Root Cause Analysis

**Points possible:** 50

|  |  |  |
| --- | --- | --- |
| Category | Criteria | % of Grade |
| Critical Thinking | Student identifies difficulties and blockers met during the week’s project and shows an understanding of why they occurred. Student exhibits problem solving skills by following the problem to the root cause and identifying solutions to overcome the problem. | 34 |
| Citations | Student references sources used to overcome the issues outlined. | 33 |
| Organization | Thoughts are concise and clear. | 33 |

**Instructions:** Identify difficulties, errors, or mistakes made this week. Ponder and research the root cause for these issues. Write what you learned from them. Push this document to your GitHub repository for this week. Add the URL for this week’s repository to this document where instructed and submit this document to your instructor when complete.

**Difficulties, Errors, or Mistakes:**

Some difficulties I had this week was building the starting layout of my program. I found myself trying to do too much at one time. So rather working in small steps, I was trying to thing bigger picture too fast. In doing this, I found myself falling into a “rabbit hole” of errors. Along with this, I also made some simple mistakes such as typos, like the mistake that I made in the Week 2 API work.

**Root Cause:**

The root cause of my mistakes this week stemmed from me trying to do too much at one time, and not focusing on the code piece by piece. This in turn, led to me making small typos as I mentioned above, which as I have learned from in the past, become an even bigger problem the further down the line in code that you get. It essentially turns into a “where’s waldo” game of finding a small error that you made.

**What did you learn from these issues?**

I learned overall that I need to slow down and focus on making sure the steps that I am taking are done correctly and are not rushed. For example, I messed up the naming of an artifact ID in my pom.xml. This led to frustration in trying to figure out what was wrong with it, when it was a simple error. Simply slowing down and focusing on accuracy would have saved me a ton of troubleshooting time in the long run. Lesson learned on that one!

**References:**

A huge point of reference for me this week was stack overflow, when I encountered an error I would proceed to check that error and stack overflow to see what recommendations were to correct said error. I also referenced back to our previous API tutorials, even though we are doing a completely different API compared to what we did in previous weeks, I still found it helpful to reference back to the Social Media API, and Inventory Management API to guide me as I create my own.

**URL to GitHub Repository:**

<https://github.com/lukebingham/Spring-Boot-Final-Coding-Assignment.git>