

BÁO CÁO THỰC HÀNH

Môn học: Lập trình mạng căn bản

Buổi báo cáo: Lab 03

Tên chủ đề: Lập trình Sockets trong C#

GVHD: Nguyễn Xuân Hà

Ngày thực hiện: 8/5/2024

THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT106.O23.2

STT	Họ và tên	MSSV	Email
1	Phạm Huỳnh Tấn Khang	22520624	22520624@gm.uit.edu.vn

1. ĐÁNH GIÁ KHÁC:

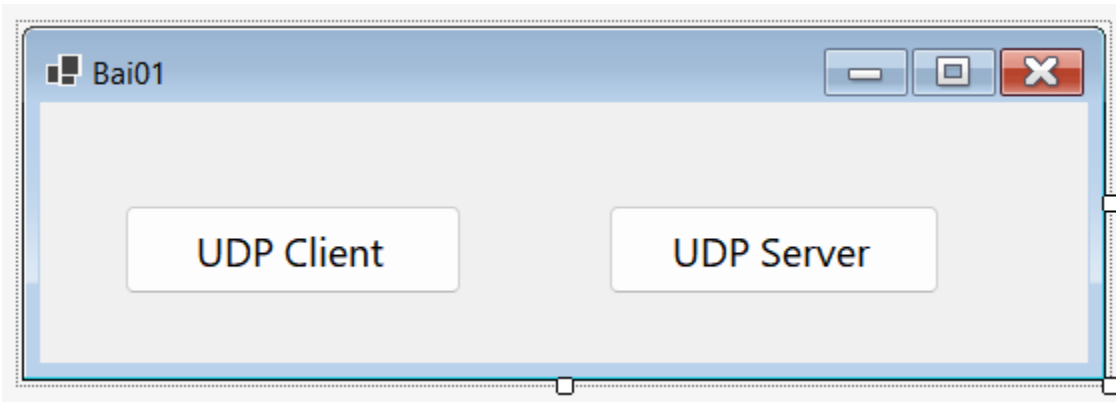
Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	10 tiếng
Link GitHub (nếu có)	https://github.com/VitalsZen
Ý kiến (nếu có) + Khó khăn + Đề xuất ...	
Điểm tự đánh giá	

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

Bài 01 –

Giao diện Dashboard

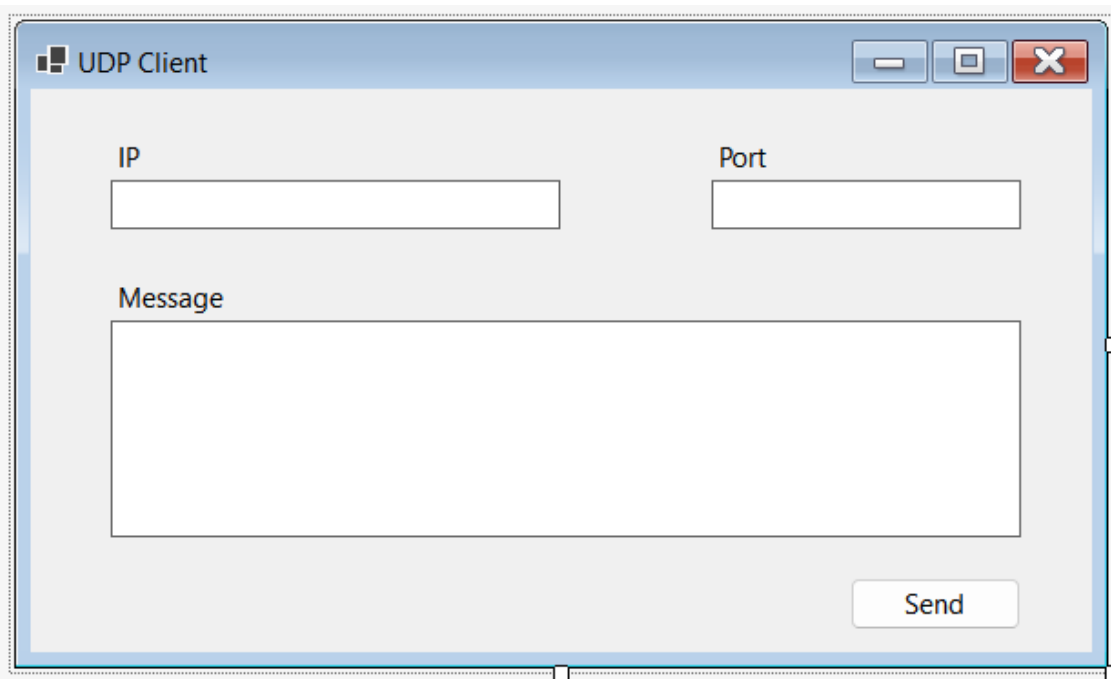


Hàm mở form

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    var Form = new UDPClient1();
    Form.Show();
}

1 reference
private void button2_Click(object sender, EventArgs e)
{
    var Form = new UDPServer1();
    Form.Show();
}
}
```

Giao diện UDP Client

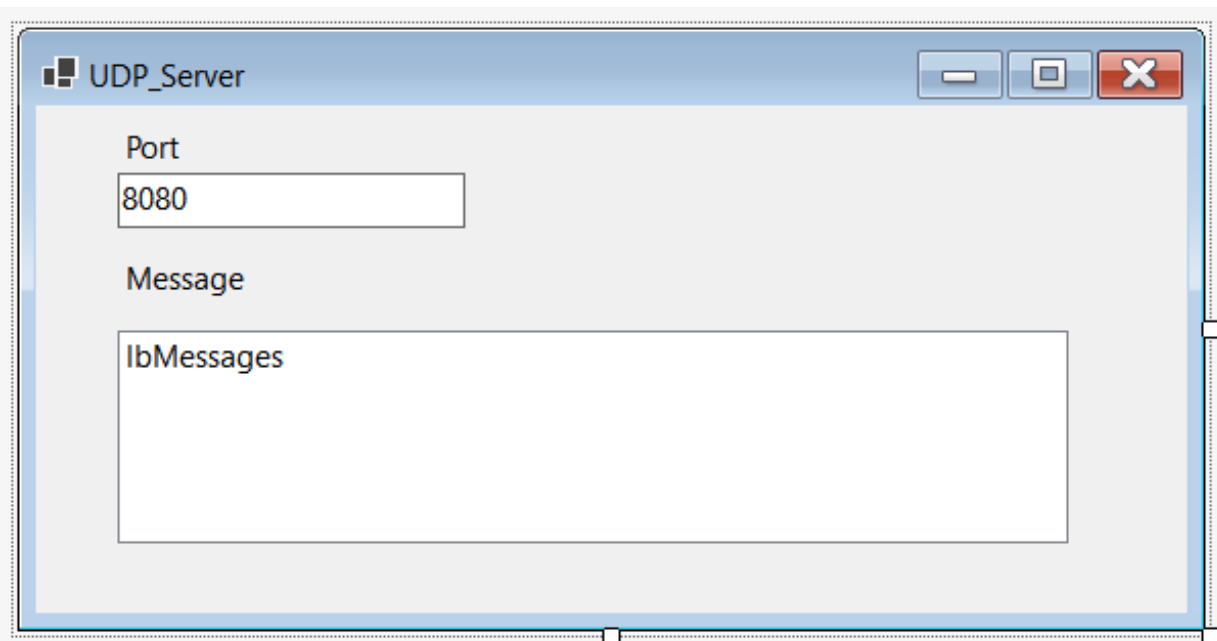


Hàm kích hoạt khi được click của button "Send", gửi thông điệp đến IP và port xác định bằng bytes

1 reference

```
private void btSend_Click(object sender, EventArgs e)
{
    if (tbIP.Text == "" || tbPort.Text == "")
    {
        MessageBox.Show("Địa chỉ IP: \nPort của Server: 8080", "Cảnh báo!!");
    }
    UdpClient udpClient = new UdpClient();
    udpClient.Connect(tbIP.Text, Int32.Parse(tbPort.Text));
    Byte[] sendBytes = Encoding.ASCII.GetBytes(tbMessage.Text);
    udpClient.Send(sendBytes);
}
```

Giao diện UDP_Server



Hàm `serverThread()` dùng để xây dựng server có giao thức UDP dùng để nhận thông điệp từ `IPEndPoint` bất kì

1 reference

```
public void serverThread()
{
    UdpClient udpClient = new UdpClient(8080);
    while (true)
    {
        IPEndPoint RemoteIPEndPoint = new IPEndPoint(IPAddress.Any, 0);
        Byte[] receiveBytes = udpClient.Receive(ref RemoteIPEndPoint);
        string returnData = Encoding.ASCII.GetString(receiveBytes);
        string mess = RemoteIPEndPoint.Address.ToString() + "(" +
            RemoteIPEndPoint.Port.ToString() + "):" + returnData.ToString();

        InfoMessage(mess);
    }
}
```

Hàm InfoMessage dùng để thực hiện để đẩy thông điệp được nhận từ hàm serverThread() lên listview lbMessages ở giao diện

Hàm UDP_Server_Load là hàm dùng để tạo thread mới để khởi động form server

2 references

```
public void InfoMessage(string info)
{
    if (lbMessages.InvokeRequired)
    {
        InfoMessageDel method = new InfoMessageDel(InfoMessage);
        lbMessages.Invoke(method, new object[] { info });
        return;
    }

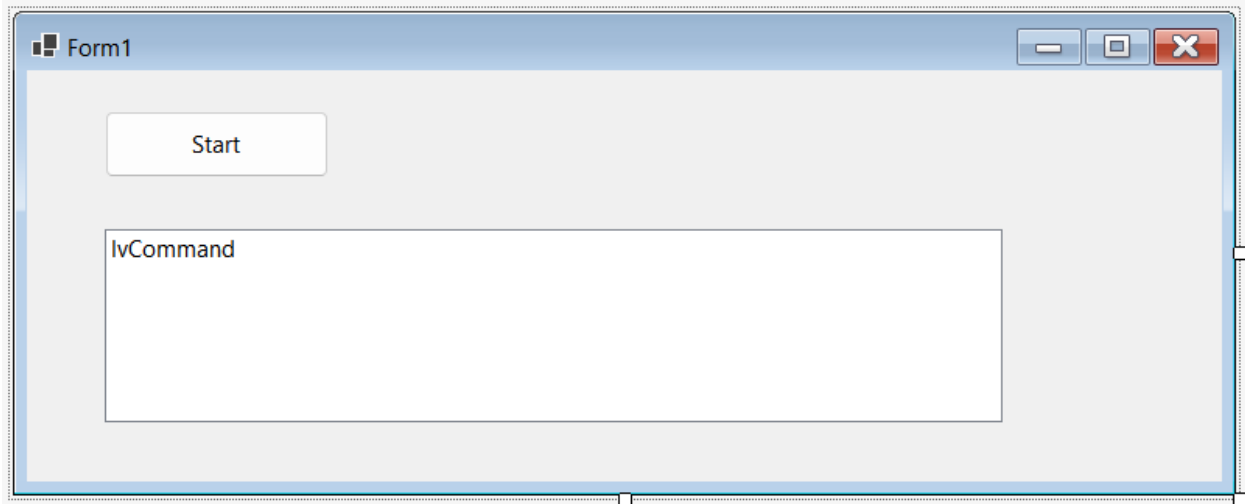
    lbMessages.Items.Add(info);
}
```

1 reference

```
private void UDP_Server_Load(object sender, EventArgs e)
{
    Thread thdUDPServer = new Thread(new ThreadStart(serverThread));
    thdUDPServer.Start();
}
```

Bài 02 –

Giao diện TCP Server (Tcp listener)



Hàm kích hoạt khi được click của button “Start”:

- Dùng để loại bỏ các lời gọi xuyên thread từ các control,...
- Dùng để tạo Unsafe Thread mới để mở server kết nối và truyền tham số là hàm StartUnsafeThread

1 reference

```
private void btStart_Click(object sender, EventArgs e)
{
    CheckForIllegalCrossThreadCalls = false;
    Thread serverThread = new Thread(new ThreadStart(StartUnsafeThread));
    serverThread.Start();
}
```

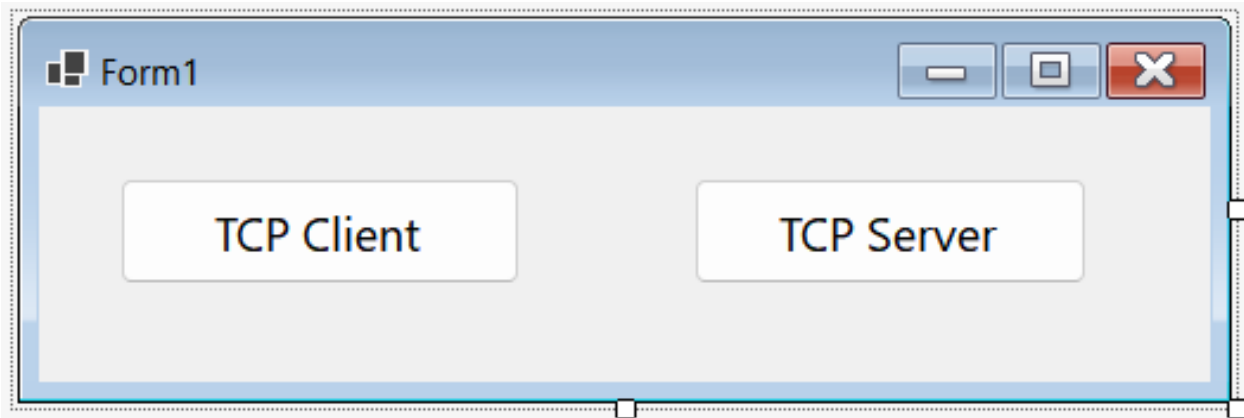
Hàm StartUnsafeThread(): dùng để mở kết nối TCP thông qua thiết lập listener socket và client socket. Từ đó khi socket được kết nối có thể truyền thông điệp qua lại với nhau và đẩy lên listbox lvCommand

```
void StartUnsafeThread()
{
    int bytesReceived = 0;
    byte[] recv = new byte[1];
    Socket clientSocket;
    Socket listenerSocket = new Socket(
        AddressFamily.InterNetwork,
        SocketType.Stream,
        ProtocolType.Tcp
    );

    IPEndPoint ipepServer = new IPEndPoint(IPAddress.Parse("172.30.199.13"), 8080);
    listenerSocket.Bind(ipepServer);
    listenerSocket.Listen(-1);
    clientSocket = listenerSocket.Accept();
    IPEndPoint remoteEndPoint = (IPEndPoint)clientSocket.RemoteEndPoint;
    lvCommand.Items.Add(new ListViewItem("New client connected: " + remoteEndPoint.Address + " " + remoteEndPoint.Port));
    while (clientSocket.Connected)
    {
        string text = "";
        do
        {
            bytesReceived = clientSocket.Receive(recv);
            text += Encoding.ASCII.GetString(recv);
        }
        while (text[text.Length - 1] != '\n');
        lvCommand.Items.Add(new ListViewItem(remoteEndPoint.Address + " (" + remoteEndPoint.Port + "): " + text));
    }
    listenerSocket.Close();
}
```

Bài 03 –

Giao diện của Dashboard3



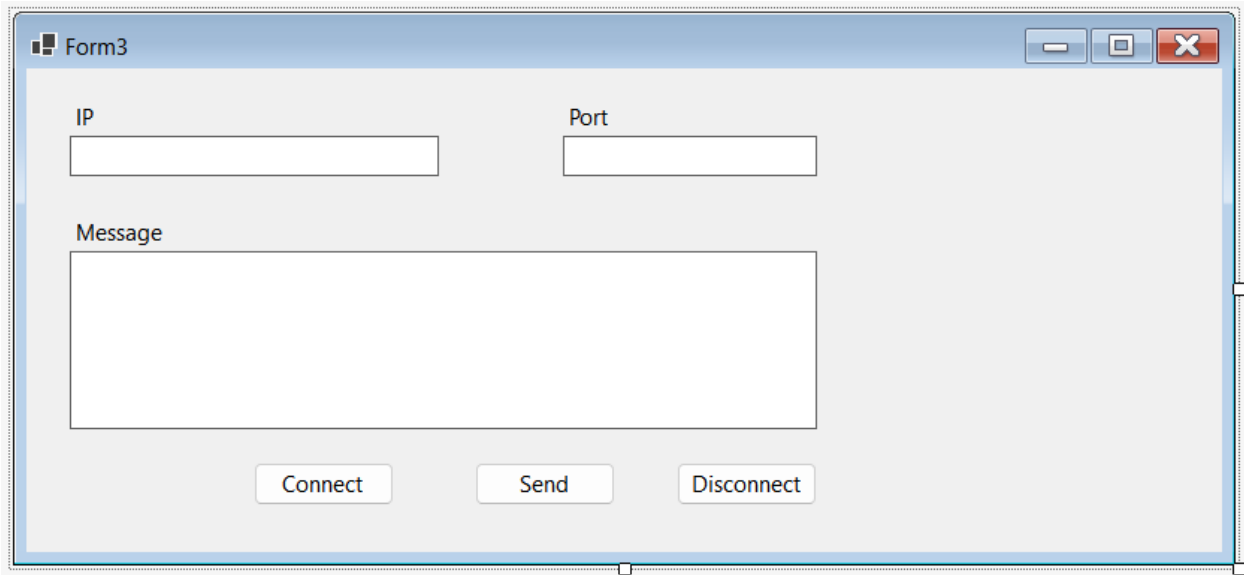
Các hàm khởi tạo form

```
Dashboard3.cs [Design]
Bai03
namespace Bai03
{
    3 references
    public partial class Dashboard3 : Form
    {
        1 reference
        public Dashboard3()
        {
            InitializeComponent();
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            var form = new TCPClient3();
            form.Show();
        }

        1 reference
        private void button2_Click(object sender, EventArgs e)
        {
            var form = new TCPServer3();
            form.Show();
        }
    }
}
```


Giao diện của client3



3 hàm kích hoạt khi được click của 3 button “Connect”, “Send”, “Disconnect”

```
13 namespace Bai03
14 {
15     3 references
16     public partial class TCPClient3 : Form
17     {
18         TcpClient tcpClient;
19         NetworkStream ns;
20
21         1 reference
22         public TCPClient3()
23         {
24             InitializeComponent();
25
26         private void btConnect_Click(object sender, EventArgs e)
27         {
28             try
29             {
30                 tcpClient = new TcpClient();
31                 IPEndPoint ipEndPoint = new IPEndPoint(IPAddress.Parse(tbIP.Text), 8080);
32                 tcpClient.Connect(ipEndPoint);
33                 ns = tcpClient.GetStream();
34             }
35             catch (Exception ex)
36             {
37                 MessageBox.Show(ex.Message);
38             }
39         }
```

```

40 private void btSend_Click(object sender, EventArgs e)
41 {
42
43     Byte[] data = System.Text.Encoding.ASCII.GetBytes(tbMessage.Text);
44     ns.Write(data, 0, data.Length);
45     ns.Flush();
46 }
47
48 1 reference
49 private void btDisconnect_Click(object sender, EventArgs e)
50 {
51     Byte[] data = System.Text.Encoding.ASCII.GetBytes("quit\n");
52     ns.Write(data, 0, data.Length);
53     ns.Close();
54     tcpClient.Close();
55 }
56 }
57

```

Giao diện của Server3

The screenshot shows a Windows Forms application window titled "Form2". The window contains a "Start Listen" button, a "Port" label with a text box containing "8080", and a large white rectangular area labeled "lvConnection" at the top left corner, which appears to be a list view or text area for displaying connections.

Hàm kích hoạt khi được click của button "Start Listen"

- Bỏ qua tất cả các lời gọi xuyên thread
- Khởi tạo thread

1 reference

```
public TCPServer3()
{
    InitializeComponent();
}
```

1 reference

```
private void btListen_Click(object sender, EventArgs e)
{
    CheckForIllegalCrossThreadCalls = false;
    Thread serverThread = new Thread(new ThreadStart(StartUnsafeThread));
    serverThread.Start();
}
```

Hàm StartUnsafeThread(): dùng để mở kết nối TCP thông qua thiết lập listener socket và client socket. Từ đó khi socket được kết nối có thể truyền thông điệp qua lại với nhau và đẩy lên listbox lvConnection

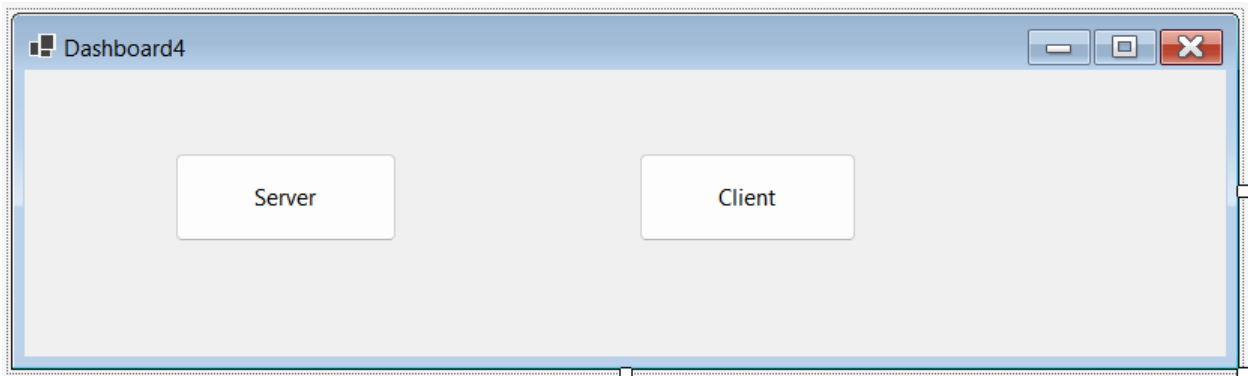
```
void StartUnsafeThread()
{
    try
    {
        int bytesReceived = 0;
        byte[] recv = new byte[1024];
        Socket clientSocket;
        Socket listenerSocket = new Socket(
            AddressFamily.InterNetwork,
            SocketType.Stream,
            ProtocolType.Tcp);
        IPEndPoint ipepServer = new IPEndPoint(IPAddress.Any, 8080);
        listenerSocket.Bind(ipepServer);
        listenerSocket.Listen(1);
        clientSocket = listenerSocket.Accept();
        IPEndPoint remoteEndPoint = (IPEndPoint)clientSocket.RemoteEndPoint;
        IPAddress remoteIPAddress = remoteEndPoint.Address;
        lvConnection.Items.Add(new ListViewItem(remoteIPAddress.ToString() + " connected"));
        while (clientSocket.Connected)
        {
            string text = "";
            bytesReceived = clientSocket.Receive(recv);
            text += Encoding.ASCII.GetString(recv);

            lvConnection.Items.Add(new ListViewItem(remoteIPAddress.ToString() + ": " + text));
            if (text == "quit")
            {
                clientSocket.Close();
                clientSocket.Disconnect(true);
                break;
            }
        }
        listenerSocket.Close();
        lvConnection.Items.Add(new ListViewItem(remoteIPAddress.ToString() + " disconnected"));
    }

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Bài 04 –

Giao diện của Dashboard4



Các hàm dùng để mở form con

```
11  namespace Bai04
12  {
13      3 references
14      public partial class Dashboard4 : Form
15      {
16          1 reference
17          public Dashboard4()
18          {
19              InitializeComponent();
20
21          private void btServer4_Click(object sender, EventArgs e)
22          {
23              var form = new Server4();
24              form.Show();
25          }
26
27          private void btClient4_Click(object sender, EventArgs e)
28          {
29              var form = new TicketDashboard4();
30              form.Show();
31          }
32      }
33  }
```

Giao diện của Server

Server04

Write to a File

Số lượng phim

Tên phim

Phòng chiếu ☐ P1 ☐ P2 ☐ P3

Giá vé chuẩn Nhập giá tiền (đơn vị: VND)

Số lần nhập còn lại Add

Thống kê Đặt vé

Các hàm để tạo thông tin phim (lab 2)

```
1 reference
22 public Server4()
23 {
24     InitializeComponent();
25     DataClient dataclient = new DataClient(8888);
26     dataclient.Start();
27     rtb1_Show.Text = @"Nhập thông tin của từng loại phim cho rạp( nếu muốn nhập mới)
28     thì ấn vào ""Write to a file"" " + '\n' + @"- Hoặc ấn ""Đặt vé""
29     để thực hiện đặt vé ngay và luôn vì file input5.txt đã có sẵn
30     data về phim (không cần nhập lại data)";
31 }
32 static int n = 0;
33 List<CPhim> Phims = new List<CPhim>();
34
35
36 private void bt1_WtF_Click(object sender, EventArgs e)
37 {
38     if (tbTotal.Text == "")
39     {
40         MessageBox.Show("Nhập số lượng phim vào Total");
41         tbTotal.Enabled = true;
42         return;
43     }
44 }
45
```

```

46 private void EnableInput()
47 {
48     tbName.Enabled = true;
49     cbRoom1.Enabled = true;
50     cbRoom2.Enabled = true;
51     cbRoom3.Enabled = true;
52     tbCost.Enabled = true;
53     btAdd.Enabled = true;
54 }

55 private void tbTotal_TextChanged(object sender, EventArgs e)
56 {
57     if (!Int32.TryParse(tbTotal.Text, out int temp))
58     {
59         MessageBox.Show(@"Nhập số nguyên vào thành ""Số lượng phim"": ");
60         return;
61     }
62     n = temp;
63     tbCount.Text = n.ToString();
64     MessageBox.Show("Nhập đầy đủ thông tin phim vào các ô phía dưới!! ");
65     EnableInput();
66 }

67 private bool CheckLoiNhap()
68 {
69     // ghi thông báo lỗi ở đây
70     return false;
71 }

72 private void btAdd_Click(object sender, EventArgs e)
73 {
74     try
75     {
76         CPhim Phim = new CPhim();
77         int num = Int32.Parse(tbCount.Text);
78         Phim.TenPhim = tbName.Text;
79         if (cbRoom1.CheckState == CheckState.Checked)
80         {
81             Phim.Phong.Add(new CPhim.CPhong { TenPhong = cbRoom1.Text.Trim() });
82         }
83         if (cbRoom2.CheckState == CheckState.Checked)
84         {
85             Phim.Phong.Add(new CPhim.CPhong { TenPhong = cbRoom2.Text.Trim() });
86         }
87         if (cbRoom3.CheckState == CheckState.Checked)
88         {
89             Phim.Phong.Add(new CPhim.CPhong { TenPhong = cbRoom3.Text.Trim() });
90         }
91         Phim.GiaVe = Int32.Parse(tbCost.Text);
92         tbCount.Text = (num - 1).ToString();
93         {
94             Phims.Add(Phim);
95             MessageBox.Show("Nhập thành công!!, còn lại " + tbCount.Text + " lần nhập");
96             tbName.Text = "";
97             tbCost.Text = "";
98             cbRoom1.CheckState = CheckState.Unchecked;
99             cbRoom2.CheckState = CheckState.Unchecked;
100             cbRoom3.CheckState = CheckState.Unchecked;
101             tbTotal.Enabled = false;
102         }
103     }

```

```

104         if (tbCount.Text == "0")
105         {
106             MessageBox.Show("Thuc hien day thong tin vao file ...", "Canh bao", MessageBoxButtons.OK);
107             SerializeJson(Phims, "input5.txt");
108             SerializeJson(Phims, "output5.txt");
109             //DisableInput();
110             //bt3_Read.Enabled = true;
111         }
112     }
113     catch (Exception ex)
114     {
115         MessageBox.Show(ex.Message);
116     }
117 }
118
119

```

```

120
121
122
123
124
125
126

```

1 reference

```

private List<CPhim> DeserializeJson(string Filepath)
{
    string json = File.ReadAllText(Filepath);
    List<CPhim> Phims = JsonSerializer.Deserialize<List<CPhim>>(json);
    return Phims;
}

```

```

127
128
129
130
131
132

```

2 references

```

private void SerializeJson(object obj, string Filepath)
{
    string json = JsonSerializer.Serialize(obj);
    File.WriteAllText(Filepath, json);
}

```

```

133
134
135
136
137
138

```

1 reference

```

private void btDatVe_Click(object sender, EventArgs e)
{
    var Form = new TicketDashboard4();
    Form.Show();
}

```

```

139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159

```

```

private static void RankPhim(List<CPhim> cphimList)
{
    for (int i = 0; i < cphimList.Count - 1; i++)
    {
        for (int j = 0; j < cphimList.Count - 1 - i; j++)
        {
            if (cphimList[j].TongTien < cphimList[j + 1].TongTien)
            {
                CPhim temp = cphimList[j];
                cphimList[j] = cphimList[j + 1];
                cphimList[j + 1] = temp;
            }
        }
    }

    for (int i = 0; i < cphimList.Count; i++)
    {
        cphimList[i].Rank = i + 1;
    }
}

```



```

160 private void bt3_Read_Click(object sender, EventArgs e)
161 {
162     progressBar1.Value = 0;
163     try
164     {
165         Phims = DeserializeJson("output5.txt");
166         RankPhim(Phims);
167         rtb1_Show.Clear();
168         int progressStep = 100 / Phims.Count;
169         foreach (var c in Phims)
170         {
171             progressBar1.Value += progressStep;
172             progressBar1.Refresh();
173
174             rtb1_Show.Text += "Ten phim: " + c.TenPhim + '\n';
175             double vetong = 0;
176             double veban = 0;
177             foreach (CPhim.CPhong c2 in c.Phong)
178             {
179                 vetong += 15;
180                 veban += c2.suat.Length;
181             }
182             rtb1_Show.Text += "So ve ban duoc: " + (vetong - veban).ToString() + '\n'
183                 + "So ve ton: " + veban.ToString() + '\n';
184             rtb1_Show.Text += "Ti le ve ban duoc: " + (((vetong - veban) / vetong) * 100).ToString("0.00")
185                 + "%\n" + "Doanh thu: " + c.TongTien.ToString() + '\n'
186                 + "Xep hang doanh thu: " + c.Rank + "\n\n";
187         }
188     }
189     catch (JsonException)
190     {
191         MessageBox.Show("File thông kê (output5.txt) hiện không có dữ liệu để trích xuất hoặc lỗi về Json ");
192     }
193     catch (Exception ex)
194     {
195         MessageBox.Show(ex.Message);
196     }
197 }

```

Các hàm thiết lập Listener, Gửi và nhận file info.txt (file thông tin phim)

```

198
199 3 references
200 public class DataClient
201 {
202     private TcpListener listener;
203     private const int BufferSize = 4096;
204
205     1 reference
206     public DataClient(int port)
207     {
208         listener = new TcpListener(IPAddress.Any, port);
209
210     }
211
212     public void Start()
213     {
214         Task.Run(() => RunServer());
215     }
216
217     private void RunServer()
218     {
219         listener.Start();
220         Console.WriteLine("Client-as-Server started. Waiting for connections...");
221
222         while (true)
223         {
224             TcpClient client = listener.AcceptTcpClient();
225             Task.Run(() => HandleClient(client));
226         }
227     }
228 }

```

```

226 private void HandleClient(TcpClient client)
227 {
228     Console.WriteLine($"Client connected: {((IPEndPoint)client.Client.RemoteEndPoint).Address}");
229
230     NetworkStream stream = client.GetStream();
231
232     // Send info.txt (serialized CPhim data) to client
233     SendFile("info.txt", stream);
234
235     // Receive file from client and save as info.txt
236     ReceiveFile("info.txt", stream);
237
238     // Close connection
239     stream.Close();
240     client.Close();
241 }
242
243 private void SendFile(string filepath, NetworkStream stream)
244 {
245
246     byte[] fileBytes = File.ReadAllBytes(filepath);
247     stream.Write(fileBytes, 0, fileBytes.Length);
248 }
249
250 private void ReceiveFile(string filepath, NetworkStream stream)
251 {
252
253     byte[] fileSizeBuffer = new byte[4];
254     stream.Read(fileSizeBuffer, 0, 4);
255     int fileSize = BitConverter.ToInt32(fileSizeBuffer, 0);
256     if (fileSize == 0)
257     {
258         return;
259     }
260     byte[] buffer = new byte[fileSize];
261     int bytesRead = 0;
262     while (bytesRead < fileSize)
263     {
264         int bytesToRead = Math.Min(fileSize - bytesRead, BufferSize);
265         int received = stream.Read(buffer, bytesRead, bytesToRead);
266         bytesRead += received;
267     }
268     File.WriteAllBytes(filepath, buffer);
269 }
270
271 }
272
273
274
275
276

```

Giao diện người dùng

The screenshot shows a Windows application window titled "Bai04_DatVe". Inside the window is a form titled "Thông tin vé". The form contains three input fields: "Họ tên:" (Last Name), "Phim:" (Movie), and "Phòng:" (Room). To the right of these fields is a section titled "Chọn chỗ ngồi" (Choose seat) which contains a list box with one item, "clb1_Seats". At the bottom of the form are three buttons: "Xác nhận" (Confirm), "Xóa" (Delete), and "Thoát" (Exit).

Các hàm khởi tạo

```
6 namespace Bai04
7 {
8     4 references
9     public partial class TicketDashboard4 : Form
10     {
11         List<CPhim> Phims = new List<CPhim>();
12         static CPhim SelectedPhim = new CPhim();
13         static CPhim.CPhong SelectedPhong = new CPhim.CPhong();
14         TcpClient client;
15         NetworkStream stream;
16
17         2 references
18         public TicketDashboard4()
19         {
20             InitializeComponent();
21
22             client = new TcpClient("localhost", 8888);
23             stream = client.GetStream();
24
25             // Start a thread for receiving files from the server
26
27             StartUnsafeThread();
28         }
29     }
```

Các hàm dưới dùng để thao tác trên giao diện gồm thiết lập phòng, phim, ghế ngồi, jsonserialize, jsondeserialize để giải quyết file truyền đi và nhận (lab2)

```
29 1 reference
30 public void StartUnsafeThread()
31 {
32     CheckForIllegalCrossThreadCalls = false;
33     Thread receiveThread = new Thread(ReceiveFile);
34     receiveThread.Start();
35 }

36 private List<CPhim> DeserializeJson(string Filepath)
37 {
38     string json = File.ReadAllText(Filepath);
39     List<CPhim> Phims = JsonSerializer.Deserialize<List<CPhim>>(json);
40
41     return Phims;
42 }
43

44 1 reference
45 private void SerializeJson(object obj, string Filepath)
46 {
47     string json = JsonSerializer.Serialize(obj);
48     File.WriteAllText(Filepath, json);
49 }

50 private void cb2_Movie_SelectionChangeCommitted(object sender, EventArgs e)
51 {
52     cb1_Theater.Items.Clear();
53     if (cb2_Movie.SelectedItem != null)
54     {
55         foreach (CPhim Phim in Phims)
56         {
57             if (Phim.TenPhim == cb2_Movie.SelectedItem.ToString())
58             {
59                 //MessageBox.Show("1");
60                 SelectedPhim = Phim;
61                 break;
62             }
63         }
64
65         if (SelectedPhim != null && SelectedPhim.Phong != null)
66         {
67             //MessageBox.Show("2");
68             foreach (CPhim.CPhong phong in SelectedPhim.Phong)
69             {
70                 cb1_Theater.Items.Add(phong.TenPhong);
71             }
72         }
73         cb1_Theater.Enabled = true;
74     }
75 }
76
```

```

77 private void cb1_Theater_SelectionChangeCommitted(object sender, EventArgs e)
78 {
79     clb1_Seats.Items.Clear();
80     if (cb1_Theater.SelectedItem != null && SelectedPhim != null && SelectedPhim.Phong != null)
81     {
82         //MessageBox.Show("3"); ;
83         foreach (CPhim.CPhong c in SelectedPhim.Phong)
84         {
85             if (c.TenPhong == cb1_Theater.SelectedItem.ToString())
86             {
87                 //MessageBox.Show("4"); ;
88                 SelectedPhong = c;
89                 break;
90             }
91         }
92
93         if (SelectedPhong.suat != null)
94         {
95             clb1_Seats.Items.AddRange(SelectedPhong.suat);
96         }
97     }
98 }
99
100

```

```

101 1 reference
102 private void clb1_Seats_ItemCheck(object sender, ItemCheckEventArgs e)
103 {
104     if (clb1_Seats.CheckedItems.Count == 1)
105     {
106         if (e.NewValue == CheckState.Unchecked)
107             bt1_Confirm.Enabled = false;
108     }
109     else
110         bt1_Confirm.Enabled = true;
111 }

```

```

112 private void bt1_Confirm_Click(object sender, EventArgs e)
113 {
114     long tong = 0; // tính tổng tiền từ checked seats
115     long cost = SelectedPhim.GiaVe;
116     foreach (string c in clb1_Seats.CheckedItems)
117     {
118         if (new[] { "A1", "A5", "B1", "B5", "C1", "C5" }.Contains(c))
119         {
120             tong += cost * 1 / 4;
121         }
122         else if (new[] { "A2", "A3", "A4", "C2", "C3", "C4" }.Contains(c))
123         {
124             tong += cost * 1;
125         }
126         else
127         {
128             tong += cost * 2;
129         }
130     }
131     string s = "Họ và tên: " + tb1_Name.Text;
132     s += System.Environment.NewLine + "Các vé đã chọn: ";
133     foreach (string c in clb1_Seats.CheckedItems)
134         s += c + " ";
135     s += System.Environment.NewLine;
136     s += "Phòng chiếu: " + cb1_Theater.Text;
137     s += System.Environment.NewLine;
138     s += "Số tiền phải trả: " + tong.ToString();
139
140     // thông báo lần cuối chắc chắn hay ko ?
141     if (MessageBox.Show(s, "Warning !!", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)
142     {
143         return;
144     }

```

```

145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
else
{
    MessageBox.Show("Bạn đã đặt vé thành công.", "Congratulations",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    SelectedPhim.TongTien += tong;
    HashSet<string> temp = SelectedPhong.suat.ToHashSet();
    foreach (string c in clb1_Seats.CheckedItems)
    {
        temp.Remove(c);
    }
    SelectedPhong.suat = temp.ToArray();
    foreach (var c in SelectedPhim.Phong)
    {
        if (c.TenPhong == SelectedPhong.TenPhong)
        {
            SelectedPhim.Phong.Remove(c);
            SelectedPhim.Phong.Add(SelectedPhong);
            break;
        }
    }
    foreach (var c in Phims)
    {
        if (SelectedPhim.TenPhim == c.TenPhim)
        {
            Phims.Remove(c);
            Phims.Add(SelectedPhim);
            break;
        }
    }
    SerializeJson(Phims, "output5.txt");
    SendFile("output5.txt");
    // blank các ô -> đẹp
    tb1_Name.Text = "";
    cb2_Movie.Text = "";
    cb1_Theater.Text = "";
    cb1_Theater.Enabled = false;
    SelectedPhim = null;
    SelectedPhong = new CPhim.CPhong();
    cb1_Theater.Items.Clear();
    clb1_Seats.Items.Clear();
}
}

```

1 reference

```

187
188
189
190
191
192
193
194
195
196
private void bt2_Reset_Click(object sender, EventArgs e)
{
    tb1_Name.Text = "";
    cb2_Movie.Text = "";
    cb1_Theater.Text = "";
    cb1_Theater.Enabled = false;
}

```

```

197 1 reference
198 private void bt3_Exit_Click(object sender, EventArgs e)
199 {
200     stream.Close();
201     client.Close();
202     this.Close();
203 }

```

Các hàm phía dưới đây là các hàm hỗ trợ kết nối và trao đổi thông điệp (lab3)

Hàm SendFile(string) dùng để ghi file và truyền file thông qua network stream

```

204 1 reference
205 private void SendFile(string filepath)
206 {
207     byte[] fileBytes = File.ReadAllBytes(filepath);
208     stream.Write(BitConverter.GetBytes(fileBytes.Length), 0, 4);
209     stream.Write(fileBytes, 0, fileBytes.Length);
210 }

```

Hàm ReceiveFile() dùng để nhận bất kì thông điệp nào từ phía server (mục tiêu là nhận file info.txt để triển khai dữ liệu trong file lên giao diện)

```

211 private void ReceiveFile()
212 {
213     try
214     {
215         while (true)
216         {
217             byte[] buffer = new byte[4096];
218             int bytesRead = stream.Read(buffer, 0, buffer.Length);
219             if (bytesRead > 0)
220             {
221                 string receivedFilepath = "received_tickets.txt";
222                 using (FileStream fileStream = new FileStream(receivedFilepath, FileMode.Create, FileAccess.Write))
223                 {
224                     fileStream.Write(buffer, 0, bytesRead);
225                 }
226                 UpdateDashboard(receivedFilepath);
227             }
228         }
229     }
230     catch (IOException ex) // thông báo lỗi/ tránh bị pause
231     {
232         MessageBox.Show("Error: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
233     }
234 }

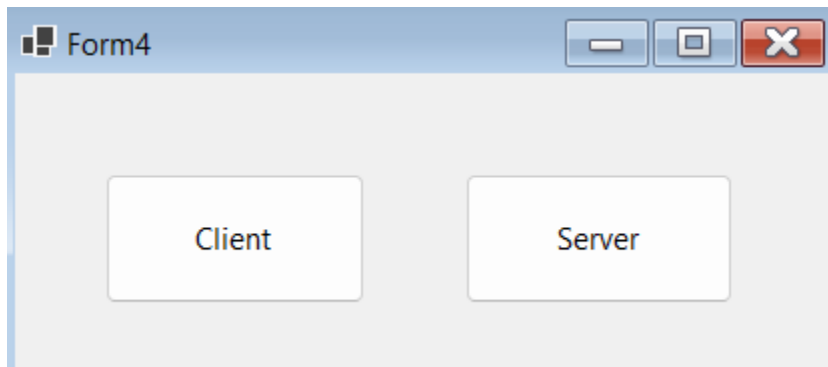
```

Hàm UpdateDashboard(string) dùng để cập nhật giao diện bằng file vừa nhận được từ server

```
235 private void UpdateDashboard(string filepath)
236 {
237     Phims = DeserializeJson(filepath);
238     try
239     {
240         foreach (var phim in Phims)
241         {
242             cb2_Movie.Items.Add(phim.TenPhim);
243         }
244     }
245     catch (Exception ex)
246     {
247         MessageBox.Show(ex.Message);
248     }
249 }
250
251 }
252
253 }
254
```

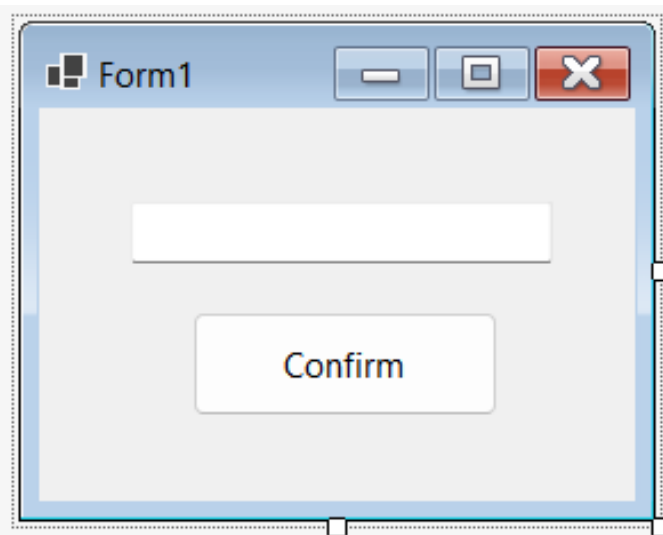

Bài 06 –

Giao diện của Dashboard6



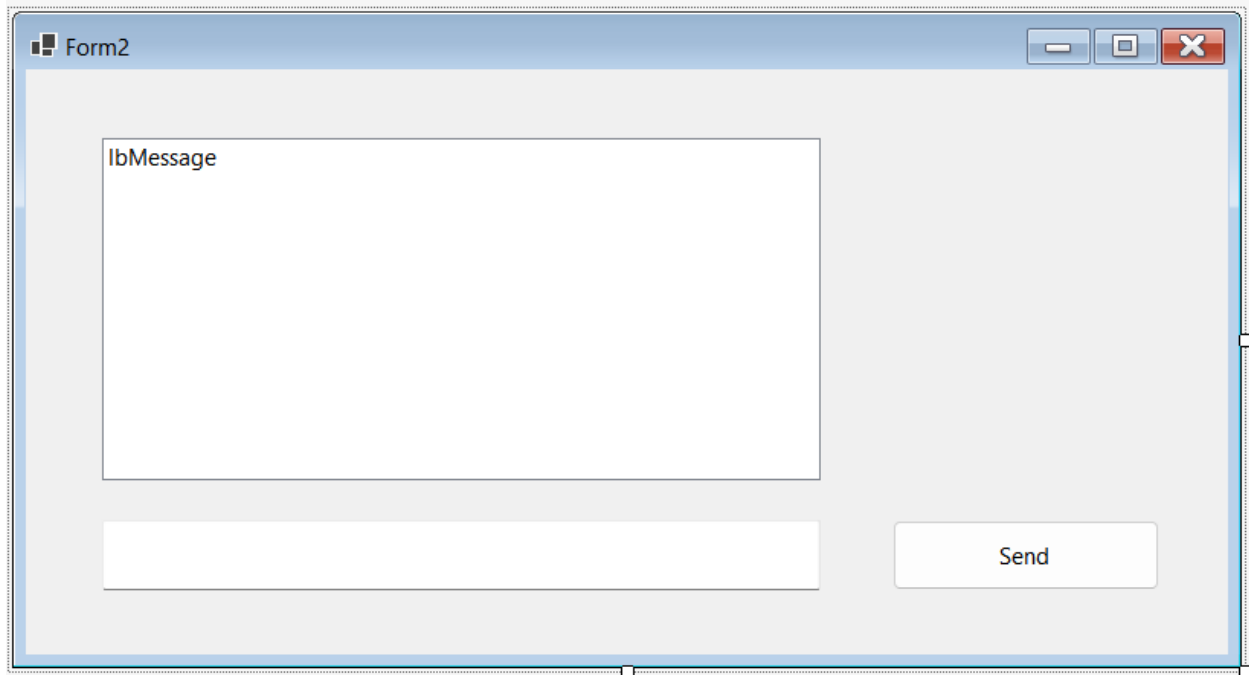
```
11 namespace Bai06
12 {
13     3 references
14     public partial class Dashboard6 : Form
15     {
16         1 reference
17         public Dashboard6()
18         {
19             InitializeComponent();
20
21         1 reference
22         private void btUser_Click(object sender, EventArgs e)
23         {
24             var form = new Form1();
25             form.Show();
26
27         1 reference
28         private void btServer_Click(object sender, EventArgs e)
29         {
30             var form = new Form3();
31             form.Show();
32         }
33     }
```

Giao diện nhập username của client



```
4 namespace Bai06
5 {
6     3 references
7     public partial class Form1 : Form
8     {
9         1 reference
10        public Form1()
11        {
12            InitializeComponent();
13
14        1 reference
15        private void tbConnect_Click(object sender, EventArgs e)
16        {
17            var form = new Form2(tbUsername.Text);
18            form.Show();
19        }
20    }
```

Giao diện nhắn tin của người dùng



```
8  namespace Bai06
9  {
10     4 references
11     public partial class Form2 : Form
12     {
13         const string SERVER_IP = "127.0.0.1";
14         const int PORT = 8888;
15         string user;
16         TcpClient client;
17         NetworkStream stream;
18         0 references
19         public Form2()
20         {
21             InitializeComponent();
```

```

22 1 reference
23 public Form2(string username)
24 {
25     InitializeComponent();
26     string user = username;
27     client = new TcpClient(SERVER_IP, PORT);
28     stream = client.GetStream();
29     Init(user);
30 }

```

```

31 public void StartUnsafeThread()
32 {
33     CheckForIllegalCrossThreadCalls = false;
34     var receiveThread = new System.Threading.Thread(() => ReceiveMessages(client));
35     receiveThread.Start();
36 }

```

```

37 1 reference
38 void Init(string username)
39 {
40     MessageBox.Show("Connected to server");
41
42
43     byte[] usernameBytes = Encoding.ASCII.GetBytes(username);
44     stream.Write(usernameBytes, 0, usernameBytes.Length);
45     stream.Flush();
46     Console.WriteLine("Type '@username message' to send a private message");
47     Console.WriteLine("Start typing your message:");
48
49     // Start receiving messages in a separate thread
50     StartUnsafeThread();
51
52     // Send messages from the console
53
54
55 }
56
57

```

```

58 private void btSend_Click(object sender, EventArgs e)
59 {
60     byte[] messageBytes = Encoding.ASCII.GetBytes(tbMessage.Text);
61     stream.Write(messageBytes, 0, messageBytes.Length);
62     stream.Flush();
63 }
64
65

```

```

66 1 reference
67 void ReceiveMessages(TcpClient client)
68 {
69     NetworkStream stream = client.GetStream();
70     byte[] buffer = new byte[1024];
71     int bytesRead;
72     while (true)
73     {
74         try
75         {
76             bytesRead = stream.Read(buffer, 0, buffer.Length);
77             if (bytesRead > 0)
78             {
79                 string received = Encoding.ASCII.GetString(buffer, 0, bytesRead)
80                 lbMessage.Items.Add(received);
81             }
82         } catch (Exception ex)
83         {
84             MessageBox.Show($"Error: {ex.Message}");
85             break;
86         }
87     }
88 }
89 }
90

```

Giao diện của server (tổng hợp tất cả tin nhắn của tất cả người dùng)

The screenshot shows a Windows application window titled "Form3". Inside the window, there is a "Start Listen" button on the left. To its right is a label "Port" above a text input field containing the value "8080". Below these elements is a large list view control labeled "lvConnection". The list view is currently empty.

```

10 namespace Bai06
11 {
12     3 references
13     public partial class Form3 : Form
14     {
15
16         static Dictionary<string, TcpClient> clients = new Dictionary<string, TcpClient>();
17         static TcpListener listener;
18         const int PORT = 8888;
19
20         1 reference
21         public Form3()
22         {
23             InitializeComponent();
24             listener = new TcpListener(IPAddress.Any, PORT);
25             listener.Start();
26             Console.WriteLine($"Server started on port {PORT}");
27             CheckForIllegalCrossThreadCalls = false;
28             // Run AcceptClients method on a separate thread
29             Task.Run(() => AcceptClients());
30         }
31
32         1 reference
33         void AcceptClients()
34         {
35             while (true)
36             {
37                 TcpClient client = listener.AcceptTcpClient();
38                 Thread clientThread = new Thread(() => HandleClient(client));
39                 clientThread.Start();
40             }
41         }

```

```

41 void HandleClient(TcpClient client)
42 {
43     string username = ((IPEndPoint)client.Client.RemoteEndPoint).Address.ToString();
44     clients.Add(username, client);
45     lvConnection.Items.Add($"Client {username} connected");
46
47     NetworkStream stream = client.GetStream();
48     byte[] buffer = new byte[1024];
49     int bytesRead;
50
51     while (true)
52     {
53         try
54         {
55             bytesRead = stream.Read(buffer, 0, buffer.Length);
56             if (bytesRead > 0)
57             {
58                 string message = Encoding.ASCII.GetString(buffer, 0, bytesRead);
59                 Console.WriteLine($"Received from {username}: {message}");
60
61                 if (message.StartsWith("@"))
62                 {
63                     string[] parts = message.Split(' ');
64                     string recipient = parts[0].Substring(1); // Remove '@' from recipient username
65                     string privateMessage = String.Join(" ", parts, 1, parts.Length - 1);
66                     SendPrivateMessage(username, recipient, privateMessage);
67                 }
68                 else
69                 {
70                     BroadcastMessage(username, message);
71                 }
72             }
73         }
74         catch (Exception ex)
75         {
76             Console.WriteLine($"Error: {ex.Message}");
77             break;
78         }
79     }
80
81     clients.Remove(username);
82     client.Close();
83     Console.WriteLine($"Client {username} disconnected");
84 }
85

```

Hàm BroadcastMessage() dùng để gửi tin nhắn cho tất cả clients kết nối với Server (thông qua clientId – địa chỉ IP)

```

86 1 reference
87 static void BroadcastMessage(string sender, string message)
88 {
89     foreach (var pair in clients)
90     {
91         TcpClient client = pair.Value;
92         NetworkStream stream = client.GetStream();
93         byte[] buffer = Encoding.ASCII.GetBytes($"[{sender}]: {message}");
94         stream.Write(buffer, 0, buffer.Length);
95     }
96 }

```

Hàm `SendMessagePrivate()` dùng để tạo kết nối với bên thứ ba và thực hiện trao đổi thông điệp

```
1 reference
97 static void SendMessagePrivate(string sender, string recipient, string message)
98 {
99     if (clients.ContainsKey(recipient))
100     {
101         TcpClient client = clients[recipient];
102         NetworkStream stream = client.GetStream();
103         byte[] buffer = Encoding.ASCII.GetBytes($"(Private) {sender}: {message}");
104         stream.Write(buffer, 0, buffer.Length);
105     }
106     else
107     {
108         Console.WriteLine($"User '{recipient}' not found.");
109     }
110 }
111 }
112 }
113 }
```


YÊU CẦU CHUNG

1) Đánh giá

- Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
- Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
- Nộp báo cáo kết quả chi tiết những đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (*nếu có*); giải thích cho quan sát (*nếu có*).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

2) Báo cáo

- File **.PDF** hoặc **.docx**. Tập trung vào nội dung, giải thích.
- Nội dung trình bày bằng Font chữ **Times New Romans/** hoặc font chữ của mẫu báo cáo này (UTM Avo)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: LabX_MSSV1_MSSV2. (trong đó X là Thứ tự buổi Thực hành).

Ví dụ: Lab01_21520001_21520002

- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT