

Trường đại học Công Nghệ Thông Tin -DHQG TPHCM

Khoa mạng máy tính và truyền thông

Báo cáo đồ án môn lập trình mạng căn bản

(Nhóm: 5)

Tên đồ án : Ứng dụng nghe nhạc từ xa

Danh sách thành viên:

Tên	Mã số sinh viên	Công việc
Lê Tuấn Lương	20521588	Làm báo cáo, hỗ trợ kết nối server.
Phạm Ngọc Lợi	20521560	Gửi file nhạc server đến client, các tính năng. (main)
Ngô Trần Thái Sơn	20521841	Kết nối server.
Phan Hữu Luân	20521585	Design client , server.
Phạm Văn Thời	20521977	Kết nối client, làm báo cáo.

Yêu cầu : các sinh viên viết một chương trình thực hiện công việc nghe nhạc online có sự giao tiếp giữa server và client

Báo cáo

A) Các công việc chính:

- . Thực hiện kết nối giữa server và client
- . Client và server giao tiếp được thông qua socket
- . Truyền file
- . Nghe nhạc
- . Cập nhật động nhạc lên server và client thành danh sách
- . Các chức năng mở rộng (hiển thị thông tin các sĩ, tìm kiếm, thông tin,)

B) Nguyên lý hoạt động :

1. Thực hiện kết nối giữa client và server
2. Server sẽ gửi về client danh sách bài hát
3. Sau khi client nhận được sẽ chọn một bài hát gửi về cho server

4. Server nhận được thông điệp từ client sẽ thực hiện xử lý thông tin từ tên bài hát của client gửi về thành link file nhạc
5. Server sau khi có link file nhạc sẽ thực hiện công việc gửi file về cho Client
6. Client nhận được file từ Server sẽ thực hiện phát nhạc ở phần nghe nhạc
7. Nhạc sẽ được phát ở phần nghe nhạc của chương trình
8. Người dùng có thể thực hiện các thao tác, chức năng của chương trình thông qua giao diện

C) Các tính năng của chương trình:

- Nghe nhạc
- Cập nhật
- Tìm kiếm
- Hiển thị danh sách bài hát giúp người dùng dễ dàng chọn bài hát
- Hiển thị thông tin ca sĩ
- Tua , tạm dừng, phát tiếp tục,...
- Hiển thị playlist gồm các bài hát mình đã chọn
- Nghe lại bài nhạc mình thích thông qua playlist
- Thoát

D) Code chương trình

1. Server:

`void Connect()` thực hiện kết nối với client thông qua port và Ipaddress

```

1 reference
void Connect()
{
    server.Bind(IP);

    Thread listen = new Thread(() =>
    {
        try
        {
            server.Listen(100);
            client = server.Accept();
            clientlist.Add(client);
            Thread receive = new Thread(Receive);
            receive.IsBackground = true;
            receive.Start(client);
            Console.ReadLine();
        }
        catch
        {
            IP = new IPEndPoint(IPAddress.Any, 9999);
            server = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.IP);
        }
    });
    listen.IsBackground = true;
    listen.Start();
}

```

`void Receive(Object obj)` : Nhận thông điệp đến từ Client nếu nhận thành công sẽ trả về tên bài hát thất bại thì sẽ hiển thị thông báo lỗi

```

1 reference
void Receive(Object obj)
{
    Socket client = obj as Socket;
    try
    {
        while (true)
        {
            byte[] data = new byte[1024 * 500000];
            client.Receive(data);
            message = (string)Deserialize(data);
            AddMessage(message);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Không thể nhận được message , Error : " + ex.Message);
    }
}

```

`void GetNameMusic()` lấy tên bài nhạc sau khi nhận từ client thực hiện đổi thành link path

```

void GetNameMusic()
{
    string[] files = Directory.GetFiles("D:/TRASH/DoAN/BackGround/ListSong/");
    for (int i = 0; i < files.Length; i++)
    {
        filenames[i] = Path.GetFileName(files[i]);
    }
}
// gọi từ đây để chỉnh sửa

```

`private void button1_Click(object sender, EventArgs e)` Sau khi có được link file bài nhạc cần gửi set chức năng cho button1 (Send) thực hiện thao tác gửi cho Client file bài hát được chọn và tên bài hát khác thông qua socket để Client thực hiện các thao tác

```

private void button1_Click(object sender, EventArgs e)
{
    Thread sendMP3 = new Thread(() =>
    {
        try
        {
            string filename = "";
            string nameMP3 = ".mp3";
            string fullname = (message + nameMP3).Trim();
            GetNameMusic();
            for (int i = 0; i < filenames.Length; i++)
            {
                if (fullname == filenames[i])
                {
                    filename = filenames[i]; // +nameMP3;
                    break;
                }
            }
            //string filename="" // = SwitchCase(ChooseMusic(message));
            string filepath = @"D:\TRASH\DoAN\BackGround\ListSong\\"; // đường dẫn
            byte[] filenamebyte = Encoding.ASCII.GetBytes(filename);
            //
            byte[] filedata = File.ReadAllBytes(filepath + filename);

            byte[] clientData = new byte[4 + filenamebyte.Length + filedata.Length];
            byte[] filenamelen = BitConverter.GetBytes(filenamebyte.Length);
            //
            filenamelen.CopyTo(clientData, 0);
            filenamebyte.CopyTo(clientData, 4);
            filedata.CopyTo(clientData, 4 + filenamebyte.Length);
            //
            client.Send(clientData);

            Console.ReadLine();
            message = "";
        }
    });
}

```

2. Client:

void ConnectionListSong() void loadTableListSong(), Kết nối tới database để hiển thị danh sách bài nhạc lên list :

```

void ConnectionListSong() // connect database
{
    comma = connect.CreateCommand();
    comma.CommandText = "select * from ListSong";
    datasql.SelectCommand = comma;
    tb.Clear();
    datasql.Fill(tb);
    dataGridView1.DataSource = tb;
    DesignColumns();
}

```

```

void loadTableListSong()
{
    try
    {
        connect = new SqlConnection(sourceStr);
        connect.Open();
        ConnectionListSong();
        connect.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Loi Ket Noi Den Database : " + ex.Message);
    }
}

```

void Connect(): Kết nối tới Server bằng IP của mình mà Port chỉ định của Server là 9999

```

//connect
IPEndPoint IP = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 9999);
Socket client = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.IP);
//kết nối server
1 reference
void Connect()
{
    try
    {
        client.Connect(IP);
    }
    catch
    {
        MessageBox.Show("không thể kết nối server!", "lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    Thread listen = new Thread(receive);
    listen.IsBackground = true;
    listen.Start();
}
//

```

void Send() Gửi tên bài hát cho server :

```

1 reference
void Send()
{
    if (textBox5.Text != string.Empty)
        client.Send(Serialize(textBox4.Text));
}

```

void receive() Nhận phản hồi từ server là link bài hát sau đó thực hiện các thao tác tiếp theo của chương trình:

```

void receive()
{
    try
    {
        while(true)
        {
            byte[] clientData = new byte[1024 * 500000];
            string receivePath = "D:/TRASH/DoAN/BackGround/RECEIVEDLIST/"; // đường dẫn nhận file mp3
            int receivedBytelen = client.Receive(clientData);
            //
            int filenamelen = BitConverter.ToInt32(clientData, 0);
            filename = Encoding.ASCII.GetString(clientData, 4, filenamelen);
            //
            BinaryWriter bwrite = new BinaryWriter(File.Open(receivePath + filename, FileMode.Append, FileAccess.Write));
            bwrite.Write(clientData, 4 + filenamelen, receivedBytelen - 4 - filenamelen);
            //
            bwrite.Close();
            Console.ReadLine();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Lỗi : " + ex.Message);
    }
}

```

private void textBox2_TextChanged_1(object sender, EventArgs e) Thực hiện thao tác tìm kiếm bài nhạc bằng tên bài hát hoặc tên ca sĩ:

```

1 reference
private void textBox2_TextChanged_1(object sender, EventArgs e)
{
    //sử dụng thuộc tính RowFilter để tìm kiếm theo tên "NameSong"
    string rowFilter = string.Format("{0} like '{1}'", "NameSong", "*" + textBox2.Text + "*");
    // (dataGridView1.DataSource as DataTable).DefaultView.RowFilter = rowFilter;
    string authorfilter = string.Format("{0} like '{1}'", "AuthorInfor", "*" + textBox2.Text + "*");
    (dataGridView1.DataSource as DataTable).DefaultView.RowFilter = authorfilter + " OR " + rowFilter;
}

```

private void iconButton5_Click(object sender, EventArgs e) Xử lý thông tin và thực hiện thao tác gửi link URL cho window media để phát nhạc qua giao diện nghe nhạc của chương trình

```

1 reference
private void iconButton5_Click(object sender, EventArgs e)
{
    string messagePath = "D:/TRASH/DoAN/BackGround/RECEIVEDLIST/" + filename;
    RunMusic.stringpath = messagePath;
    // RunMusic.idMusic = filename;
    for( int i =0;i < dataGridView1.Rows.Count-1; i++ )
    {
        if (filename == (dataGridView1.Rows[i].Cells[0].Value.ToString()+".mp3"))
        {
            RunMusic.nameMusic=dataGridView1.Rows[i].Cells[1].Value.ToString();
            break;
        }
    }
    // RunMusic.Show();
    ActivePlaying(sender, RGBcolors.color2);
    OpenChildForm(RunMusic);
}

```

private void button2_Click_1(object sender, EventArgs e) Hiển thị thông tin ca sĩ khi nhấn vào

```

1 reference
private void button2_Click_1(object sender, EventArgs e)
{
    int i = 0;
    i = dataGridView1.CurrentRow.Index;
    InforSinger inf = new InforSinger(dataGridView1.Rows[i].Cells[2].Value.ToString());
    inf.Show();
}

```

Các chức năng mở rộng của các form khác:

Hiển thị thông tin ca sĩ

```

1 reference
private void InforSinger_Load(object sender, EventArgs e)
{
    switch (ChooseSinger(DataSource))
    {
        case 1:
        {
            label3.Text = "Nguyễn Thanh Tùng";
            label7.Text = "5/7/1994";
            label8.Text = "Thái Bình City";
            label9.Text = "Sơn Tùng - MTP";
            break;
        }

        case 2:
        {
            label3.Text = "Nguyễn Đức Phúc";
            label7.Text = "15/10/1996";
            label8.Text = "Hà Nội Capital";
            label9.Text = "Đức Phúc";
            break;
        }

        case 3:
        {
            label3.Text = "Nguyễn Đức Thiện";
            label7.Text = "8/4/1991";
            label8.Text = "Hà Nội Capital";
            label9.Text = "Rhythmic";
            break;
        }

        case 4:
        {
            label3.Text = "Da Lab";
            label7.Text = "2007";
            label8.Text = "Northside";
            label9.Text = "Da Lab";
            break;
        }

        case 5:
        {
            label3.Text = "Nguyễn Vũ Hoàng Long";
            label7.Text = "2/3/1999";
            label8.Text = "Thái Bình City";
            label9.Text = "RPT ROCK, Long Ngự, YoungItachi..";
            break;
        }
    }
}

```

Nghe nhạc khi giao diện chính nhấn nút nghe của giao diện chính

```

1 reference
private void timer1_Tick(object sender, EventArgs e)
{
    try
    {
        x += a;
        label3.Location = new Point(x, y);
        if (x >= 910)
        {
            a = -1;
            //label3.ForeColor = Color.FromArgb(random.Next(0,255),random.Next(0,255), random.Next(0, 255));
        }
        if (x <= 731)
        {
            a = 1;
            //label3.ForeColor = Color.FromArgb(random.Next(0, 255), random.Next(0, 255),random.Next(0, 255));
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

E) Video demo của nhóm:

<https://drive.google.com/file/d/1ETCqbiv8BF4wESpclZF239dMxhl3522y/view?usp=sharing>

https://drive.google.com/file/d/1z_LqI44kHZHep4XmBNehOg5mCe53kaPe/view?usp=sharing

(link dự phòng).