

תיק פרויקט

המורה: יהודה אבני

מגיש: אופק כנרי

ת.ז: 212140008

בית ספר: שמעון בן צבי

כיתה: י"ב 2

תאריך: 16.05.2019

תוכן עניינים

תוכן

3.....	רקע לפרויקט
4.....	תחומים
4.....	מטרות הפרויקט
6.....	שפות התכנות וסביבות העבודה
7.....	פירוט הבעיה האלגוריתמית וניתוחה
9.....	פתרונות - יתרונות וחסרונות
9.....	1. ביצוע תקשורת עם מספר לקוחות בו-זמנית
9.....	2. הצפנת המידע נגד צד שלישי
10.....	3. יצירת CAPTCHA - בדיקת אנושיות - בצד שרת ולהעביר אותה ללקוח
11.....	4. זיהוי הבקשות של הלקוח ותגובות השרת
11.....	5. שליחת דואר אלקטרוני
12.....	הפתרון שנבחר
20.....	דרישות ומגבלות המערכת
21.....	ממשק משתמש
22.....	תרשים זרימה למחלקות השרת
23.....	חלונות הפרוייקט
27.....	פעולות בפרוייקט
27.....	טבלת פעולות – לקוח
42.....	טבלת פעולות - שרת

רקע לפרויקט

הפרויקט שלי הוא תוכנה המאפשרת יצירת קשר בין שני מחשבים ברשת מקומית כך שלכל משתמש קיימת אופציה לשלוח בקשה לשחק עם משתמש שני, במידה והמשתמש השני הסכים לבקשה לשני המשתמשים יפתח חלון משותף ובו לוח המשחק וצ'אט. אותה התוכנה עולה בשני המחשבים על ידי המשתמש והסנכרון בין שני המחשבים נעשה על ידי הודעות טקסט בלבד, האבטחה של התוכנה באה לידי ביטוי בכך שכל התקשורת בין המשתמשים מוצפנת בעזרת מפתחות הצפנה.

התוכנה בנויה משני צדדים: צד שרת וצד לקוח. התקשורת נעשית על ידי התחברות של כל אחד מהלקוחות לשרת אחד קבוע שרץ ללא הפסקה.

בחרתי לעשות את הפרויקט על נושא זה מכמה סיבות:

נושא זה של תקשורת בין שני משתמשים במשחק אינטרנטי עניין אותי רבות מכיוון שאני משחק בעצמי במשחקי מחשב הדורשים חיבור לאינטרנט.

משתמשים פוטנציאליים:

- צעירים
- חובבי המשחק צוללות

התוכנה שלי מורכבת מלקוחות שכוללים בתוכם מספר טפסים ושרת:

מספר משתמשים וסרבר אחד, המשתמשים מתקשרים עם השרת והוא אחראי לשליחת ההודעות אל המשתמש השני. המשתמשים מתקשרים אחד עם השני באמצעות הודעות טקסט בלבד, כל שינוי בחלון האפליקציה גם הוא עובר באמצעות הודעת טקסט.

תחומים

אבטחת מידע

תקשורת בין מחשבים

בסיס נתונים

מטרות הפרויקט

המוצר המוגמר אמור לבצע את הדברים הבאים:

המוצר המוגמר אמור לתפקד בתור תוכנה המאפשרת משחק צוללות וצ'אט מאובטח ונוח בין שני משתמשים.

מטרות מרכזיות:

לאפשר יצירת קשר בין משתמשים לשרת ואז לזהותם (בעזרת הרשמה והתחברות) ולאחר מכן יצירת מרחב למשחק זהה בין שני המשתמשים, כל משתמש ישחק בתורו. כל שינוי שייעשה יעשה באחריות המשתמשים וירשם בקובץ טקסט שישמר במחשב השרת, קצב העברת הנתונים יהיה מהיר, ללא עיכובים, אחרת עלול להשתבש עיבוד ההודעות.

בנוסף לכך, הקשר בין המשתמשים לכל אורך הדרך יהיה מוצפן ומאובטח כך שגם אם צד שלישי מאזין לקשר, הוא לא יידע לפענח את ההודעות העוברות.

אבטחת החשבונות תתבצע באמצעות דואר אלקטרוני אשר מקושר לחשבונות של המשתמש ובעזרתו הם יאשרו את חשבונם

שפות התכנות וסביבות העבודה

בפרויקט שלי עשיתי שימוש ב C# ו SQL, כאשר ב- C# רשמתי את מרבית הפרויקט והוא אחראי על ההצפנה והפענוח, על שליחת המידע דרך הסוקטים, על ממשק המשתמש, ועל פעילות הסרבר שכולל את קבלת ההודעות פענוח שלהם עיבודם ושליחת הודעת תגובה למשתמש/ים הנדרשים. המידע על המשתמשים הרשומים נשמר בשרת SQL, השרת מעביר את המידע לסרבר והסרבר שולף ממנו נתונים בעזרת שילוב השפות SQL ו C#.

הפרויקט נכתב בתוכנת הפיתוח Visual Studio 2015.

הפרויקט נרשם ונבדק אך רק על מערכת ההפעלה windows 10 64 bit, אולם הוא אמור לעבוד בכל שאר מערכות ההפעלה המודרניות של ווינדוס.



פירוט הבעיה האלגוריתמית וניתוחה

1. ביצוע תקשורת עם מספר לקוחות בו-זמנית.

השרת רשאי לתקשר אך ורק עם לקוח אחד על סוקט מסוים, לכן הוא אינו יכול לתפקד כאשר מחוברים אליו 2 משתמשים או יותר אם הוא עובד רק עם Socket אחד. במערכת שלנו יש צורך שהשרת ידע לתקשר בו זמנית עם מספר רב של משתמשים.

2. הצפנת המידע נגד צד שלישי:

על מנת שצד שלישי אשר "יושב" על החיבור בין הלקוח לשרת לא יוכל להבין את ההודעות אשר עוברות בין שני הצדדים, ועל מנת שלא נצטרך להעביר מידע רגיש כלשהו כדי להתחיל את התקשורת, יש להשתמש בהצפנה א-סימטרית. עם הצפנת המידע צצה עוד בעיה, להמיר את המידע לבייטים, להצפין אותו, לשלוח ולהמיר בחזרה בלי שהמידע ייפגם בנוסף על הצפנה א-סימטרית קיימת הגבלה על גודל המחרוזת שנשלחת.

3. יצירת CAPTCHA - בדיקת אנושיות:

על מנת שתכנות רובוטיות וזדוניות למיניהן לא יוכלו להציף את השרת או לנסות לפרוץ לחשבונות הלקוחות, יש לבצע בדיקת CAPTCHA בכל התחברות של לקוח חדש.

4. זיהוי הבקשות של הלקוחות ותגובות השרת

השרת מאזין כל הזמן ללקוח ולהפך, הבעיה העיקרית של תקשורת זו היא ההבנה ביניהם, לדוגמה כאשר הלקוח רוצה לבקש מלקוח אחר לשחק איתו כיצד השרת יודע שהלקוח רוצה לבקש משחק ולא לכתוב בצ'אט. השרת צריך לדעת איזו פעולה הלקוח רוצה לעשות בכל פעם שהוא מקבל ממנו הודעה, גם כן הלקוח צריך לדעת מהי המשמעות של כל הודעה מהשרת.

הוספת רובד של אבטחה ואימות מתרחשת בעת קישור דואר אלקטרוני לכל משתמש. לכן, בכל הרשמה של משתמש חדש יש על השרת לשלוח הודעת דואר אלקטרוני ללקוח, אשר מכילה קוד שעל המשתמש לשלוח בחזרה לשרת. שליחת הדואר האלקטרוני מתבצעת באמצעות חשבון מייל קבוע מראש כאשר ישנו שימוש ב SMTP - Simple Mail Transfer Protocol לכתיבת הודעה מהדואר האלקטרוני במקרה שלי - Gmail. התוכנה מייצרת קוד - רצף של אותיות באופן אקראי ושולח אותם בגוף ההודעה, לאחר מכן ללקוח נפתח חלון להכנסת הקוד (ישנה אפשרות לקבלת קוד חדש מהשרת). במידה והקוד נכון הלקוח יתחבר לתוכנה, במידה ולא תופיע הודעת שגיאה שהקוד שהוזן לא נכון.

פתרונות - יתרונות וחסרונות

1. ביצוע תקשורת עם מספר לקוחות בו-זמנית

a. שימוש בחיבור בפרוטוקול TCP-

i. יתרונות-

1. חיבור אמין ועקבי בין הלקוח לשרת, כל מידע שיישלח מהלקוח יתקבל בשרת ולהפך.

ii. חסרונות-

1. חיבור איטי לעומת אופציות אחרות, דורש מעבר של יותר הודעות.

b. שימוש ב Socket נפרד לכל לקוח

i. יתרונות-

1. עבודה עם כמה לקוחות בו זמנית ולמנוע ו"דריסת" Socket.

ii. חסרונות-

1. יוצר עומס, חוסר יעילות בניצול מירבי של כל Socket.

2. הצפנת המידע נגד צד שלישי

a. שימוש ב-RSA-

i. יתרונות-

1. כל התקשורת תהיה מאובטחת. פחות סכנות של האזנות מצד שלישי.

2. לא נדרשת הבנה מעמיקה כיצד מתבצעת ההצפנה

ii. חסרונות-

1. לא ניתן לדעת מה השרת מעביר ללקוח בכדי לפקח על תגובות השרת.

2. הגבלה על גודל המחרוזת הנשלחת.

b. שימוש בהצפנה סימטרית

i. יתרונות-

1. פחות הגבלה על גודל המחרוזת הנשלחת.

ii. חסרונות-

1. רמת אבטחה נחותה מאשר הצפנה א-סימטרית

3. יצירת CAPTCHA - בדיקת אנושיות - בצד שרת ולהעביר אותה ללקוח

a. בדיקה של קוד-

i. יתרונות-

1. מונע מעבר של לקוחות ללא בדיקה. במידה

והבדיקה בצד לקוח, הלקוח יכול לאשר את הבדיקה
בעצמו.

ii. חסרונות-

1. מכיוון שהקוד נוצר בצד הלקוח קיימת אפשרות
להסתכל על דרך יצירת הקוד.

2. אין עבודה מול השרת – יכול להוביל לבעיית
אבטחה.

b. יצירת ה CAPTCHA-

i. יתרונות-

1. צירוף אותיות אקראי כל פעם מחדש.

ii. חסרונות-

1. נתון לפריצה מכיוון ששמור בצד הלקח.

4. זיהוי הבקשות של הלקוח ותגובות השרת

a. יצירת הודעה בתבנית מוסכמת על ידי הלקוח-

i. יתרונות-

1. ייעול הקוד

ii. חסרונות-

1. שימוש בתבנית קבועה מגביל את המידע שניתן להעביר בהודעה

b. זיהוי הפקודות על ידי השרת בעזרת מילות מפתח-

i. יתרונות-

1. הבנה טובה של השרת והלקוח.

ii. חסרונות-

1. לא מוגן מפני צד שלישי. מאוד קל לזהות מהן הפקודות המוסכמות.

5. שליחת דואר אלקטרוני

a. שימוש בחשבון מייל

i. יתרונות-

1. ניהול חשבון בצורה נוחה.

ii. חסרונות-

1. נתון לפרצה.

b. שימוש בפרוטוקול Smtip

i. יתרונות-

1. אוטומציה פשוטה ונוחה לשליחת מייל

2. תמיכה בפרוטוקול SSL

ii. חסרונות-

1. פרטי ההתחברות למייל מופיעים בקוד

הפתרון שנבחר

כעת נפרט יותר על הפתרונות שמימשנו, למה מימשנו אותם, איך מימשנו אותם בקוד, ונביא דוגמאות.

1. ביצוע תקשורת עם מספר לקוחות בו-זמנית

תחילה השרת פותח את האזנה ומחכה ללקוחות שיתחברו. ברגע החיבור השרת מייצר עצם של המחלקה Client.

אחר יצירת האובייקט Client מתחיל תהליך התקשורת בין אותו לקוח לשרת.

לכל לקוח השרת מייצר TCPClient משלו ועל ידי כך פותח Socket לכל לקוח.

השרת מאזין ללקוחות באופן א-סינכרוני על מנת לבצע קשר עם מספר משתמשים במקביל.

```
class Program
{
    const int portNo = 500;
    private const string ipAddress = "127.0.0.1";

    static void Main(string[] args)
    {
        System.Net.IPAddress localAdd = System.Net.IPAddress.Parse(ipAddress);
        //System.Net.IPAddress localAdd = System.Net.IPAddress.Parse(add);
        //TcpListener listener = new TcpListener(System.Net.IPAddress.Any, portNo);
        TcpListener listener = new TcpListener(localAdd, portNo);

        Console.WriteLine("Experts4D - Simple TCP Server");
        Console.WriteLine("Listening to ip {0} port: {1}", ipAddress, portNo);
        Console.WriteLine("Server is ready.");

        // Start listen to incoming connection requests
        listener.Start();

        // infinit loop.
        while (true)
        {
            // AcceptTcpClient - Blocking call
            // Execute will not continue until a connection is established

            // We create an instance of Client so the server will be able to
            // server multiple client at the same time.
            Client user = new Client(listener.AcceptTcpClient());
        }
    }
}
```

2. הצפנה - RSA

בתחילת התקשורת שני הצדדים מעבירים אחד לשני את המפתח הציבורי.

קוד העברת המפתח הציבורי וקבלת המפתח הציבורי של הצד השני:

לקוח:

```
public Form1()
{
    InitializeComponent();
    //connect = new Thread(new ThreadStart(IsConnected));
    client = new TcpClient();
    client.Connect(ipAddress, portNo);
    r = new RSA();
    SendMessage("client public key:" + r.GetPublicKey());
    data = new byte[client.ReceiveBufferSize];
    client.GetStream().BeginRead(data,
                                0,
    System.Convert.ToInt32(client.ReceiveBufferSize),
                                ReceiveMessage,
                                null);
}
```

שרת:

```
if (messageReceived.StartsWith("client public key:"))//get the client public
key
{
    this.clientPublicKey =
messageReceived.Remove(0,18);//remove the header from the client message

    Console.WriteLine(clientPublicKey);

    string serverpublickey = rsa.GetPublicKey();//get the
server public key
    SendMessage("server public key:"+
serverpublickey);//send the server public key to client
}
```

לאחר מכן, כל מידע שעובר בין השרת והלקוח עובר קודם כל הצפנה ולאחר מכן פענוח:

לקוח – הצפנה:

```
private void sendBtn_Click(object sender, EventArgs e)
{
    string encryptedmss = r.Encrypt(textBox1.txt + textBox2.txt ,
serverPublicKey);
    MessageBox.Show("encryptedmss\r\n" + encryptedmss);
    SendMessage(encryptedmss);
}
```

שרת – פענוח:

```
else
{
    Console.WriteLine("messageReceived:\r\n" +
messageReceived);
    Console.ReadKey();
    string decryptedMessage = rsa.Decrypt(messageReceived);
    Console.WriteLine(decryptedMessage);
}
```

פעולות הצפנה ופענוח:

```
public string Decrypt(string data,string privateKey)
{
    var dataArray = data.Split(new char[] { ',' });
    byte[] dataByte = new byte[dataArray.Length];
    for (int i = 0; i < dataArray.Length; i++)
    {
        dataByte[i] = Convert.ToByte(dataArray[i]);
    }

    _rsa.FromXmlString(privateKey);
    var decryptedByte = _rsa.Decrypt(dataByte, false);
    return _encoder.GetString(decryptedByte);
}
```

```
public string Encrypt(string data,string publicKey)
{
    var rsa = new RSACryptoServiceProvider();
    rsa.FromXmlString(publicKey);
    var dataToEncrypt = _encoder.GetBytes(data);
    var encryptedByteArray = rsa.Encrypt(dataToEncrypt, false);
    var length = encryptedByteArray.Length;
    var item = 0;
    var sb = new StringBuilder();
    foreach (var x in encryptedByteArray)
    {
        item++;
        sb.Append(x);
    }
}
```

```
        if (item < length)
            sb.Append(",");
    }
    return sb.ToString();
}
```

3. יצירת CAPTCHA - בדיקת אנושיות - בצד שרת והעברתה ללקוח

התוכנה מייצרת את CAPTCHA במהלך ניסיון ההתחברות / ההרשמה /
שחזור סיסמא.

-המחלקה Captcha:

```
public partial class Captcha : Form
{
    Utilities u = new Utilities();
    int x;
    private string messageR;
    public Captcha()
    {
        InitializeComponent();
        pictureBox1.Visible = false;
        pictureBox2.Visible = false;
        this.ControlBox = false;
        this.Text = " ";
    }

    private void Captcha_Load(object sender, EventArgs e)
    { //Randomize What Captch The User Will Need To Complete.
        Random rnd = new Random();
        x = rnd.Next(1, 3);

        if (x == 1)
        {
            pictureBox1.Visible = true;
        }
        else
        {
            pictureBox2.Visible = true;
        }
    }

    private void button1_Click(object sender, EventArgs e)
    { //Check If The User Complete The Captch Correctly
      //Yes - Message Pop Up "Correct", No - Message Pop Up "Try Again".
        string s = "";
        string str = "";

        if(pictureBox2.Visible == true)
        {
            s = "Website";
            str = "website";
        }

        if(pictureBox1.Visible == true)
        {
            s = "Foodie";
            str = "foodie";
        }

        if(textBox1.Text.Equals(s) || textBox1.Text.Equals(str))
        {
            MessageBox.Show("Correct");

            this.Visible = false;
        }
    }
}
```



```

else
{
    MessageBox.Show("Try Again");
}
}
public void UpdateMss(string str)
{
    this.messageR = str;
}
}
}

```

-לאחר יצירת התמונה התוכנה תציג חלון שבה מופיעה התמונה.

-כאשר הלקוח מכניס את תשובתו, מתקיים אימות ולאחר האימות הלקוח רשאי להמשיך.

4. זיהוי הבקשות של הלקוח ותגובות השרת-

-כאשר השרת מקבל הודעה מהלקוח מתבצע ניתוח ההודעה לפי מילות / סימני מפתח בתחילת המחרוזת.

מילות מפתח:

- 1. /
- 2. #
- 3. %
- 4. Success
- 5. Eboard
- 6. Ask
- 7. GameS
- 8. Bchat
- 9. Lchat
- 10. New Pass:
- 11. Failed
- 12. Ready
- 13. Mboard
- 14. Newl
- 15. Yes
- 16. No

5. שליחת דואר אלקטרוני-

-לצורך שליחת הדואר האלקטרוני קיימת מחלקה ובה נוצרת ההודעה ומתרחשת ההתחברות אל חשבון Gmail ושולחת את הודעה אל המייל הדרוש

```
public void sendMail(string Address, string Message, string Subject)
{
    //Send Mail Function

    var fromAddress = new MailAddress("ofekPro21@gmail.com",
    "Project"); //Define The Sender
    var toAddress = new MailAddress( Address ); // Define The Reciver
    const string fromPassword = "Ofek!Pro1"; //Sender Password
    string subject = Subject; //Define Message Subject
    string body = Message; //Define Message Content

    var smtp = new SmtpClient // Define The Mail Service
    {
        Host = "smtp.gmail.com", // Mail Service Host
        Port = 587,
        EnableSsl = true, // Enable Secure Sending
        DeliveryMethod = SmtpDeliveryMethod.Network, // Define The
        Delivery Method
        UseDefaultCredentials = false, //Disable Default Setting
        Credentials = new NetworkCredential(fromAddress.Address,
        fromPassword) // Define The Credentials
    };
    using (var message = new MailMessage(fromAddress,
    toAddress)) //Define The Message Setting
    {
        Subject = subject,
        Body = body
    })
    {
        smtp.Send(message); // Sending The Message
        MessageBox.Show("Message send Successfully");
    }
}
```

יצור הקוד שנשלח:

```
public string CodeGenerate ()
    {
        //Generating A Code With 8 Chars.
        var chars =
            "ABCDEFGHIIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"; //Define The
            Chars Appers In Code
        var stringChars = new char[8];
        var random = new Random();

        for (int i = 0; i < stringChars.Length; i++)
        {
            stringChars[i] = chars[random.Next(chars.Length)];
        }

        string finalString = new String(stringChars);
        return finalString;
    }
```

דרישות ומגבלות המערכת

דרישות

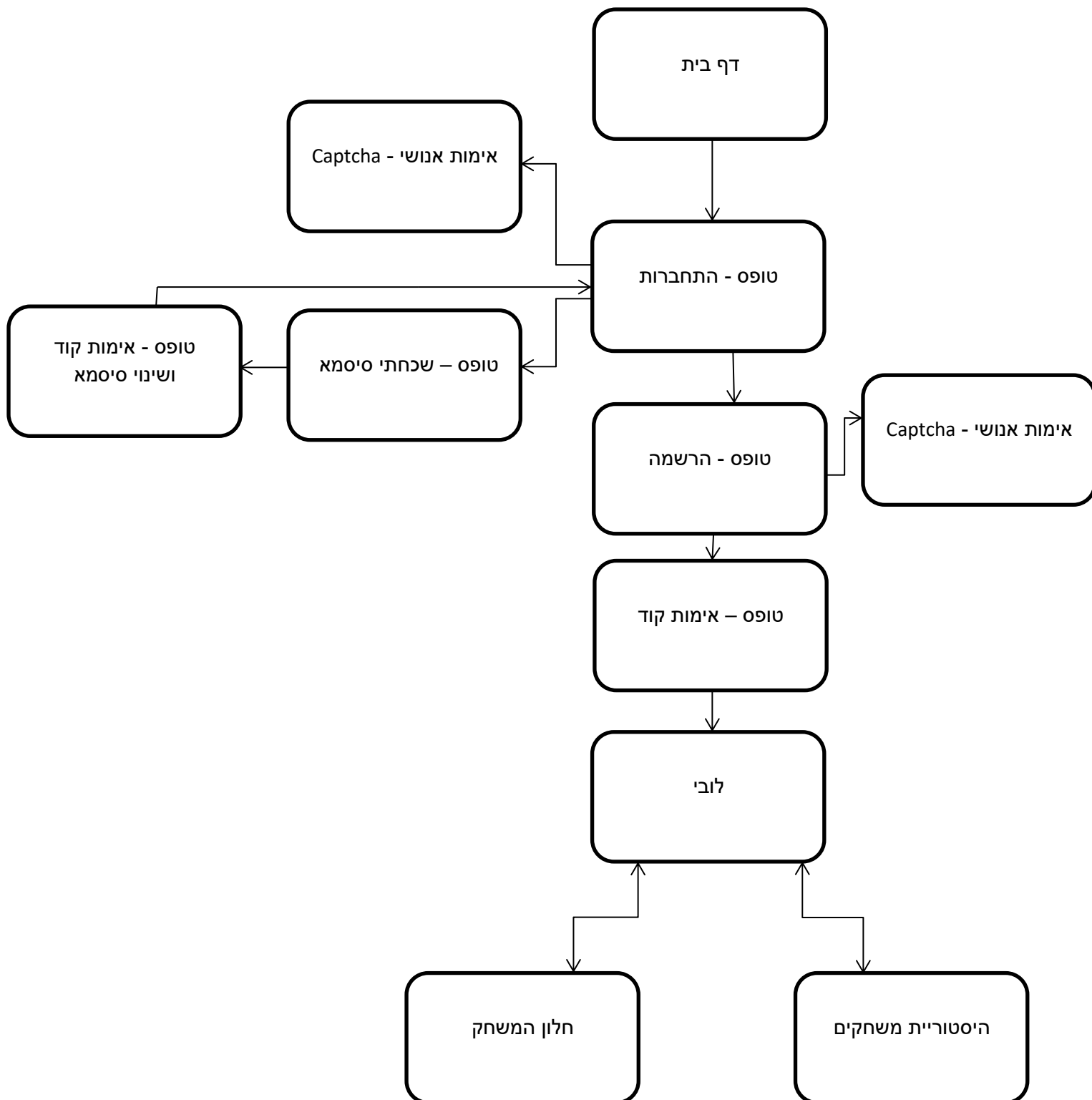
על מנת שהתכנית תפעל ישנן כמה דרישות שצריכות להיענות קודם לכן:

- מערכת הפעלה Windows XP ומעלה על המחשב.

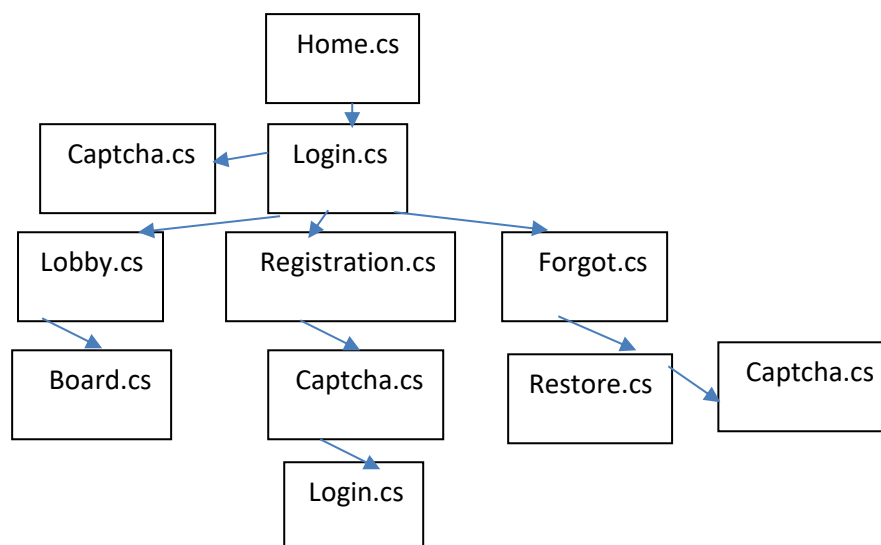
מוגבלויות

- תמיכה בשפה האנגלית בלבד.
- הנחה בסיסית שהלקוח יודע את הIP והפורט שעליהם מאזין השרת.
- הגבלה של 256 בייטים בסטרים בין השרת ללקוח.
- יכול להיווצר עיכוב בתקשורת בין הלקוח לשרת בתחילת התקשורת עקב החלפת המפתחות הציבוריים.

ממשק משתמש

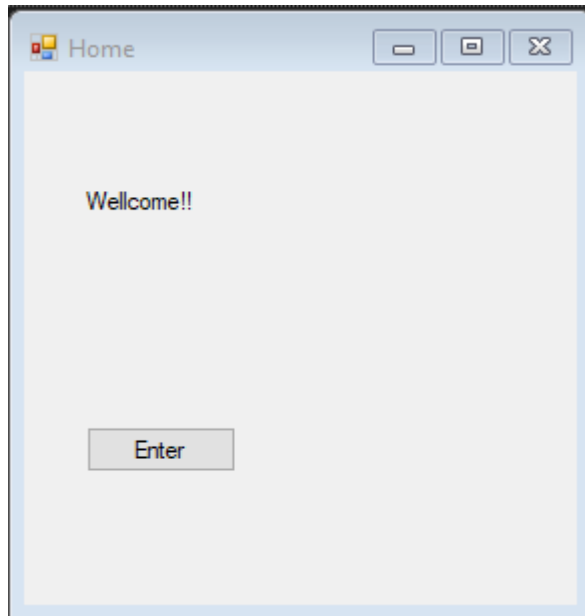


תרשים זרימה למחלקות השרת

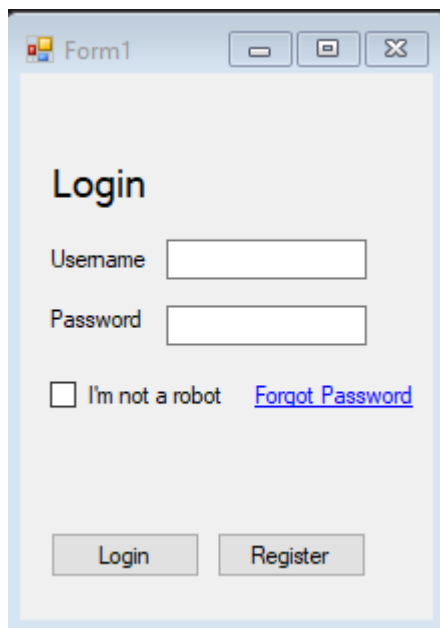


חלונות הפרוייקט

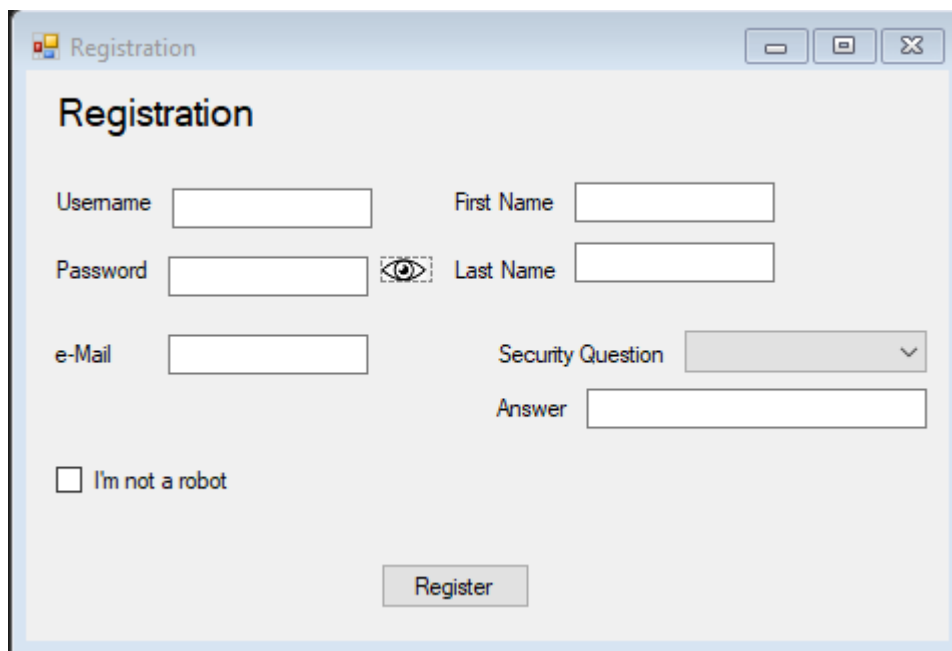
Home:



Login:




Registration:



A screenshot of a Windows-style application window titled "Registration". The window contains several input fields and a button. The fields are arranged in two columns. The left column has "Username", "Password", and "e-Mail". The right column has "First Name", "Last Name", "Security Question" (a dropdown menu), and "Answer". There is a checkbox labeled "I'm not a robot" below the "e-Mail" field. A "Register" button is centered at the bottom. A small eye icon is next to the "Password" field.

Registration

Username First Name

Password  Last Name

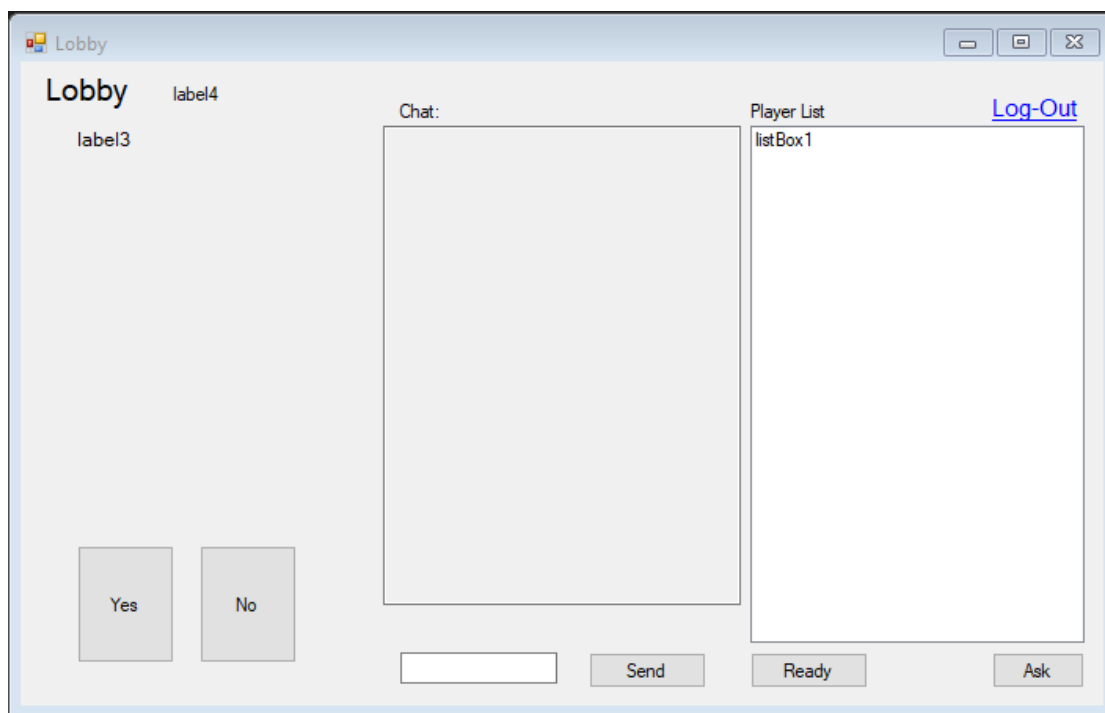
e-Mail Security Question

Answer

☐ I'm not a robot

Register

Lobby:



A screenshot of a Windows-style application window titled "Lobby". The window is divided into several sections. On the left, there are labels "label3" and "label4". In the center, there is a large rectangular area labeled "Chat:". To the right of the chat area is a "Player List" section containing a list box labeled "listBox1". At the bottom left, there are two buttons labeled "Yes" and "No". At the bottom center, there is a text input field, a "Send" button, and a "Ready" button. At the bottom right, there is an "Ask" button and a "Log-Out" link.

Lobby

label4

label3

Chat:

Player List

listBox1

Log-Out

Yes No

Send Ready Ask

Board:

Board

label32

Your Board

A	B	C	D	E	F	
						6
						5
						4
						3
						2
						1

Direction

Position

Type

Set Ship

Enemy Board - label34

A	B	C	D	E	F	
						6
						5
						4
						3
						2
						1

Position

Wait For Opponent To Start Attacking His Ships

Attack

Chat With Player

Send

Forgot:

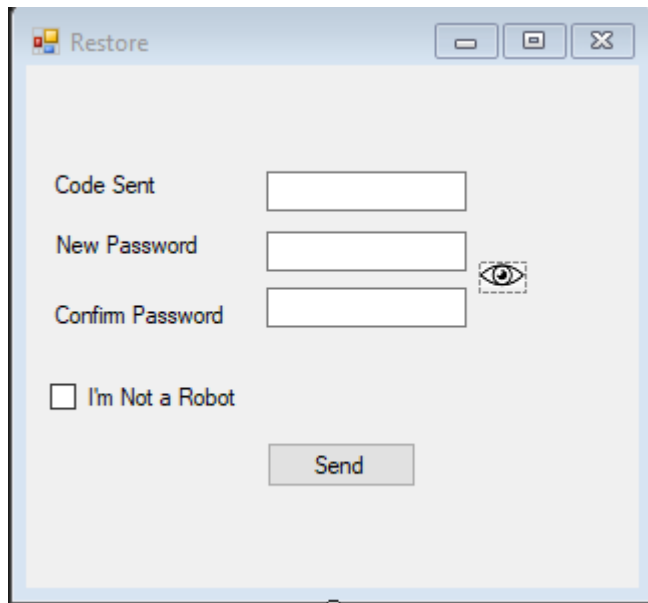
Forgot

Mail Adress

Username

Send

Restore:



A screenshot of a web browser window titled "Restore". The window contains a form with the following elements:

- A label "Code Sent" followed by a text input field.
- A label "New Password" followed by a text input field.
- A label "Confirm Password" followed by a text input field.
- A checkbox labeled "I'm Not a Robot".
- A "Send" button.
- A small eye icon to the right of the password fields, indicating a toggle for password visibility.

Captcha:



A screenshot of a web browser window titled "Captcha". The window contains a form with the following elements:

- A captcha image showing the word "Website" in a stylized font.
- A text input field for the user to enter the captcha text.
- A "Submit" button.

פעולות בפרויקט

טבלת פעולות – לקוח

כותרת	מחלקה	טענת קלט / פלט
Board(Lobby I , String opp)	Board.cs	בנאי - הבנאי מקבל טיפוס (Opp וסטרינג Login מסוג השחקן השני) , ומאתחלת את המשתנים.
send_Click(object sender , EventArgs e)	Board.cs	הפעולה מתרחשת כאשר Send נלחץ כפתור ה - ושולחת את הודעת הטקסט של הלקוח לסרבר (משמש להפעלת הצ'אט)
Set_click(object sender , EventArgs e)	Board.cs	הפעולה שומרת את ערכי המשבצת שנבחרה ומגדירה אותה כסירה של השחקן
delUpdateHistory(string str)	Board.cs	של הפונקצייה Delegate המעדכנת את הצ'אט
UpdateHistory(string str)	Board.cs	מעדכנת את הצ'אט
get_posA(string s)	Board.cs	מחזיר את הערך הגימטרי של האות המתקבלת
get_posB(string s)	Board.cs	מחזיר את המספר שהתקבל
get_lPos(int i)	Board.cs	מחזיר את האות לפי הערך המספרי שהתקבל
updateSBoard(string c, string r)	Board.cs	מקבל משבצת בלוח ומעדכן את המטריצה שמייצגת את הלוח של המשתמש
rem_op()	Board.cs	מסיר את האופציה של הסירה שנבחרה לשים
Att_Click(object sender, EventArgs e)	Board.cs	תוקף רת לוח הסירות של היריב לפי המשבצת שנבחרה
updateEBoard(string c, string r)	Board.cs	מקבל משבצת בלוח ומעדכן את המטריצה שמייצגת את הלוח של היריב

מחזיר את תכונת ה - של CommunicationHelper הForm	Board.cs	CommunicationHelper GetCh()
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Board.cs	UpdateMss(string str)
בנאי - מאתחל את המחלקה ומגדיר משתנים	Captcha.cs	Captcha()
עולה בוחר FORM כאשר ה ערך רנדומלי ומציב במשתנה וריו יודע איזו תמונה להציג X למשתמש	Captcha.cs	Captcha_Load(object sender, EventArgs e)
מאמת את המידע שהוכנס על ידי המשתמש ובודק האם מתאים לתמונה שנבחרה	Captcha.cs	button1_Click(object sender, EventArgs e)
בנאי המחלקה - מגדיר את פרטי הלקוח	CommunicationHelper.cs	CommunicationHelper(Home c)
Form מאתחל את תכונת ה שאליו נכנסים	CommunicationHelper.cs	InitializeBoardForm(Board brd)
Form מאתחל את תכונת ה שאליו נכנסים	CommunicationHelper.cs	InitializeForgotForm(Forgot fgt)
Form מאתחל את תכונת ה שאליו נכנסים	CommunicationHelper.cs	InitializeGameHForm(GameH gh)
Form מאתחל את תכונת ה שאליו נכנסים	CommunicationHelper.cs	InitializeLobbyForm(Lobby lb)
Form מאתחל את תכונת ה שאליו נכנסים	CommunicationHelper.cs	InitializeRegistrationForm(Registration rg)

Form מאתחל את תכונת ה שאליו נכנסים	CommunicationH elper.cs	InitializeRestoreFo rm(Restore rst)
Form מאתחל את תכונת ה שאליו נכנסים	CommunicationH elper.cs	InitializeHomeFor m(Home localclient)
מקבל את ההודעה מהשרת הפתוח FORM ושולח אותה ל	CommunicationH elper.cs	ReceiveInfo(IAsync Result ar)
מקבל את ההודעה מהלקוח ושולח אות לשרת	CommunicationH elper.cs	SendInfo(string info)
מחזיר את שם המשתמש	CommunicationH elper.cs	get_name()
מקבל סטרינג ומגדיר אותו כשם המשתמש	CommunicationH elper.cs	set_name(string s)
של הפונקצייה Delegate הממאתחלת את הפורם	CommunicationH elper.cs	delHomeUpdateM ss(string str);
של הפונקצייה Delegate הממאתחלת את הפורם	CommunicationH elper.cs	delBoardUpdateM ss(string str);
של הפונקצייה Delegate הממאתחלת את הפורם	CommunicationH elper.cs	delCaptchaUpdate Mss(string str);
של הפונקצייה Delegate הממאתחלת את הפורם	CommunicationH elper.cs	delForgotUpdateM ss(string str);
של הפונקצייה Delegate הממאתחלת את הפורם	CommunicationH elper.cs	delGameHUpdate Mss(string str);

של הפונקצייה Delegate הממאתחלת את הפורם	CommunicationHelper.cs	delLobbyUpdateMss(string str);
של הפונקצייה Delegate הממאתחלת את הפורם	CommunicationHelper.cs	delLoginUpdateMss(string str);
של הפונקצייה Delegate הממאתחלת את הפורם	CommunicationHelper.cs	delRegistrationUpdateMss(string str);
של הפונקצייה Delegate הממאתחלת את הפורם	CommunicationHelper.cs	delRestoreUpdateMss(string str);
בנאי - מאתחל את המחלקה ומגדיר משתנים	Forgot.cs	Forgot(Login l)
שולח את המייל ושם המשתמש לשרת לאימות	Forgot.cs	button1_Click(object sender, EventArgs e)
מחזיר את תכונת ה - של CommunicationHelper הForm	Forgot.cs	CommunicationHelper GetCh()
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Forgot.cs	UpdateMss(string str)
בנאי - מאתחל את המחלקה חדש לחיבור Thread ומפעיל לשרת.	Home.cs	Home()
עובר לחלון ההתחברות	Home.cs	button1_Click(object sender, EventArgs e)
מחזיר את תכונת ה - של CommunicationHelper הForm	Home.cs	CommunicationHelper getCh()
בנאי - מאתחל את המחלקה ומגדיר משתנים	Login.cs	Login(Home h)

בנאי - מאתחל את המחלקה ומגדיר משתנים	Login.cs	Login(Registration r)
בנאי - מאתחל את המחלקה ומגדיר משתנים	Login.cs	Login(Restore rs)
מחזיר את תכונת ה - של CommunicationHelper GetCh() הForm	Login.cs	CommunicationHel per GetCh()
עובר לחלון ההרשמה	Login.cs	button2_Click(object sender, EventArgs e)
שולח את פרטי ההתחברות לשרת	Login.cs	button1_Click(object sender, EventArgs e)
captchaפופות את חלון ה	Login.cs	checkBox1_CheckedChanged(object sender, EventArgs e)
Forgotפופות את חלון ה	Login.cs	linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Login.cs	UpdateMss(string str)
מפעיל את התוכנה	Program .cs	static void Main()
בנאי - מאתחל את המחלקה ומגדיר משתנים	Registration.cs	Registration(Login l)
בודק אם הפרטים שהוזנו מתאימים לתנאים שהוחלטו בעת כתיבת התוכנה	Registration.cs	Check()
מראה את הסיסמא	Registration.cs	pictureBox1_Click(object sender, EventArgs e)

מסתיר את הסיסמא	Registration.cs	pictureBox1_DoubleClick(object sender, EventArgs e)
captcha פותח את חלון ה	Registration.cs	checkbox1_CheckedChanged(object sender, EventArgs e)
אם הפרטים שהוזנו תאימים שולח אותם לשרת אחרת מראה הודעה על הפרטים שלא מתאימים	Registration.cs	button1_Click(object sender, EventArgs e)
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Registration.cs	UpdateMss(string str)
מחזיר את תכונת ה - של CommunicationHelper הForm	Registration.cs	CommunicationHelper GetCh()
בנאי - מקבל את שם המשתמש והקוד שהוגרל. מאתחל את המחלקה	Restore.cs	Restore(string s , string user , Forgot f)
בודק אם הקוד שהוזן מתאים לקוד שנוצר בודק אם הסיסמאות מתאימות ושז שולח את הסיסמא החדשה ושם המשתמש לשרת	Restore.cs	button1_Click(object sender, EventArgs e)
מראה את הסיסמא	Restore.cs	pictureBox1_Click(object sender, EventArgs e)
מסתיר את הסיסמא	Restore.cs	pictureBox1_DoubleClick(object sender, EventArgs e)
captcha פותח את חלון ה	Restore.cs	checkBox1_CheckedChanged(object

		sender, EventArgs)
מחזיר את תכונת ה - של CommunicationHelper הForm	Restore. cs	CommunicationHelper GetCh()
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Restore. cs	UpdateMss(string str)

טבלת פעולות – שרת

כותרת	מחלקה	טענת קלט / פלט
Board(Lobby l , String opp)	Board.cs	בנאי - הבנאי מקבל טיפוס Login מסוג Opp (וסטרינג השחקן השני) , ומאתחלת את המשתנים.
send_Click(object sender , EventArgs e)	Board.cs	הפעולה מתרחשת כאשר נלחץ Send כפתור ה - ושולחת את הודעת הטקסט של הלקוח לסרבר (משמש להפעלת הצ'אט)
Set_click(object sender , EventArgs e)	Board.cs	הפעולה שומרת את ערכי המשבצת שנבחרה ומגדירה אותה כסירה של השחקן
delUpdateHistory(string str)	Board.cs	של Delegate הפונקציה המעדכנת את הצ'אט
UpdateHistory(string str)	Board.cs	מעדכנת את הצ'אט
get_posA(string s)	Board.cs	מחזיר את הערך הגימטרי של האות המתקבלת
get_posB(string s)	Board.cs	מחזיר את המספר שהתקבל

מחזיר את האות לפי הערך המספרי שהתקבל	Board.cs	get_IPos(int i)
מקבל משבצת בלוח ומעדכן את המטריצה שמייצגת את הלוח של המשתמש	Board.cs	updateSBoard(string c, string r)
מסיר את האופציה של הסירה שנבחרה לשים	Board.cs	rem_op()
תוקף רת לוח הסירות של היריב לפי המשבצת שנבחרה	Board.cs	Att_Click(object sender, EventArgs e)
מקבל משבצת בלוח ומעדכן את המטריצה שמייצגת את הלוח של היריב	Board.cs	updateEBoard(string c, string r)
מחזיר את תכונת ה - Communication Helper של הForm	Board.cs	CommunicationHelper GetCh()
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Board.cs	UpdateMss(string str)
בנאי - מאתחל את המחלקה ומגדיר משתנים	Captcha.cs	Captcha()
FORM כאשר ה עולה בוחר ערך רנדומלי ומציב וריו אבמשתנה	Captcha.cs	Captcha_Load(object sender, EventArgs e)

יודע איזו תמונה להציג למשתמש		
מאמת את המידע שהוכנס על ידי המשתמש ובודק האם מתאים לתמונה שנבחרה	Captcha.cs	button1_Click(object sender, EventArgs e)
בנאי המחלקה - מגדיר את פרטי הלקוח	Communication Helper.cs	CommunicationHelper(Home c)
מאתחל את תכונת שאליו Form הנכנסים	Communication Helper.cs	InitializeBoardForm(Board brd)
מאתחל את תכונת שאליו Form הנכנסים	Communication Helper.cs	InitializeForgotForm(Forgot fgt)
מאתחל את תכונת שאליו Form הנכנסים	Communication Helper.cs	InitializeGameHForm(GameH gh)
מאתחל את תכונת שאליו Form הנכנסים	Communication Helper.cs	InitializeLobbyForm(Lobby lb)
מאתחל את תכונת שאליו Form הנכנסים	Communication Helper.cs	InitializeRegistrationForm(Registration rg)
מאתחל את תכונת שאליו Form הנכנסים	Communication Helper.cs	InitializeRestoreForm(Restore rst)
מאתחל את תכונת שאליו Form הנכנסים	Communication Helper.cs	InitializeHomeForm(Home localclient)
מקבל את ההודעה מהשרת ושולח FORM אותה ל הפתוח	Communication Helper.cs	ReceiveInfo(IAsyncResult ar)

מקבל את ההודעה מהלקוח ושולח אות לשרת	Communication Helper.cs	SendInfo(string info)
מחזיר את שם המשתמש	Communication Helper.cs	get_name()
מקבל סטרינג ומגדיר אותו כשם המשתמש	Communication Helper.cs	set_name(string s)
של Delegate הפונקציה הממאתחלת את הפורם	Communication Helper.cs	delHomeUpdateMss(string str);
של Delegate הפונקציה הממאתחלת את הפורם	Communication Helper.cs	delBoardUpdateMss(string str);
של Delegate הפונקציה הממאתחלת את הפורם	Communication Helper.cs	delCaptchaUpdateMss(string str);
של Delegate הפונקציה הממאתחלת את הפורם	Communication Helper.cs	delForgotUpdateMss(string str);
של Delegate הפונקציה הממאתחלת את הפורם	Communication Helper.cs	delGameHUpdateMss(string str);
של Delegate הפונקציה הממאתחלת את הפורם	Communication Helper.cs	delLobbyUpdateMss(string str);
של Delegate הפונקציה הממאתחלת את הפורם	Communication Helper.cs	delLoginUpdateMss(string str);

של Delegate הפונקצייה הממאתחלת את הפורם	Communication Helper.cs	delRegistrationUpdate Mss(string str);
של Delegate הפונקצייה הממאתחלת את הפורם	Communication Helper.cs	delRestoreUpdateMss(s tring str);
בנאי - מאתחל את המחלקה ומגדיר משתנים	Forgot.cs	Forgot(Login l)
שולח את המייל ושם המשתמש לשרת לאימות	Forgot.cs	button1_Click(object sender, EventArgs e)
מחזיר את תכונת ה - של Communication Helper הForm	Forgot.cs	CommunicationHelper GetCh()
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Forgot.cs	UpdateMss(string str)
בנאי - מאתחל את המחלקה ומפעיל חדש Thread לחיבור לשרת.	Home.cs	Home()
עובר לחלון ההתחברות	Home.cs	button1_Click(object sender, EventArgs e)
מחזיר את תכונת ה - של Communication Helper הForm	Home.cs	CommunicationHelper getCh()

בנאי - מאתחל את המחלקה ומגדיר משתנים	Login.cs	Login(Home h)
בנאי - מאתחל את המחלקה ומגדיר משתנים	Login.cs	Login(Registration r)
בנאי - מאתחל את המחלקה ומגדיר משתנים	Login.cs	Login(Restore rs)
מחזיר את תכונת ה - Communication Helper של הForm	Login.cs	CommunicationHelper GetCh()
עובר לחלון ההרשמה	Login.cs	button2_Click(object sender, EventArgs e)
שולח את פרטי ההתחברות לשרת	Login.cs	button1_Click(object sender, EventArgs e)
פותח את חלון ה captcha	Login.cs	checkBox1_CheckedCha nged(object sender, EventArgs e)
פותח את חלון ה Forgot	Login.cs	linkLabel1_LinkClicked(o bject sender, LinkLabelLinkClickedEve ntArgs e)
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Login.cs	UpdateMss(string str)
מפעיל את התוכנה	Program.cs	static void Main()
בנאי - מאתחל את המחלקה ומגדיר משתנים	Registration.cs	Registration(Login l)
בודק אם הפרטים שהוזנו מתאימים	Registration.cs	Check()

לתנאים שהוחלטו בעת כתיבת התוכנה		
מראה את הסיסמא	Registration.cs	pictureBox1_Click(object sender, EventArgs e)
מסתיר את הסיסמא	Registration.cs	pictureBox1_DoubleClick(object sender, EventArgs e)
פותח את חלון ה captcha	Registration.cs	checkbox1_CheckedChanged(object sender, EventArgs e)
אם הפרטים שהוזנו תאימים שולח אותם לשרת אחרת מראה הודעה על הפרטים שלא מתאימים	Registration.cs	button1_Click(object sender, EventArgs e)
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Registration.cs	UpdateMss(string str)
מחזיר את תכונת ה - Communication Helper של הForm	Registration.cs	CommunicationHelper GetCh()
בנאי - מקבל את שם המשתמש והקוד שהוגרל. מאתחל את המחלקה	Restore.cs	Restore(string s , string user , Forgot f)
בודק אם הקוד שהוזן מתאים לקוד שנוצרץ בודק אם הסיסמאות	Restore.cs	button1_Click(object sender, EventArgs e)

מתאימות ושאז שולח את הסיסמא החדשה ושם המשתמש לשרת		
מראה את הסיסמא	Restore.cs	pictureBox1_Click(object sender, EventArgs e)
מסתיר את הסיסמא	Restore.cs	pictureBox1_DoubleClick(object sender, EventArgs e)
פותח את חלון ה captcha	Restore.cs	checkBox1_CheckedChanged(object sender, EventArgs e)
מחזיר את תכונת ה - Communication Helper של הForm	Restore.cs	CommunicationHelper GetCh()
מקבל את ההודעה מהשרת ומחליט איזו פעולה לבצע לפיה	Restore.cs	UpdateMss(string str)

כותרת	מחלקה	טענת קלט / פלט
Client(TcpClient C)	Client.cs	בנאי - מקבל לקוח ששלח בקשה להתחברות לשרת ולפיו מאתחל את פרטי המחלקה
SendMessage(string message)	Client.cs	שולח הודעה ללקוח
ReceiveMessage(IAsync Result ar)	Client.cs	מקבל את ההודעה ששלח הלקוח מפענח אותה ומבצע פעולות לפיה
Broadcast(string message)	Client.cs	שולח הודעה לכל המשתמשים המחוברים
Spec(string message , string Reciver)	Client.cs	שולח הודעה למשתמש ספציפי שמחובר לפי שם המשתמש שלו
GameCast(string P1 , string P2 , string message)	Client.cs	שולח הודעה לשתי משתמשים שמחוברים יחד למשחק אחד
Dal()	Dal.cs	בנאי - מאתחל את המחלקה ויוצר חיבור למסד הנתונים
AddUser(string user, string pass, string email, string first, string last, string seq , string answer)	Dal.cs	מקבל את פרטי ההרשמה ומוסיף אותה למסד הנתונים
CheckLogin (string user, string pass)	Dal.cs	בודק אם קיים משתמש העונה לנתונים שהוזנו
CheckMail(string email, string user)	Dal.cs	בודק אם קיים המשתמש שהוזן מחובר אל חשבון המייל שהוזן
UpdatePass(string user , string pass)	Dal.cs	מעדכן את הסיסמא לפי הנתונים שהוזנו

מעדכן את המייל לפי הנתונים שהוזנו	Dal.cs	UpdateEmail(string user, string mail)
בנאי - מאתחל את המחלקה ומגדיר משתנים	GameS.cs	GameS(string p1 , string p2 , string n1 , string n2)
מגדיר את שחקן 1	GameS.cs	Setp1 (string s)
מגדיר את שחקן 2	GameS.cs	Setp2 (string s)
מחזיר את שחקן 1	GameS.cs	Getp1()
מחזיר את שחקן 2	GameS.cs	Getp2()
מגדיר את הלוח של שחקן 1	GameS.cs	Get_B1()
מגדיר את הלוח של שחקן 2	GameS.cs	Get_B2()
מחזיר את הלוח של שחקן 1	GameS.cs	Update_p1B(int r, int c)
מחזיר את הלוח של שחקן 2	GameS.cs	Update_p2B(int r, int c)
בנאי - מאתחל את המחלקה ומגדיר משתנים	Player.cs	Player(string n , bool r)
מחזיר את שם השחקן	Player.cs	GetName()
מחזיר האם השחקן מוכן לשחק	Player.cs	GetReady()
מגדיר האם השחקן מוכן לשחק	Player.cs	SetReady(bool b)
פעולה ראשית - מפעיל את השרת ומתחיל להאזין למשתמשים באופן א-סנכרוני	Program.cs	Main(string[] args)