Name: Romelo Gilbert
Course: CYSE 368
Term: Spring 2026
Instructor: Professor Teresa Duvall
TA: Joshua Russell
Company/Organization: Watsco Inc
Reflection #: 3

This week I worked on a mix of backend development tasks and system monitoring activities. My work included implementing features related to the CSD (Cloud Suite Distribution) post cycle count process, creating a new format for modules used in the mobile application, adding validation checks for transfer type orders, configuring monitoring and alerting using Datadog and PagerDuty and investigating an edge case in NestJS related to how large arrays are handled.

One of the tasks I worked on involved adding validation checks for transfer type orders. Specifically, I needed to ensure that a tied sales order was not locked before allowing the transfer to proceed. Without this check in place, the system could allow operations to occur in an inconsistent or unintended state, leading to reduced data integrity.

I also spent time working on formatting updates for mobile application modules and implementing logic related to the CSD post cycle count process. These tasks required a strong understanding of how data flows through the system and how different components interact. This is helping me develop a mindset where I not only focus on making features work, but also consider how they might fail or be misused.

A significant portion of this week was spent working with Datadog and PagerDuty to set up automated alerts based on application logs. I configured a Datadog analytics dashboard using APM and created multiple alert thresholds to trigger notifications based on specific conditions. For example, I set alerts for 5XX errors occurring five times within five minutes. 404 errors occurring thirty times within one minute and general 4XX errors occurring five times within thirty seconds. This exercise helped me understand how monitoring and alerting can be used to detect abnormal system behavior in real time.

This experience was particularly valuable because it introduced me to a more operational aspect of security and risk awareness. Instead of focusing only on preventing issues during development, I was able to see how systems are monitored in production to identify potential problems as they occur. High rates of 5XX errors, could indicate system failures or unhandled exceptions. While spikes in 4XX errors could point to misuse of the API or unexpected client behavior. Setting up these alerts made me realize that risk assessment is not only about identifying vulnerabilities ahead of time, but also about detecting when something is going wrong in a live environment and responding quickly.

Another interesting task I worked on involved handling an edge case in NestJS when arrays with a length greater than 20 are received in API requests. I discovered that NestJS, through the qs Javascript library, converts arrays into objects once they exceed a certain size for efficiency reasons. This behavior caused downstream issues in the API, as parts of the system were expecting an array rather than an object. Investigating this issue helped me understand how underlying libraries can introduce unexpected behavior that impacts how data is processed. It also made me more aware of how important it is to understand how frameworks handle input data. Especially when assumptions about data structure can lead to errors or inconsistencies.

Setting up alerts using Datadog and PagerDuty gave me direct exposure to how systems can be monitored in real time and how automated notifications can be used to respond to potential issues. By configuring thresholds for different types of errors, I was able to see how abnormal system behavior can be detected and escalated without manual intervention. This helped me understand that security and reliability are not only about preventing issues, but also about having the right monitoring and alerting in place to quickly identify and respond to them.