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:**

Encountered many errors to begin…

Could not determine type for: java.util.Set, at table: user, for columns: [org.hibernate.mapping.Column(plans)]

Also ran into a temporal json loop which I resolved using @jsonignore on the user variable

Mistake on join column names. They were reversed.

@ManyToMany(cascade = CascadeType.ALL)

@JoinTable(name = "learning\_plan\_courses",

joinColumns = @JoinColumn(name = "courseId", referencedColumnName = "id"),

inverseJoinColumns = @JoinColumn(name = "planId", referencedColumnName = "id"))

public Set<LearningPlan> getPlan() {

return plan;

}

**Root Cause:**

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

Learned you can us @jsonignore to skip over variables.

Still wrapping my head around JPA

**References:**

**URL to GitHub Repository:**

<https://github.com/jgarcia-network/LMS>