



פרויקט גמר

למילוי חלקי של הדרישות לקבלת תואר הנדסאי

הנדסת תוכנה

בהתמחות: מחשבים

נושא הפרויקט: ספריה

שם הסטודנטית: מיכל שחטר

מספר זהות: 322613951

העבודה בוצעה בהנחיית : נחמה היימן





תוכן עניינים

3.....	הקדמה:
4.....	צילומי מסך:
7.....	תרשים UML
8.....	פירוט מחלקות
8.....	:Server Side
10.....	Client Side
11.....	קטעי קוד
11.....	קטעי קוד בצורת שרת
14.....	קטעי קוד בצד לקוח



הקדמה:

בפרויקט המעבדה מוצג הקשר בין צד שרת לצד לקוח, בפרויקט שלי אעסוק בהשאלת ספרים מספריה,

בפרויקט זה, התוכנה מדמה תוכנת ספרניות בספריה, כלומר מאפשרת להשאיל ספר מאחד הספרים שיש בספריה ולהחזיר את הספר לספריה.

אם הספר תפוס ע"י משאיל אחר תוצג הודעה- "הספר בהשאלה" ורק לאחר שהספר יוחזר לספריה יוכלו להשאיל אותו.

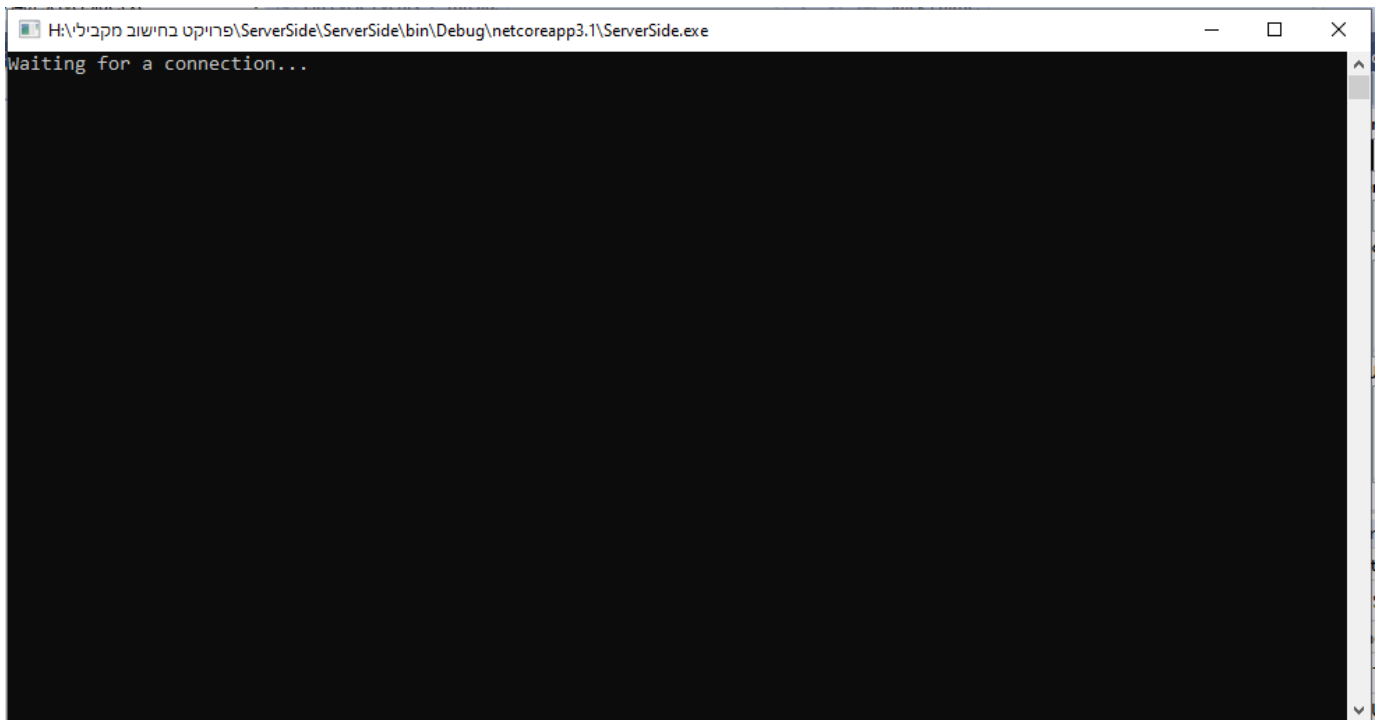
בצד השרת נשמרים הנתונים על הספרים ובצד הלקוח אפשר לבחור ספר להשאלה או להחזיר ספר מושאל.

בפרויקט זה השתמשתי ב-threads על מנת ליעל את המערכת בכך ששני שואלים יוכלו להשאיל ספרים בו זמנית, אך בשביל שלא ישאלו את אותו ספר ביחד השתמשתי גם במנגנון נעילה.



צילומי מסך:

ראשית מריצים את צד השרת,
כך יראה החלון אחרי ההרצה



כעת השרת מחכה שיפנו אליו שואלים כדי לשאול ספר מהספריה.

לאחר מכן , נריץ את ממשק המשתמש , בו המשתמש יכול להשאל ספר מהספרים הקיימים
בספריה-



המשתמש יבחר את הספר שברצונו להשאיל, לאחר הלחיצה יוצג למשתמש הודעה. אם הספר בהשאלה אצל משתמש אחר תוצג לו הודעה "הספר בהשאלה", אם הספר פנוי תוצג

Form1

ספריה

החזר ספר

ספר 4	ספר 1
ספר 5	ספר 2
ספר 6	ספר 3

בחר ספר להשאלה

ספר 1
ספר 2
ספר 3
ספר 6

×

הספר בידך, קריאה נעימה

אישור

Form1

ספריה

החזר ספר

ספר 4	ספר 1
ספר 5	ספר 2
ספר 6	ספר 3

בחר ספר להשאלה

ספר 1
ספר 2
ספר 3
ספר 6

×

הספר בהשאלה

אישור



כאשר המשתמש ירצה להחזיר את הספר לספריה הוא יחלץ על הספר שברצונו להחזיר שתחת הכותרת -החזר ספר ותוצג לו הודעה "תודה, מקווים שנהנת" רק לאחר שיוחזר הספר לספריה משתמש נוסף יוכל לקחת את הספר.

Form1

ספריה

החזר ספר

ספר 4

ספר 1

ספר 5

ספר 2

ספר 6

ספר 3

בחר ספר להשאלה

ספר 1

ספר 2

ספר 3

ספר 6

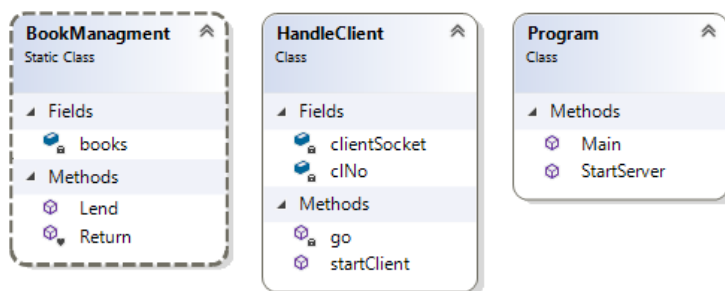
×

תודה, מקווים שנהנת

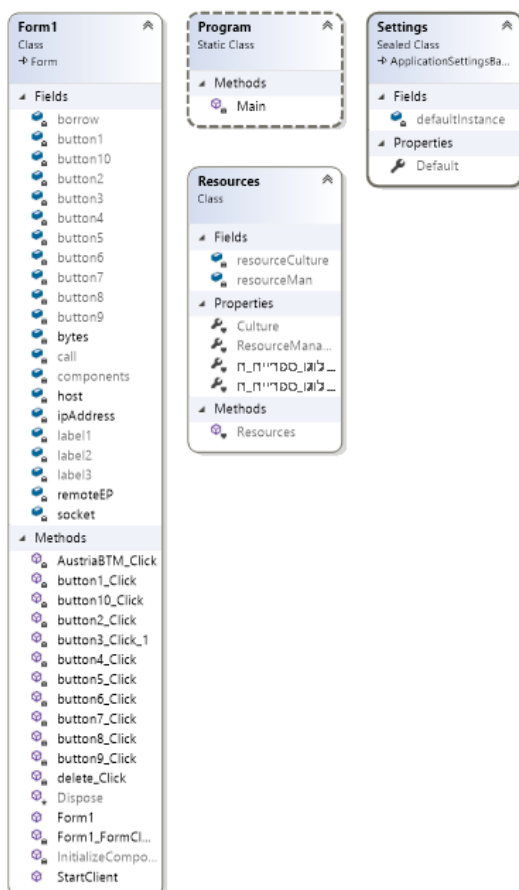


תרשים UML Class Diagram

server



Client





פירוט מחלקות

:Server Side

מחלקת Program-

פונקציות המחלקה-

פונקציות	פירוט
<code>public static int Main(String[] args)</code>	הפונקציה הראשית
<code>public static void StartServer()</code>	פונקציה המפעילה את צד השרת

מחלקת HandleClient-

מחלקה שמקבלת את מה שה client שלח לה וחותכת אותו ע"י Split ל 2 חלקים, החלק הראשון מציין האם הספר להשאלה או החזרה והחלק השני מציין את מספר הספר שרוצים לשאול/להחזיר וכאשר החלק הראשון הוא 1 ז"א שרוצים להשאיל את הספר ואז שולחים לפונקציה Lend שנמצאת במחלקת BookManagment ושם בודקים האם הספר נמצא בהשאלה אצל משתמש אחר או שהוא נמצא בספריה ואפשר להשאיל אותו, אם הוא 0 ז"א שרוצים להחזיר את הספר ושולחים לפונקציה Return ושם מציבים במערך במקום של הספר 0 ושולחים הודעה לclient.

משתני המחלקה-

מאפיינים	פירוט
Socket clientSocket	הסוקט שנוצר כאשר מתקבלת קריאה מצד לקוח
string clineNo	מספר מזהה של הסוקט

פונקציות המחלקה-

פונקציות	פירוט
<code>public void startClient(Socket inClientSocket, string clineNo)</code>	פותחת סוקט עבור Client חדש שמתחבר לשרת ומפעילה thread חדש
<code>private void go()</code>	מפעילה את הסוקט



מחלקת BookManagment

מחלקה המיצגת מערך של הספרים הנמצאים בספריה, כאשר הספר נמצא בספריה הוא מיוצג במערך-0 וכאשר הוא מושאל מוצג 0 במערך.

משתני המחלקה:

<u>שם:</u>	<u>סוג:</u>	<u>פרוט:</u>
Books	<u>Int[]</u>	מערך בגודל מספר הספרים שיש בספריה

פונקציות המחלקה-

<u>פונקציה:</u>	<u>פירוט:</u>
<code>public static void Lend(Socket clientSocket, int numBook)</code>	הפונקציה מקבלת את מספר הספר ובודקת האם במערך במקום של ספר זה יש 1 או 0, אם יש 1-זה אומר שהספר בהשאלה והיא מדפיסה הודעה למשתמש "הספר בהשאלה" אם יש במערך 0 היא מעדכנת במערך ל1 ומדפיסה הודעה למשתמש "הספר בידך, קריאה נעימה"
<code>internal static void Return(Socket clientSocket, int numBook)</code>	הפונקציה מעדכנת במערך במקום של מספר הספר ל 0 ומדפיסה הודעה ללקוח "תודה רבה, מקווים שנהנת"



Client Side

ספריה

מחלקת Program-

פונקציות	פירוט
<code>static void Main()</code>	הפונקציה הראשית

מחלקת Form1-

מאפיינים	פירוט
<code>IPHostEntry host</code>	חיבור לשרת מרוחק
<code>IPAddress ipAddress</code>	כתובת ה IP של השרת
<code>IPEndPoint remoteEP</code>	חיבור בין השרת והלקוח בשימוש ב Port
<code>Socket socket</code>	סוקט של הלקוח
<code>byte[] bytes = new byte[1024]</code>	מערך עזר של בתים לקבלת ושליחת הודעות מקודדות

פונקציות	פירוט
<code>public void StartClient()</code>	פותחת חיבור בין הלקוח לשרת
<code>public Form1()</code>	בנאי המאתחל את ה Form
<code>private void button1_Click(object sender, EventArgs e)</code>	בלחיצה על הכפתור תשלח הודעה לשרת האם הספר הוא להשאלה או החזרה ואת מספר הספר, כך בלחיצה על כל כפתורי ההשאלה. בלחיצה על השאלה ישלח לשרת המספר 1 ומספר הספר וכאשר הספר הוא להחזרה ישלח לשרת המספר 0 ומספר הספר
<code>private void Form1_FormClosing(object sender, FormClosingEventArgs e)</code>	בסגירת הטופס הפונקציה סוגרת את הסוקט



קטעי קוד קטעי קוד בצורת שרת מחלקת Program

```
class Program
{

    public static int Main(String[] args)
    {
        StartServer();
        return 0;
    }

    public static void StartServer()
    {
        // Get Host IP Address that is used to establish a connection
        // In this case, we get one IP address of localhost that is IP :
        127.0.0.1
        // If a host has multiple addresses, you will get a list of
        addresses
        int counter = 0;

        IPEndPoint host = Dns.GetHostEntry("localhost");
        IPAddress ipAddress = host.AddressList[0];
        IPEndPoint localEndPoint = new IPEndPoint(ipAddress, 3000);
        // Create a Socket that will use Tcp protocol

        Socket listener = new Socket(ipAddress.AddressFamily,
        SocketType.Stream, ProtocolType.Tcp);
        // A Socket must be associated with an endpoint using the Bind
        method
        listener.Bind(localEndPoint);
        // Specify how many requests a Socket can listen before it gives
        Server busy response.
        // We will listen 10 requests at a time
        listener.Listen(10);

        while (true)
        {
            try
            {

                Console.WriteLine("Waiting for a connection...");
                Socket handler = listener.Accept();
                counter++;
                HandleClient client = new HandleClient();
                client.startClient(handler, Convert.ToString(counter));

            }
            catch (Exception e)
            {
            }
        }
    }
}
```



```

        Console.WriteLine(e.ToString());
    }

    //Console.WriteLine("\n Press any key to continue...");
    //Console.ReadKey();
}

}

}

```

מחלקת HandeleClient

```

class HandleClient
{
    Socket clientSocket;
    string clNo;
    public void startClient(Socket inClientSocket, string clineNo)
    {
        this.clientSocket = inClientSocket;
        this.clNo = clineNo;
        Thread ctThread = new Thread(go);
        ctThread.Start();
    }
    private void go()
    {
        // Incoming data from the client.
        string data = null;
        byte[] bytes = null;
        bytes = new byte[1024];

        try
        {
            while (true)
            {
                bytes.Clone();
                int bytesRec = clientSocket.Receive(bytes);
                data = Encoding.UTF8.GetString(bytes, 0, bytesRec);
                string[] str= data.Split(',');
                if (data.IndexOf("<EOF>") > -1)
                {
                    break;
                }

                switch (str[0])
                {
                    //חזרה או השאלה האם בודקת
                    case "1":
                        //השאלה אם
                        switch (str[1])
                        {
                            //הספר מספר את בודקת
                            case "1":
                                BookManagment.Lend(clientSocket, 1);
                                break;
                            case "2":
                                BookManagment.Lend(clientSocket, 2);
                                break;
                            case "3":
                                BookManagment.Lend(clientSocket, 3);
                                break;
                        }
                    }
                }
            }
        }
    }
}

```



```

case "4":
    BookManagment.Lend(clientSocket, 4);
    break;
case "5":
    BookManagment.Lend(clientSocket, 5);
    break;
case "6":
    BookManagment.Lend(clientSocket, 6);
    break;
}
break;
case "0":
    //ספר של החזרה אם
    BookManagment.Return(clientSocket, Convert.ToInt32(
str[1]));
    break;
}
}
}
catch (Exception ex)
{
    Console.WriteLine(" >> " + ex.ToString());
}
finally
{
    clientSocket.Shutdown(SocketShutdown.Both);
    clientSocket.Close();
}
}
}

```



קטעי קוד בצד לקוח מחלקת Program

```
static class Program
{
    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new Form1());
    }
}
```

מחלקת Form1

```
public partial class Form1 : Form
{
    IPEndPoint remoteEP;
    Socket socket;
    //בסוקט הנתונים את ששולח אסקי משתנה
    byte[] bytes = new byte[1024];

    public void StartClient()
    {
        try
        {
            // Connect to a Remote server
            // Get Host IP Address that is used to establish a connection
            // In this case, we get one IP address of localhost that is IP
: 127.0.0.1
            // If a host has multiple addresses, you will get a list of
addresses
            host = Dns.GetHostEntry("localhost");
            ipAddress = host.AddressList[0];
            remoteEP = new IPEndPoint(ipAddress, 3000);

            // Create a TCP/IP socket.
            socket = new Socket(ipAddress.AddressFamily,
                SocketType.Stream, ProtocolType.Tcp);
        }
    }
}
```



```
// Connect the socket to the remote endpoint. Catch any errors.
try
{
    // Connect to Remote EndPoint
    socket.Connect(remoteEP);

    Console.WriteLine("Socket connected to {0}",
        socket.RemoteEndPoint.ToString());

}
catch (ArgumentNullException ane)
{
    Console.WriteLine("ArgumentNullException : {0}",
ane.ToString());
}
catch (SocketException se)
{
    Console.WriteLine("SocketException : {0}", se.ToString());
}
catch (Exception e)
{
    Console.WriteLine("Unexpected exception : {0}",
e.ToString());
}

}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
}

public Form1()
{
    InitializeComponent();
    StartClient();
}

private void delete_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("1,2");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

    // Displays the MessageBox.
    MessageBox.Show(s);
}

private void AustriaBTM_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("1,1");
```



```
// Send the data through the socket.
int bytesSent = socket.Send(msg);

// Receive the response from the remote device.
int bytesRec = socket.Receive(bytes);
String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

// Displays the MessageBox.
MessageBox.Show(s);
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    // socket.Shutdown(SocketShutdown.Both);
    // socket.Close();
}

private void button6_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("0,2");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

    // Displays the MessageBox.
    MessageBox.Show(s);
}

private void button5_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("0,1");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

    // Displays the MessageBox.
    MessageBox.Show(s);
}

private void button7_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("0,3");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);
}
```




```
// Displays the MessageBox.
MessageBox.Show(s);
}

private void button8_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("0,4");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

    // Displays the MessageBox.
    MessageBox.Show(s);
}

private void button9_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("0,5");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

    // Displays the MessageBox.
    MessageBox.Show(s);
}

private void button10_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("0,6");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

    // Displays the MessageBox.
    MessageBox.Show(s);
}

private void button1_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("1,3");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
}
```



322613951 מיכל שחטר

```
String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

// Displays the MessageBox.
MessageBox.Show(s);
}

private void button2_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("1,4");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

    // Displays the MessageBox.
    MessageBox.Show(s);
}

private void button3_Click_1(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("1,5");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

    // Displays the MessageBox.
    MessageBox.Show(s);
}

private void button4_Click(object sender, EventArgs e)
{
    byte[] msg = Encoding.UTF8.GetBytes("1,6");

    // Send the data through the socket.
    int bytesSent = socket.Send(msg);

    // Receive the response from the remote device.
    int bytesRec = socket.Receive(bytes);
    String s = Encoding.UTF8.GetString(bytes, 0, bytesRec);

    // Displays the MessageBox.
    MessageBox.Show(s);
}
}
```