Elijah Saavedra

CST-350

Activity 6

**Part 1: The Debugger**

**Screenshot 1: In this screenshot, you can see the user variable**

**A screenshot of a computer program

Description automatically generated**

**Screenshot 2: In this screenshot, we are stepping into a function**

**A screenshot of a computer

Description automatically generated**

**Screenshot 3: In this screenshot, we are inspecting the properties**

**A screenshot of a computer program

Description automatically generated**

**Screenshot 4: In this screenshot, we can see if the user was found or not, here success it true**

**A screenshot of a computer

Description automatically generated**

**Screenshot 5: In this screenshot, we have a conditional breakpoint**

**A screenshot of a computer

Description automatically generated**

**Screenshot 6: In this screenshot, we have a debugger messageA screenshot of a computer screen

Description automatically generated**

**Screenshot 7: In this screenshot, we can see the Call Stack**

**A screenshot of a computer

Description automatically generated**

In this part of the activity, we used the debugger to set a breakpoint and examine the state of the variables during the breakpoint. We learned how to step over, into, and out of the code to see a variety of ways to debug our application. Along with this, we also learned how to set a conditional breakpoint, log messages with a break point, and examine a call stack of the application.

**Part 2: NLog**

**Screenshot 1: In this screenshot, we can see the text log file with the information using the Logging Framework.**

**A screenshot of a computer

Description automatically generated**

**Screenshot 2: In this screenshot, we can see the log file implementing the Singleton Design Pattern**

**A screenshot of a computer

Description automatically generated**

In this part of the activity, we learned how to use nLogs, installing the nLog Framework. We also learned about Singleton Pattern which allowed us to create a single instance of an object. NLog is important as it provides a flexible logging framework, which allows users to get runtime information easier, troubleshoot issues, and monitor application health.