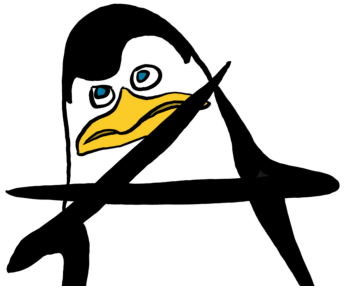


System Observability, Analytics & Insights Platform

Team Kowalski

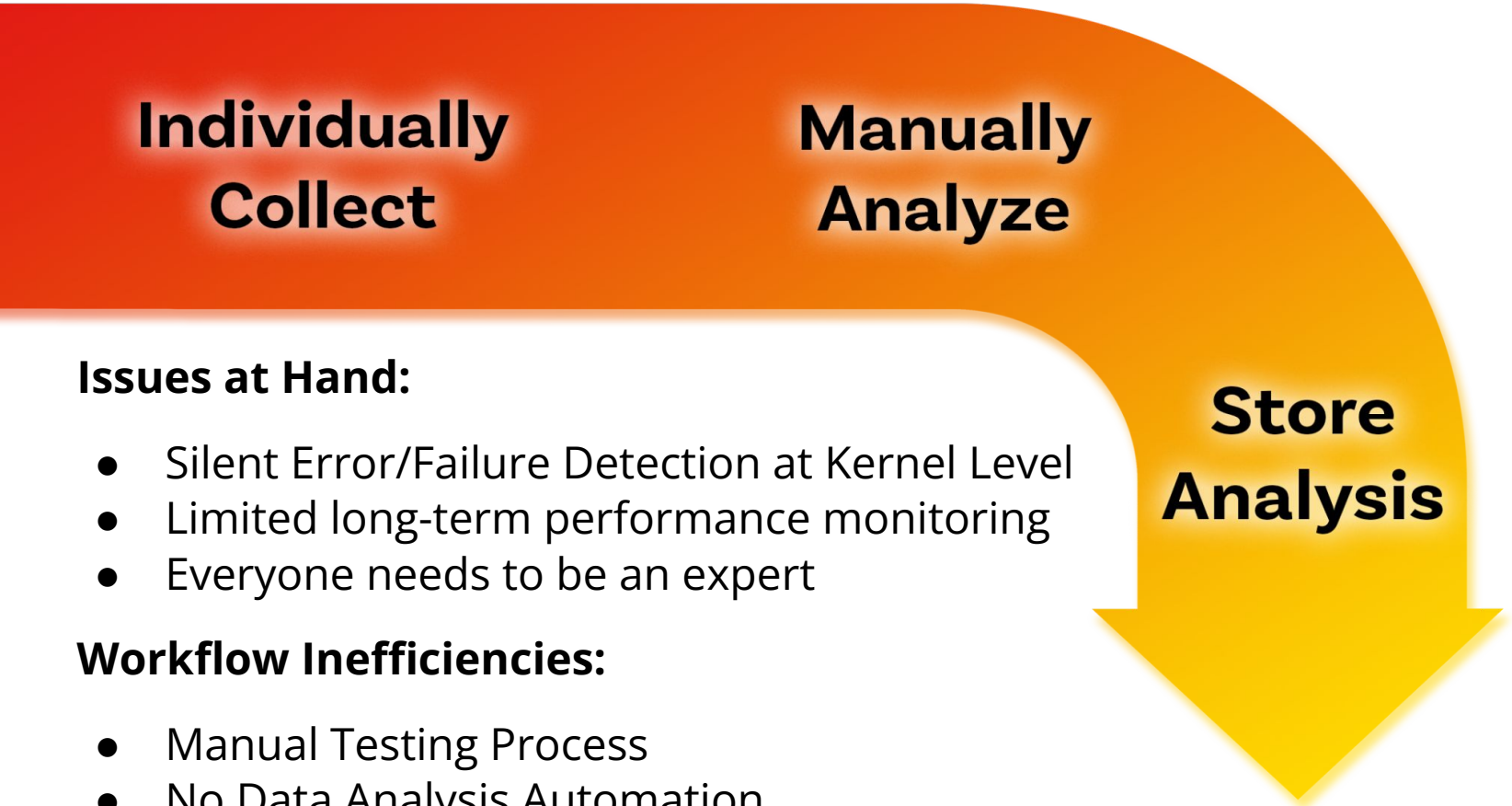
Members

Jake Borneman
> Team Leader
> Testing
> Sanitization

Erick Salazar
> Data Collection
> Data Storage

Bailey McCauslin
> Data Collection
> Sanitization

Nick Wiltshire
> Visual Dashboard Manager



Issues at Hand:

- Silent Error/Failure Detection at Kernel Level
- Limited long-term performance monitoring
- Everyone needs to be an expert

Workflow Inefficiencies:

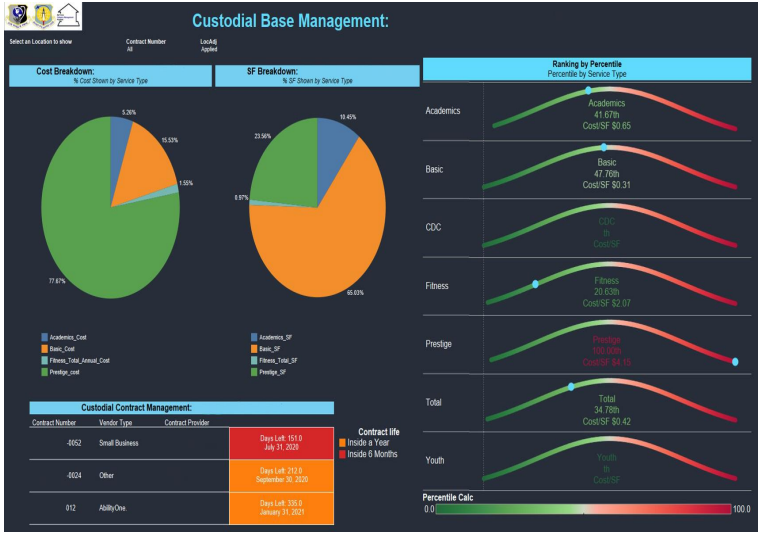
- Manual Testing Process
- No Data Analysis Automation
- Individual Device Testing

The Problem

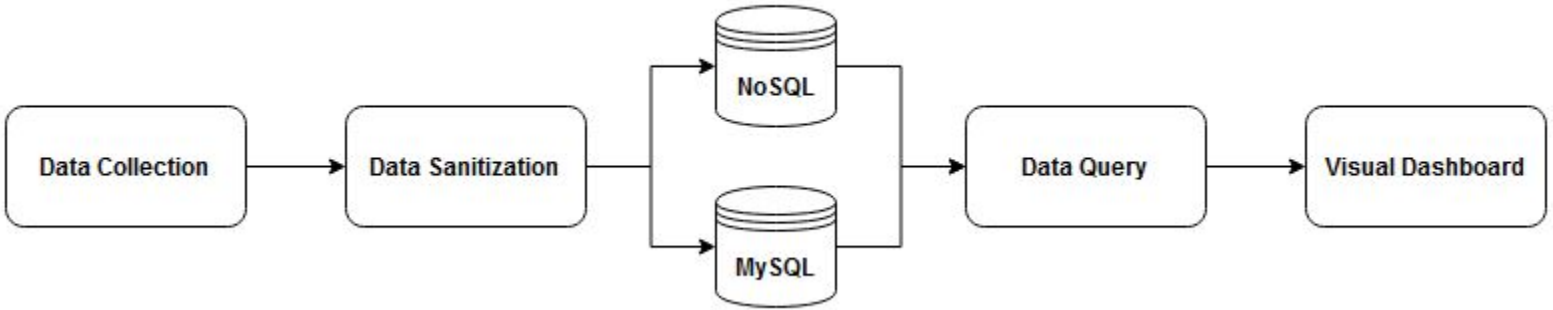
Solution Overview

Comprehensive Data Analytics Dashboard:

- Takes user input for desired test runs(thresholds, what to test, etc.)
- Automates data collection and data storage
- Handles data analysis to be displayed on visual dashboard

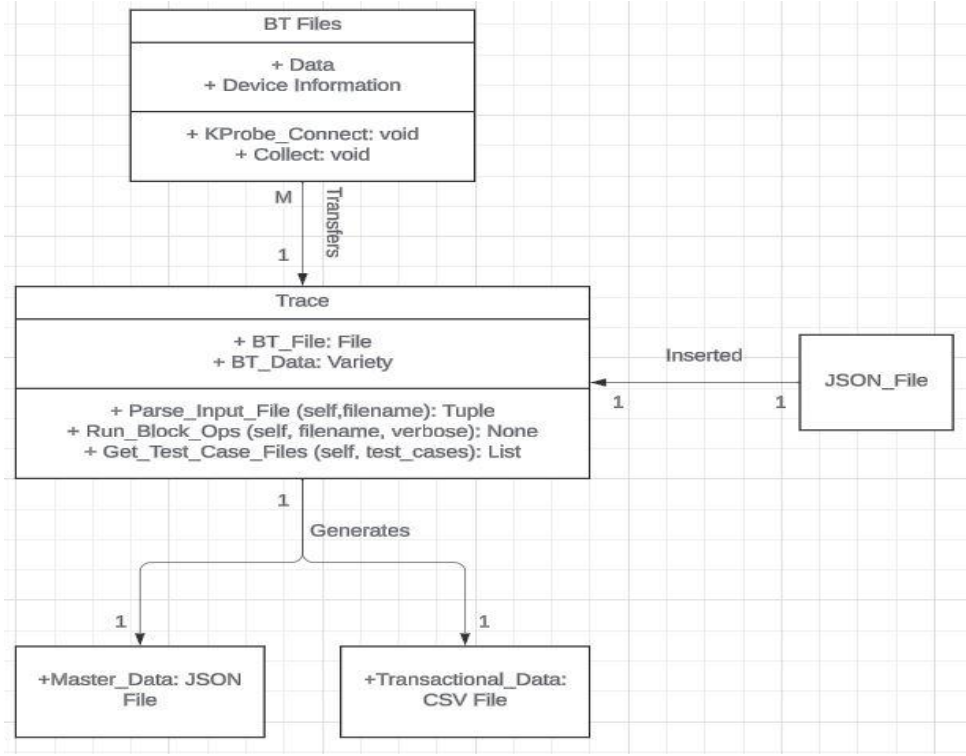


Implementation Overview



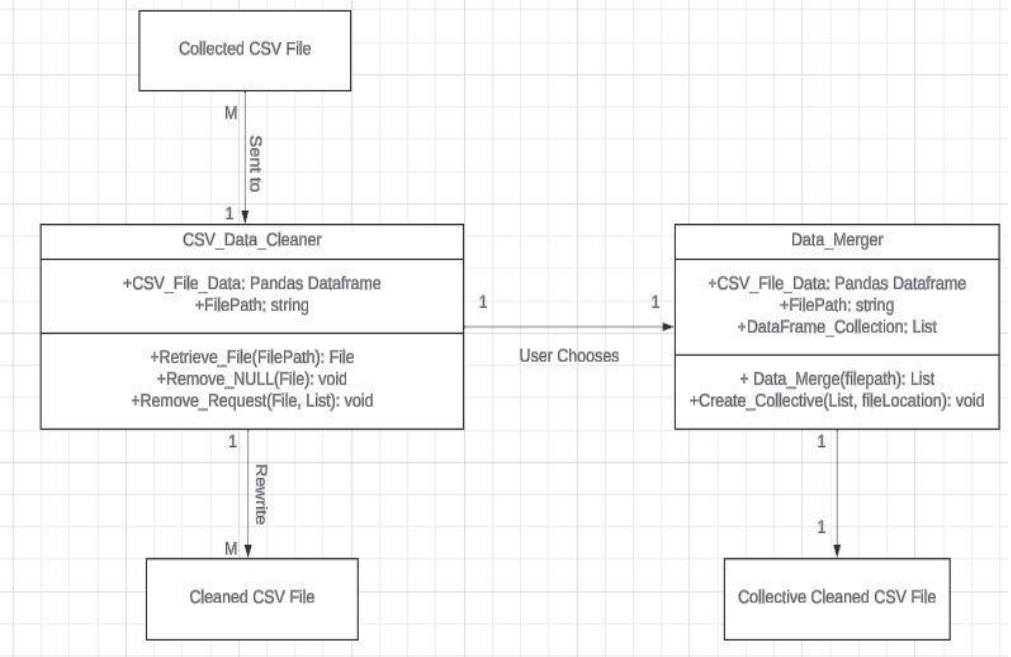
Data Collection Implementation

- Implemented a trace class that:
  - Takes in user inputs via GUI or json file.
  - Load in bt files and start tracing kernel events.
  - Product output files(.csv).
  - Sends them off to different location.
- Supports operations:
  - Biolatency
  - Biopattern
  - Bioerr
  - BlockRQComplete
  - BlockRQError
  - NVMElatency(Not yet)



Data Sanitization Implementation

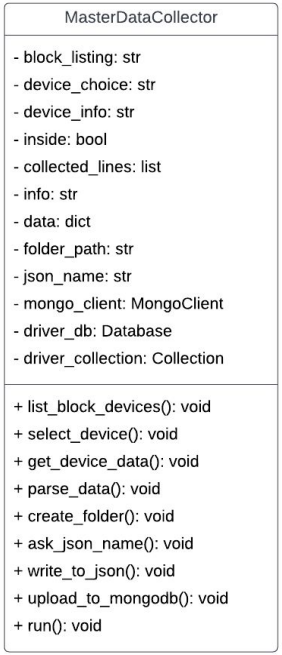
- CSV_Data_Cleaner
  - Retrieves recently collected information
    - Cleans information
    - Overwrites original file
- Data_Merger (If user selects it)
  - Merges all collected information post-Cleaner
  - Used for one file grabs and observation

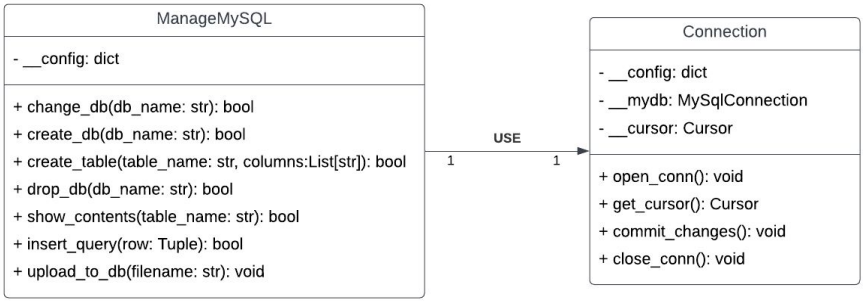


Data Storage Implementation

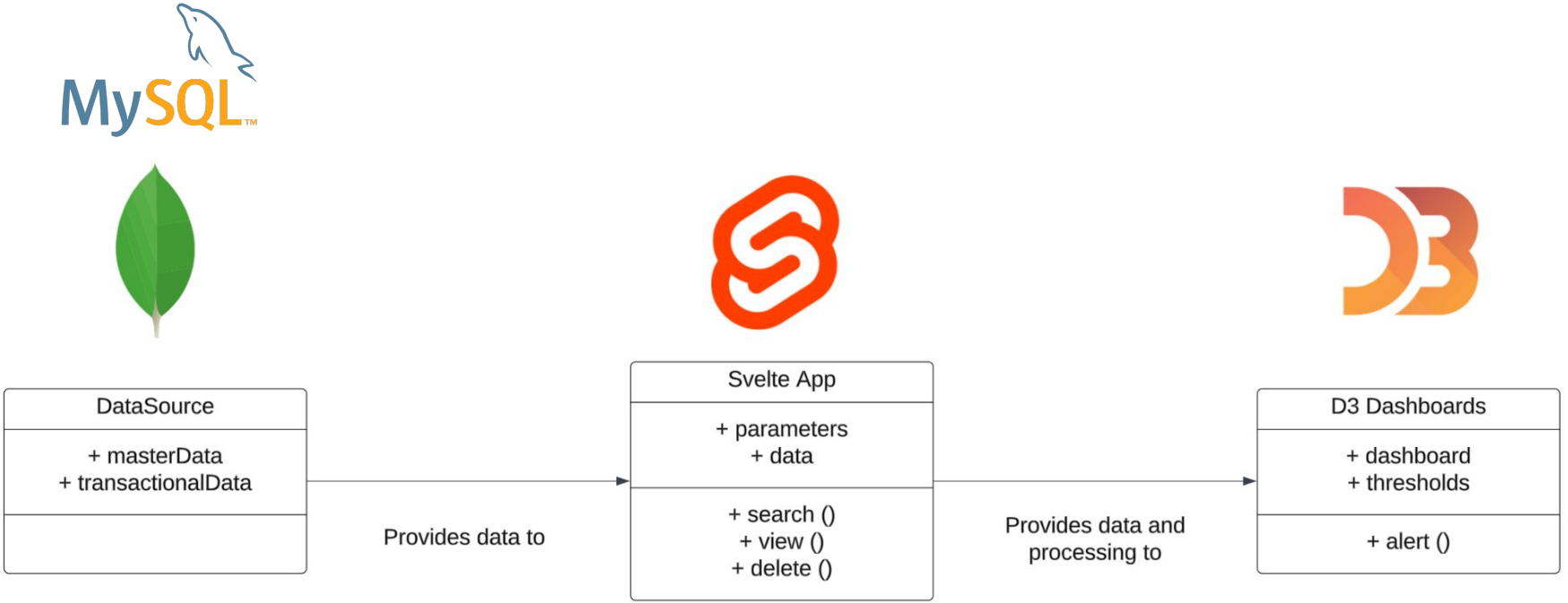
- MasterDataCollector Class:
  - Runs Linux Tools: smartctl and lsblk
    - lsblk: list connected block devices
    - smartctl: display information of connected device
  - Collects displayed information as json file
  - Uploads json file to MongoDB

- ManageMySQL Class:
  - General db functionalities (change, drop, create, show)
  - Automated creation of table from csv file to upload
  - Upload csv file data to designated table in MySQL





Query + Data Visualization Implementation



Challenges/Resolution

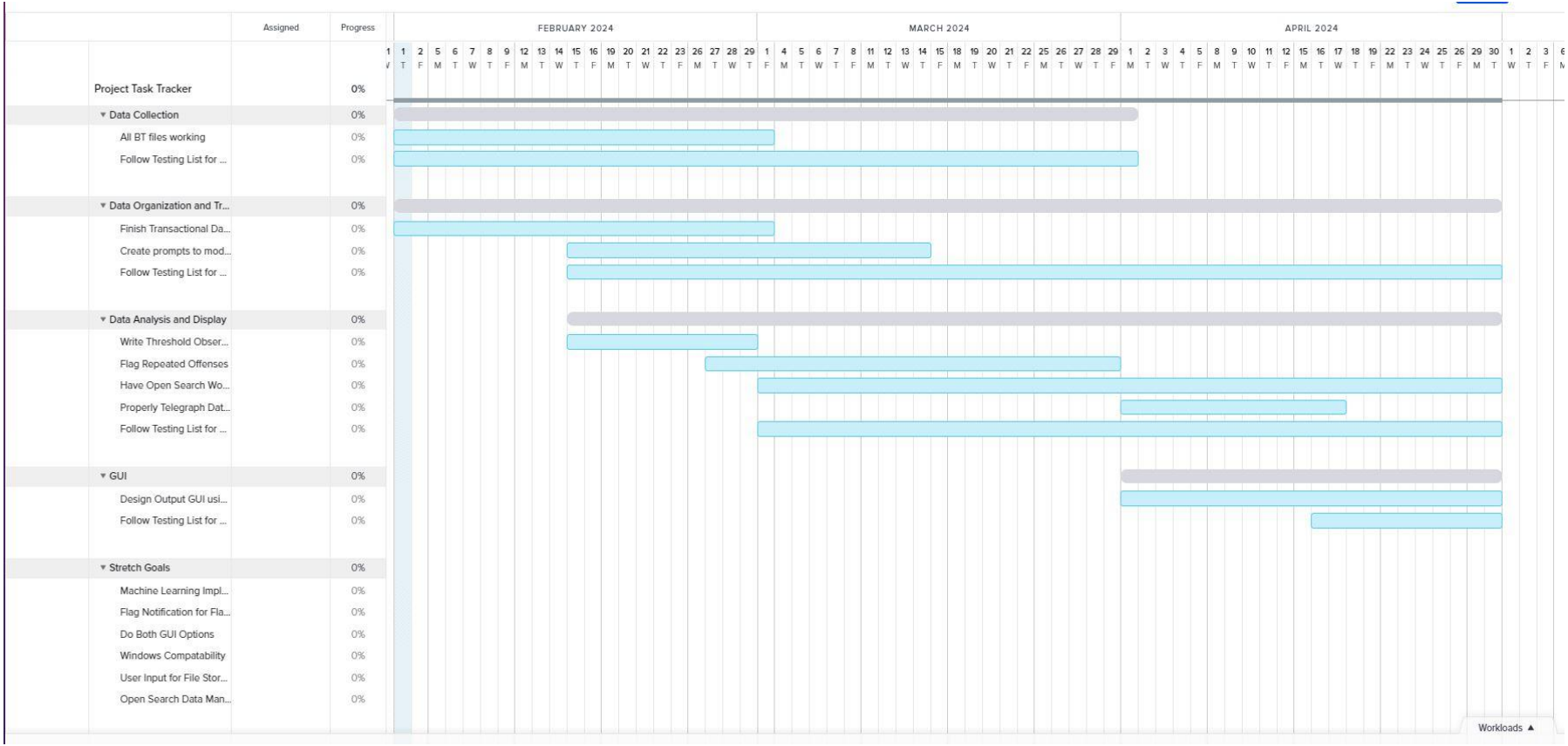
Challenges

- eBPF file Creation (NVME Latency and testing BioError)
- Low Support/Documentation for eBPF, OpenSearch, and other related tools.

Resolution

- Refer to eBPF book and videos when possible
- Research into alternative visual platforms, most likely D3.js

Schedule


Assigned
Progress
FEBRUARY 2024
MARCH 2024
APRIL 2024
Project Task Tracker 0%
Data Collection 0%
All BT files working 0%
Follow Testing List for ... 0%
Data Organization and Tr... 0%
Finish Transactional Da... 0%
Create prompts to mod... 0%
Follow Testing List for ... 0%
Data Analysis and Display 0%
Write Threshold Obser... 0%
Flag Repeated Offenses 0%
Have Open Search Wo... 0%
Properly Telegraph Dat... 0%
Follow Testing List for ... 0%
GUI 0%
Design Output GUI usi... 0%
Follow Testing List for ... 0%
Stretch Goals 0%
Machine Learning Impl... 0%
Flag Notification for Fla... 0%
Do Both GUI Options 0%
Windows Compatability 0%
User Input for File Stor... 0%
Open Search Data Man... 0%
Workloads

Conclusion

Problem:

- Silent Failures
- Long data collection process

Solution:

- An autonomous kernel level data collection system that would display all results onto a visualization platform

Architecture:

- Data Collection
- Data Sanitization
- Data Storage
- Data Query + Visualization

Thank You