# Requirements

Sdmay23-06

Colton Carlson, Jay Arnold, Hayden Havelka, Connor Linn, Noah Meyer

# Functional Reqs

- When a request is made the system shall produce a new report
- When an email request is made, the system shall email the most recently made report to the specified email address
- Where the line number is included, the system shall only look in the line specified for the work order.
- When the time turns 11:59 PM CST the system shall email a report for every work order that finished during the previous 24 hours.
- If a work order does not exist, then the system shall notify the user.
- When a report is produced, then any locally obtained data shall be deleted.
- When a report is requested, add date, time, and work order number to the log file
- When a report is finalized, add date, time, and work order number to the log file
- When a report is emailed, add date, time, and work order number to the log file
- If a report fails to be generated, then add date, time, work order number, and error to the log file, then retry.
- If a report fails to be generated 5 times in a row, display the error message on the front end for the user to see.
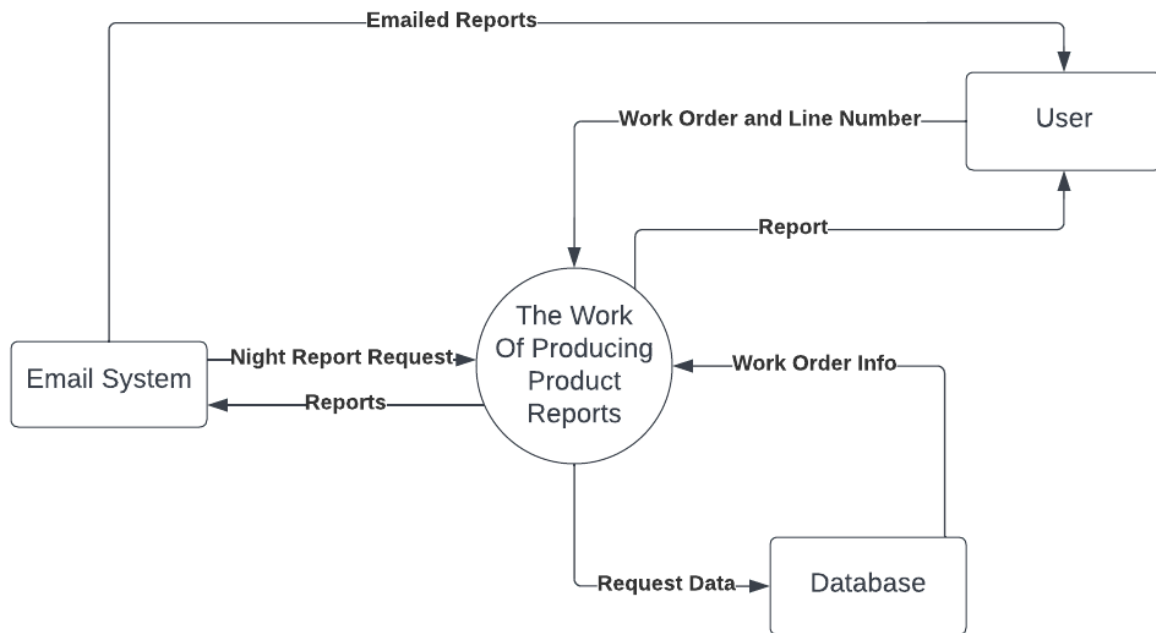
# Non Functional Reqs

- The system shall produce a report in 15 seconds. ***Constraint***
- The system shall produce a PDF report based on the template decided on. ***Constraint***
- The system shall produce a PDF that contains the machine data from the requested work order.
- The system shall use company colors. ***Constraint***
- The system shall be able to support new lines/components.
- The systems frontend shall use React/Node.js libraries  ***Constraint***
- The systems frontend shall be accessible from Mac and Windows.  ***Constraint***
- The system shall run on Safari, Chrome, Firefox, Opera, Edge. ***Constraint***
- The system's backend shall run on a Windows 2019 Server. ***Constraint***
- The system shall not include null/zero values in the report.
- The system shall use FTP to send the report to the front end. ***Constraint***
- The system shall use the internal email system to send emails.
- The system shall use IMAP as the protocol for sending emails. ***Constraint***
- The system shall use OPC UA as the protocol for communicating with Canary ***Constraint***
    - This allows canary and backend communication to be OS independent.
- The system shall have access to the work machine's data via the Canary Database.

# IEEE Standards

- IEEE 26515-2018 - Agile Development Cycle
    - Developmental cycle we will use. Allows us to be flexible with the customer and have clear constant deliverables.
- IEEE 829 - Software Test Documentation
    - Allows us to easily document our tests, easily shows what test does what and explains expected results.
- IEEE P1363 - Public Key Cryptography
    - Allows us to ensure that only authenticated users will request the generated reports.

# Context Diagram:

# Architecture Overview

Requirement
Assignment

Connor
Noah

React/Node JS

Hayden
Jay
Colton
Windows Server 2019

Canary

Database

John

Front End

(HTTP) Workorder #

Backend

OPC UA

(FTP) PDF

OPC UA

Windows     Safari
Mac         Chromium
            Firefox
            Opera
            Edge

Delete Local data
after PDF generation

PDF

Email API

IMAP

John

Colton

Log File

Start/End Times For Report Generation