Part One

Identify four principles associated with TDD and Unit Testing.

1. A class must include a sole responsibility. This principle will make each class very simple to test.
2. Developers should be able to add new features to a program without making any changes to the code that already exists.
3. High-level elements must never rely on low-level elements. High and low-level components should rely on abstractions.
4. Objects in the software should be replaceable with instances of their subtypes. But, this must be possible without making any changes to the software that is correct.

Answer these questions:

1. What advantages as a developer does TDD provide?
   - The code is clear and simple.
   - It improves the quality of the code.
   - It ensures that the written code is covered by at least one test, thus increasing confidence in the code.
   - Reduces time spent debugging.
2. What steps do I take to implement TDD?
   - Add a test, which most likely fails.
   - Run all the tests you currently have and see if any test fails.
   - Write only enough code to pass all the tests.
   - Run all the tests. If any test fails, go back to the last step.
   - Refactor (restructure) the code.
   - If a new test is added, repeat from the beginning.
3. What tools or software can I use to do unit testing?
   - Loadrunner, Eggplant Functional, Mocha, Jasmine, Jtest, JUnit, Mockito, Etc.

Part Two

Update your personal development folder with your reflections and status.

Part Three

Provide your meeting minutes notes and results.

- Class meetings were going over the SRS to improve it and learning about TDD.
- The first week was still meetings with the RunTime Team to make sure everything got pushed and we were up to date on our stories.
- The second week we joined new teams and are learning new responsibilities. We are also figuring out who will be the team lead for the Database team I joined.

Bi-Weekly Status Report

Name: McKennah Palmer

Project Title: Acorn Nursery

Overall Status: A little behind on the SRS Document, but Still on track with the project itself.

Summary of Project tasking
Include the following:

- **Meeting attended and minutes:** I stated this above but will restate it for the report. Class meetings were going over the SRS to improve it and learn about TDD. The first week was still meetings with the RunTime Team to make sure everything got pushed and we were up to date on our stories. The second week we joined new teams and are learning new responsibilities. We are also figuring out who will be the team lead for the Database team I joined.
- **Training notes Summary:** We will be improving the SRS, and we learned about Unit Testing and Test Driven Development.
- **Stories assigned:** Figure out the Team Lead for the Database team. Start Outline and Stubs for Nursery. See if Requirements Elicitations needs help with improving the SRS.

- ➔ **Project artifacts and notes identified in project tasking:** Get a Team Lead and learn the Stories and Responsibilities of the Database Team.

Include a self-evaluation of the rubric for the project development's rubric

Number of Hours

- ➔ **Number of Hours worked this week (expectation: 7-9 hours):** 8.5 hours the first week and 7 hours the second week.
- ➔ **Total Number of Hours this semester (anticipated 120/semester):** 40.5 hours so far this semester.

Accomplishments (since the last status report), include completed stories.

- ➔ **Number of Story Points completed:** 4
- ➔ **Number of Stories completed:** 4
- ➔ **Personal Velocity (Points/Stories):** 1
- ➔ **Project Velocity (Teams Points/Teams Stories):** 4
- ➔ **Earned Value (Actual Points/Hour - Estimated Points/Hour):** 4/15.5

—> I might need the above section explained to me, or I am up too late working on this again because I am not sure I understand story points even though I do understand stories.

- ➔ **Performance**
  - ◆ **Last Week's Performance Average (Actual Total Points/ Total Hours):** 3/15
  - ◆ **This Week's Performance (Actual Points/Hours):** 4/15.5
- ➔ **Challenges encountered and resolutions found (since the last status report):** One challenge for the whole class was the SRS document which we ended up needing extra time for the whole document. Another challenge was splitting us into new teams evenly, it seems like a lot of us like Backend work. Other than the class challenges, my teams seem to be doing fine.
- ➔ **Plans/Goals/Tasking for the coming week/sprint.**
  - ◆ Find and start on the Stories for the Database Teams.
  - ◆ Create a basic outline of Acorn Nursery in GitHub.
  - ◆ Create Stubs.
  - ◆ **Follow-up/Reporting**
    - No other Comments.