ASSIGNMENT

DOCTOR PRESCRIPTION CONNECTING WITH DB AND UNIT TESING WITH LOG4NET

Program.cs

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Week4AssessmentApp;
using log4net;


namespace Week4AssessmentApp
{

    public class ServerException : Exception
    {
        public ServerException(string message) : base(message) { }
    }
    public class DoctorPrescription
    {
        public int DoctorID { get; set; }
        public string PatientName { get; set; }
        public string Medication { get; set; }
        public double Dosage { get; set; }

        public DoctorPrescription(int doctorID, string patientName, string medication, double dosage)
        {
            DoctorID = doctorID;
            PatientName = patientName;
            Medication = medication;
            Dosage = dosage;
        }



        public override string ToString()
        {
            return $"DoctorID
:{DoctorID},PatientName:{PatientName},Medication:{Medication},Dosage:{Dosage}";
        }
    }
    public class DoctorPrescriptionService
    {
        private static string connectionString = "Data Source=(localdb)\\MSSQLLocalDB;Initial
Catalog=Week4AssessmentDb;Integrated Security=True;";
        public static void Read(DoctorPrescription[] doctorPres)
        {

            /* for (int i = 0; i < doctorPres.Length; i++)
             {
                Console.Write("Enter the DoctorID:");
                int DoctorID = int.Parse(Console.ReadLine());
                Console.Write("Enter the PatientName:");
                string PatientName = Console.ReadLine();
                Console.Write("Enter the Medication:");
                string Medication = Console.ReadLine();
                Console.Write("Enter the Dosage:");
                double Dosage = double.Parse(Console.ReadLine());
            doctorPres[i] = new DoctorPrescription(DoctorID, PatientName, Medication, Dosage);
          }*/

            try
            {
                using (SqlConnection conn = new SqlConnection(connectionString))
                {
                    string query = "SELECT DoctorID, PatientName, Medication, Dosage FROM
DoctorPrescription";
                    SqlCommand cmd = new SqlCommand(query, conn);

                    conn.Open();
                    SqlDataReader reader = cmd.ExecuteReader();



                    for (int i = 0; i < doctorPres.Length; i++)
                    {
                        if (!reader.Read())
                        {
                            throw new ServerException("[0101]Server Error.");//throw error
                        }
                        doctorPres[i] = new DoctorPrescription(
                        (int)reader["DoctorID"],
                        (string)reader["PatientName"],
                        (string)reader["Medication"],
                        (double)reader["Dosage"]
                    );

                }
                }
            }
```

```
    catch (SqlException ex)

    {

        // Handle SQL exceptions

        //Console.WriteLine($"SQL Error: {ex.Message}");

        throw new ServerException($"[0102]Server Error.{ex.Message}");//throw Error

    }

    catch (ServerException ex)

    {

        throw ex;

    }

    catch (Exception ex)

    {

        // Handle other exceptions

        //Console.WriteLine($"Error: {ex.Message}");

        throw new ServerException($"[0103]Server Error.{ex.Message}");//throw Error

    }


}

public static void Sort(DoctorPrescription[] doctorPres)

{

    int n = doctorPres.Length;

    for (int i = 0; i < n - 1; i++)

    {

        int minIndex = i;

        for (int j = i + 1; j < n; j++)

        {

            if (string.Compare(doctorPres[j].PatientName, doctorPres[minIndex].PatientName) < 0)

            {

                minIndex = j;

            }
            }
            if (minIndex != i)
            {
                DoctorPrescription temp = doctorPres[minIndex];
                doctorPres[minIndex] = doctorPres[i];
                doctorPres[i] = temp;
            }
        }

    }
    public static DoctorPrescription DisplayDocLeastDos(DoctorPrescription[] doctorPres, int
doctorID)
    {
        double minDosage = double.MaxValue;
        DoctorPrescription minPres = null;

        foreach (DoctorPrescription dosage in doctorPres)
        {
            if (dosage.DoctorID == doctorID && dosage.Dosage < minDosage)
            {
                minDosage = dosage.Dosage;
                minPres = dosage;
            }
        }



        Console.WriteLine($"Minimum dosage prescribed by DoctorID {doctorID}:{minPres.Dosage}");
            return minPres;



    }
    public static DoctorPrescription DisplayThirdMaxDosage(DoctorPrescription[] doctorPres)
    {
        for (int i = 0; i < doctorPres.Length; i++)
        {
            int min = i;
            for (int j = i + 1; j < doctorPres.Length; j++)
            {
                if (doctorPres[j].Dosage < doctorPres[min].Dosage)
                {
                    min = j;

                }
            }
            if (i != min)
            {
                DoctorPrescription temp = doctorPres[min];
                doctorPres[min] = doctorPres[i];
                doctorPres[i] = temp;
            }
        }
        Console.WriteLine($"Third Max Dosage is:{doctorPres[doctorPres.Length - 2]}");
        return doctorPres[doctorPres.Length - 2];
    }


}
public class Program
{
    private static readonly ILog log = LogManager.GetLogger(typeof(Program));
    static void Main(string[] args)
    {
            DoctorPrescription[] doctorPres = new DoctorPrescription[3];
            try
            {
              DoctorPrescriptionService.Read(doctorPres);
            }
            catch (ServerException ex)
            {
                log.Error($"{ex.Message}");
            }
            DoctorPrescriptionService.Read(doctorPres);
            log.Info("enter the doctor id to find the minimum dosage:");
            int doctorID = int.Parse(Console.ReadLine());
            DoctorPrescriptionService.DisplayDocLeastDos(doctorPres, doctorID);
            DoctorPrescriptionService.DisplayThirdMaxDosage(doctorPres);
            DoctorPrescriptionService.Sort(doctorPres);
            log.Info("sorted by patient name:");
            foreach (DoctorPrescription patient in doctorPres)
            {
                log.Info(patient);
            }
        }
    }
}
```

SQL QUERY

```
CREATE DATABASE Week4AssessmentDb;
USE Week4AssessmentDb;
CREATE TABLE DoctorPrescription(
DoctorID INT PRIMARY KEY,
PatientName VARCHAR(225),
Medication NVARCHAR(100),

Dosage FLOAT

);

INSERT INTO  DoctorPrescription(DoctorID,PatientName,Medication,Dosage)VALUES

(1,'Fidha','Dolo 650',2),(2,'Sarika','Para',1),(3,'Athuliya','Vicks',3);
```

AssemblyInfo.cs

```
[assembly: log4net.Config.XmlConfigurator]
```

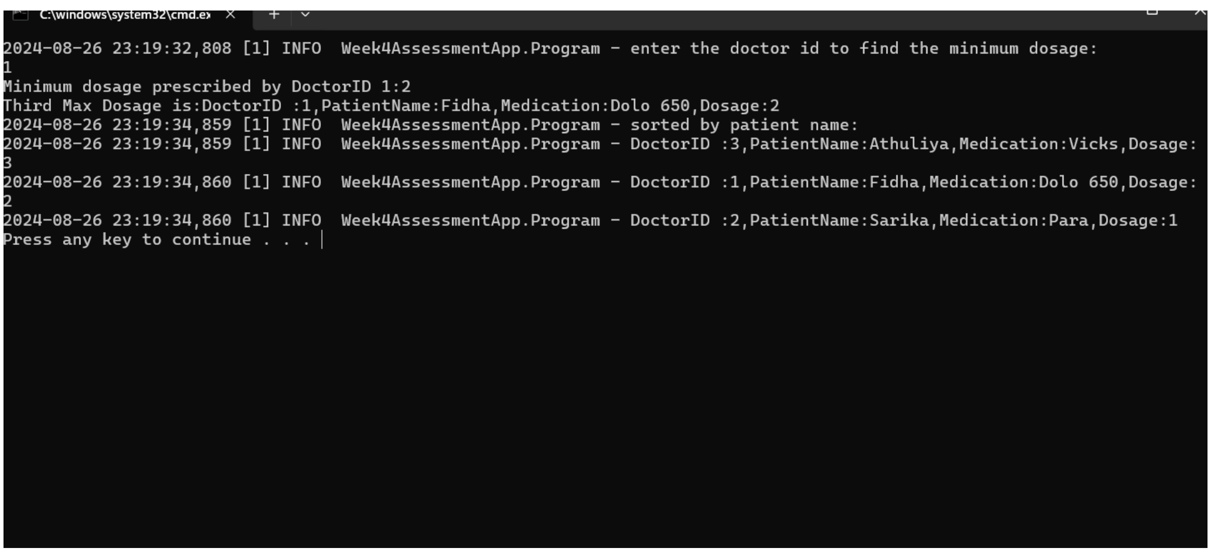
App.config

```
<?xml version="1.0" encoding="utf-8" ?>

<configuration>

	<configSections>

		<section name="log4net"
type="log4net.Config.Log4NetConfigurationSectionHandler, log4net" />

	</configSections>


	<log4net>

		<!-- File Appender -->

		<appender name="FileAppender" type="log4net.Appender.RollingFileAppender">

			<file value="week4assessment_app_log.log" />

			<appendToFile value="true" />

			<rollingStyle value="Size" />

			<maxSizeRollBackups value="5" />

			<maximumFileSize value="10MB" />

			<staticLogFileName value="true" />

			<layout type="log4net.Layout.PatternLayout">

				<conversionPattern value="%date [%thread] %-5level %logger -
%message%newline" />

			</layout>
```

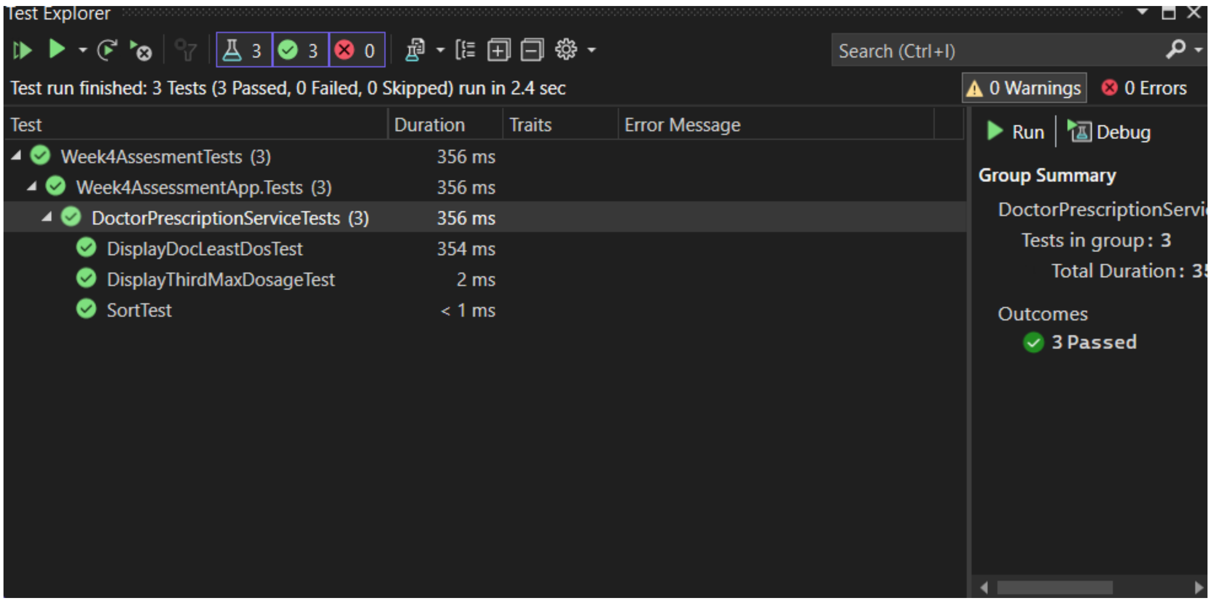
```
				</appender>


				<!-- Console Appender -->
				<appender name="ConsoleAppender" type="log4net.Appender.ConsoleAppender">
					<layout type="log4net.Layout.PatternLayout">
						<conversionPattern value="%date [%thread] %-5level %logger -
%message%newline" />
					</layout>
				</appender>


				<!-- Root logger -->
				<root>
					<level value="ALL" />
					<appender-ref ref="FileAppender" />
					<appender-ref ref="ConsoleAppender" />
				</root>
			</log4net>
			<startup>
		<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
	</startup>
</configuration>
```

DoctorPrescriptionServicesTest.cs

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Week4AssessmentApp;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;


namespace Week4AssessmentApp.Tests
{
    [TestClass()]
    public class DoctorPrescriptionServiceTests
    {
        [TestMethod()]
        public void DisplayDocLeastDosTest()
        {
            DoctorPrescription[] doctorPres = new DoctorPrescription[3];
            DoctorPrescriptionService.Read(doctorPres);
            int doctorID = 1; // Assuming you want to test for DoctorID 1
            DoctorPrescription expected = new DoctorPrescription(1, "Fidha", "Dolo 650", 2);
            DoctorPrescription actual = DoctorPrescriptionService.DisplayDocLeastDos(doctorPres,
doctorID);
            Assert.AreEqual(expected.ToString(), actual.ToString());
        }



        [TestMethod()]
        public void DisplayThirdMaxDosageTest()
        {
            DoctorPrescription[] doctorPres = new DoctorPrescription[3];
            DoctorPrescriptionService.Read(doctorPres);
            DoctorPrescription expected = new DoctorPrescription(1, "Fidha", "Dolo 650", 2);
            DoctorPrescription actual = DoctorPrescriptionService.DisplayThirdMaxDosage(doctorPres);
            Assert.AreEqual(expected.ToString(), actual.ToString());
        }
        [TestMethod()]
        public void SortTest()
        {
            DoctorPrescription[] doctorPres = new DoctorPrescription[3];
            DoctorPrescriptionService.Read(doctorPres);
            DoctorPrescription expected = new DoctorPrescription(3, "Athuliya", "Vicks", 3);
            DoctorPrescriptionService.Sort(doctorPres);
            DoctorPrescription actual = doctorPres[0];
            Assert.AreEqual(expected.ToString(), actual.ToString());
        }
    }
}
```



```
C:\windows\system32\cmd.ex
2024-08-26 23:19:32,808 [1] INFO  Week4AssessmentApp.Program - enter the doctor id to find the minimum dosage:
1
Minimum dosage prescribed by DoctorID 1:2
Third Max Dosage is:DoctorID :1,PatientName:Fidha,Medication:Dolo 650,Dosage:2
2024-08-26 23:19:34,859 [1] INFO  Week4AssessmentApp.Program - sorted by patient name:
2024-08-26 23:19:34,859 [1] INFO  Week4AssessmentApp.Program - DoctorID :3,PatientName:Athuliya,Medication:Vicks,Dosage:
3
2024-08-26 23:19:34,860 [1] INFO  Week4AssessmentApp.Program - DoctorID :1,PatientName:Fidha,Medication:Dolo 650,Dosage:
2
2024-08-26 23:19:34,860 [1] INFO  Week4AssessmentApp.Program - DoctorID :2,PatientName:Sarika,Medication:Para,Dosage:1
Press any key to continue . . .
```



Test Explorer

Test run finished: 3 Tests (3 Passed, 0 Failed, 0 Skipped) run in 2.4 sec

0 Warnings   0 Errors

Test	Duration	Traits	Error Message
Week4AssesmentTests (3)	356 ms		
Week4AssessmentApp.Tests (3)	356 ms		
DoctorPrescriptionServiceTests (3)	356 ms		
DisplayDocLeastDosTest	354 ms		
DisplayThirdMaxDosageTest	2 ms		
SortTest	< 1 ms		

Group Summary

DoctorPrescriptionServi

Tests in group: 3

Total Duration: 3

Outcomes

3 Passed