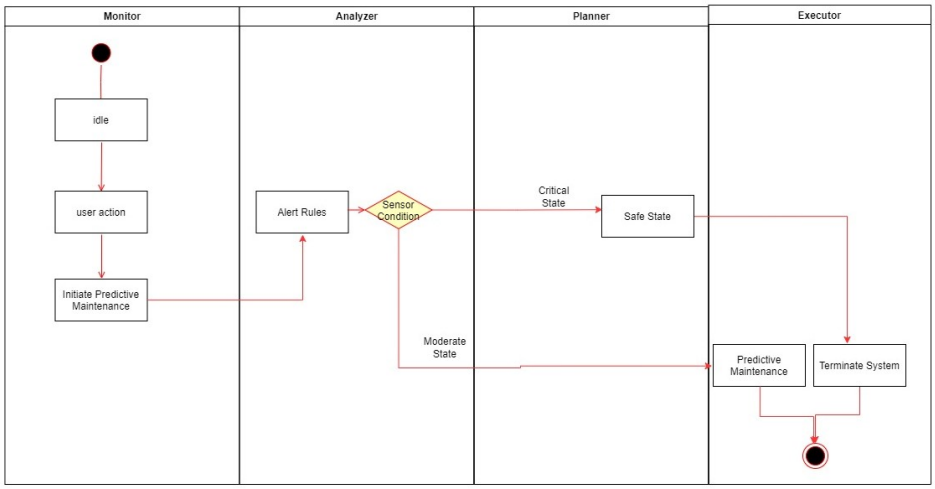
# 23 UML Diagrams




**Class Diagram**






# 24 References

https://ieeexplore.ieee.org/abstract/document/7151267
https://mitpress.mit.edu/books/engineering-safer-world
https://www.ncbi.nlm.nih.gov/books/NBK209656/
https://doi.org/10.1016/j.jisa.2016.05.008
https://doi.org/10.1016/j.proeng.2015.11.498
https://doi.org/10.1007/s11219-017-9396-0
https://shemesh.larc.nasa.gov/iria03/p13-leveson.pdf
https://doi.org/10.1007/s11219-017-9396-0
https://ieeexplore.ieee.org/document/467427
https://scikit-learn.org/stable/
https://www.researchgate.net/publication/313291643_Automatic_feature_extraction_and_selection_for_classification_of_cyclical_time_series_data
http://www.rvs.uni-bielefeld.de/publications/Incidents/DOCS/ComAndRep/Warsaw/warsaw-report.html
https://www.faa.gov/sites/faa.gov/files/2021-08/Final_JATR_Submittal_to_FAA_Oct_2019.pdf
https://www.researchgate.net/publication/3974620_Application_of_hazard_analysis_to_software_quality_modelling