

# 응 용 소 프 트 웨 어 실 습

## Homework #2



담당 교수님 : 이윤구 교수님

담당 조교님 : 윤여경 조교님

이훈민 조교님

제 출 날 짜 : 2018. 05. 03.

학 과 : 컴퓨터소프트웨어학과

학 번 : 2014707040

이 름 : 유 진 혁

항목 (+보고서 페이지) [x 는 작성하지 못한 경우]	슬라이드 번호	배점 (작성여부)
1. 1. 폼 구성 (server)	3	✓
1. 2. 폼 구성 (client) – (1)Main (2)Home (3)Search (4)Upload (5)MyPage	3	✓
2. 1. 서버, 클라이언트 연결	4	✓
3. 1. 회원가입	6	✓
3. 2. 로그인	8	✓
4. 1. Search Icon Picture Box Click – 가입된 전체 계정 List 보여주기	9	✓
4. 1. Search Icon Picture Box Click – 검색어로 계정 검색하기	10	✓
5. Upload Icon Picture Box Click – 게시물 업로드하기	10	✓
6. 1. MyPage Icon Picture Box Click – 프로필 정보 (프로필 사진, 자기소개, 게시물 개수)	11	✓
6. 2. MyPage Icon Picture Box Click – View 기능 구현	11	Δ
7. Home Icon Picture Box Click – 전체 게시물 최근 업로드된 순으로 보여주기		X
추가 0. MyPage Icon Picture Box Click – View 기능 2 가지 모두 구현한 경우		X
추가 1. MyPage Icon Picture Box – Profile Edit Button Click : 프로필 수정		X
추가 2. Search Icon Picture Box – listBox Item DoubleClick : 다른 계정 화면 보기		X

\* 전체 프로젝트 파일을 두 가지 버전으로 준비했습니다.

6. 2. View 기능을 리스트 모양으로 구현하였으나 가끔씩 클라이언트 로그인 시 에러가 발생하여, View 기능이 없는 프로젝트 파일도 같이 압축하여 보냅니다.

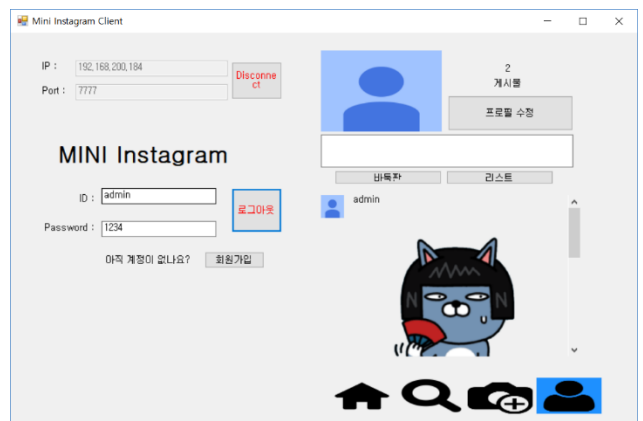
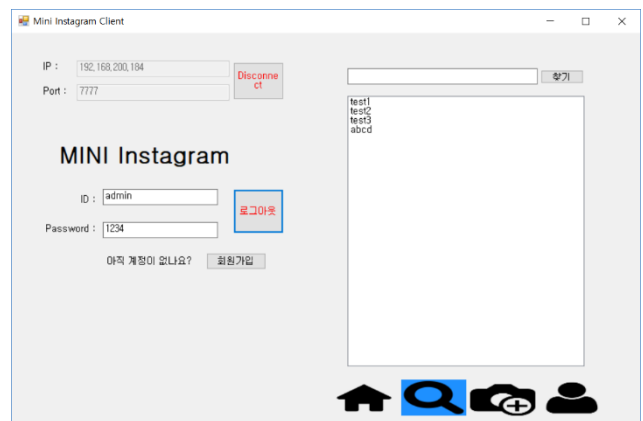
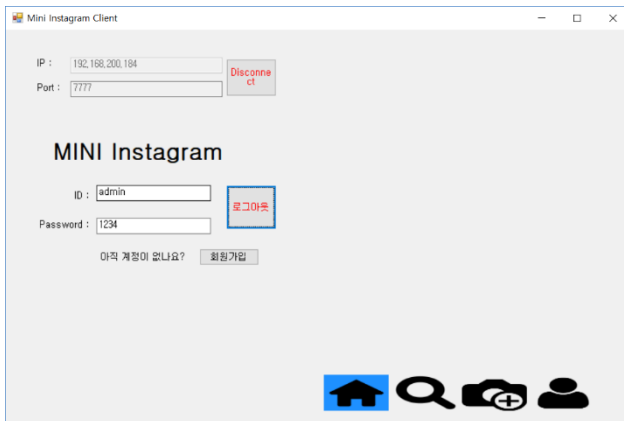
솔루션 파일 및 패킷 라이브러리는 각 Server 폴더 안에 있습니다.

## 1.1. 폼 구성 (server)

The screenshot shows the 'Mini Instagram Server' application window. At the top, there is a title bar with the text 'Mini Instagram Server' and standard window controls. Below the title bar, there is a section for IP and Port configuration. The IP field contains the text '실행시, 서버 PC의 IP를 자동으로 입력받아와서 textBox에 표시'. To the right of the IP field is a 'Port' field and a 'Start' button. Below this, there are two main panels: 'Member Account List' on the left and 'Server Log' on the right. The 'Member Account List' panel contains a table with three columns: 'Index', 'ID', and 'Password'. The 'Server Log' panel is currently empty.

## 1. 2. 폼 구성 (client) – (1)Main (2)Home (3)Search (4)Upload (5)MyPage

The screenshot shows the 'Mini Instagram Client' application window. At the top, there is a title bar with the text 'Mini Instagram Client' and standard window controls. Below the title bar, there is a section for IP and Port configuration. The IP field contains the text '서버PC IP 입력'. To the right of the IP field is a 'Port' field and a 'Connect' button. Below this, there is a large section titled 'MINI Instagram'. Under this title, there are two input fields: 'ID' and 'Password'. To the right of these fields is a '로그인' (Login) button. Below the login fields, there is a link '아직 계정이 없나요?' (Don't you have an account yet?) and a '회원가입' (Sign Up) button. At the bottom of the window, there is a navigation bar with four icons: a home icon, a search icon, a camera icon with a plus sign, and a user profile icon.



## 2. 1. 서버, 클라이언트 연결

```
/* 서버 IP 가져오기 */
private string GetServerIP()
{
    string ServerIP = "";
    IPEndPoint host = Dns.GetHostEntry(Dns.GetHostName());

    foreach (IPAddress ip in host.AddressList)
        if (ip.AddressFamily == AddressFamily.InterNetwork)
            ServerIP = ip.ToString();

    return ServerIP;
}
```

System.Net 에 정의되어 있는 클래스와 메소드를 이용하여 서버 PC 의 IP 주소를 받아오는 메소드를 작성하였다.

```
private void btnStart_Click(object sender, EventArgs e)
{
    if (txtIP.Text.Equals("") == true || txtPort.Text.Equals("") == true)
    {
        MessageBox.Show("IP와 Port를 입력하세요.");
        return;
    }
}
```

Start 버튼의 Click 핸들러 내용으로 IP/Port TextBox 가 하나라도 입력되어 있지 않으면 메시지박스가 뜨게 하였다.

```

_port = Convert.ToInt32(txtPort.Text);
_listener = new TcpListener(_port);
_listener.Start();
_isStart = true;

DirectoryInfo di = new DirectoryInfo(_storage);
if (di.Exists == false)
    di.Create();

LoadMemberList();

this.Invoke(new MethodInvoker(delegate ()
{
    txtLog.AppendText("Server - Start\n");
    txtLog.AppendText("Storage Path : C:\WWW\Instagram_server\WWW\n");
    txtLog.AppendText("waiting for client access...\n");
})));

while (_isStart)
{
    TcpClient client = _listener.AcceptTcpClient();

    if (client.Connected)
    {
        _isConnect = true;
        this.Invoke(new MethodInvoker(delegate ()
        {
            txtLog.AppendText("Client Access !!\n");
        })));
    }
}

```

입력된 포트 번호를 가져와 리스너를 만들고 지정된 공간에 서버 저장소 디렉토리를 만들게 하였다. 이때 디렉토리 유무를 검사하여 없을 때 만들게 한다. 그리고 가입된 계정을 추가하는 메소드를 호출한다. 메소드 내용은 아래와 같다.

```

this.Invoke(new MethodInvoker(delegate ()
{
    if (lvwMember.Items.Count > 0)
        lvwMember.Items.Clear();
})));

_fileReader = new StreamReader(_storage + "member.txt");

int index = 1;
while (_fileReader.Peek() >= 0)
{
    var lvwItem = new ListViewItem(new string[] { lvwMember.Columns.Count });
    for (int i = 0; i < lvwMember.Columns.Count; i++)
        lvwItem.SubItems[i].Name = lvwMember.Columns[i].Name;
    lvwItem.SubItems[0].Text = index.ToString();
    for (int i = 1; i < lvwMember.Columns.Count; i++)
        lvwItem.SubItems[i].Text = _fileReader.ReadLine();

    this.Invoke(new MethodInvoker(delegate ()
    {
        lvwMember.Items.Add(lvwItem);
    })));
    index++;
}

_fileReader.Close();
_fileReader.Dispose();

```

일단 ListView 에 이미 작성된 항목들이 있다면 전부 비우고 서버 저장소의 텍스트 파일을 읽어 들어와 이를 바탕으로 ListView 항목들을 만들어낸다. 이 항목들을 ListView 에 추가하고 이 작업을 해당 텍스트 파일을 다 읽을 때까지 반복한 후 파일을 닫는다. 이 텍스트 파일을 후에 회원가입을 할 때 회원 정보가 담기는 파일이다. 계정 정보를 추가했으면 Log txtBox 에 안내 문구를 쓰고 Client 의 연결을 기다렸다가 연결되면 연결을 알리는 텍스트를 쓴다.

```
if(btnConnect.Text == "Connect")
{
    if (txtIP.Text.Equals("") == true || txtPort.Text.Equals("") == true)
    {
        MessageBox.Show("IP와 Port를 입력하세요.");
        return;
    }

    Connect();

    if (_isConnect == false)
    {
        MessageBox.Show("Server가 시작되지 않았습니다.");
        return;
    }
}
```

클라이언트 측에선 Connect 버튼 클릭 시 각 상황에 맞는 에러 메시지박스가 뜨게 했고,

```
if (_isConnect)
{
    btnConnect.Text = "Disconnect";
    btnConnect.ForeColor = Color.Red;
    txtIP.Enabled = false;
    txtPort.Enabled = false;
    txtID.Enabled = true;
    txtPassword.Enabled = true;
    btnLogin.Enabled = true;
    btnJoin.Enabled = true;
}
```

연결되었을 경우, 버튼의 텍스트와 색깔, 그리고 다른 항목들의 Enable 상태가 바뀌도록 하였다.

### 3. 1. 회원가입

```
private void btnJoin_Click(object sender, EventArgs e)
{
    if (!_isConnect)
        return;

    Join join = new Join();
    join.Type = (int)PacketType.JOIN;
    join._id = txtID.Text;
    join._password = txtPassword.Text;

    Packet.Serialize(join).CopyTo(_sendBuffer, 0);
    Send();
}
```

회원 정보란에 입력된 정보로 Join 이라는 Packet 클래스를 상속받은 클래스의 객체를 만들고 객체의 필드 내용을 채워 서버로 전송한다.

```
case (int)PacketType.JOIN:
{
    _isError = false;
    _joinMember = (Join)Packet.Desserialize(_readBuffer);
    _fileReader = new StreamReader(_storage + "member.txt");
    while (_fileReader.Peek() >= 0)
    {
        if (_joinMember._id.Equals(_fileReader.ReadLine()))
        {
            Msg msg = new Msg();
            msg._text = "이미 사용중인 ID 입니다.";
            msg.Type = (int)PacketType.MSG;
            Packet.Serialize(msg).CopyTo(_sendBuffer, 0);
            Send();
            _isError = true;
            break;
        }
        _fileReader.ReadLine();
    }
    _fileReader.Close();
    _fileReader.Dispose();
    _fileReader = null;
}
```

Join 객체를 전달받은 서버는 계정 정보가 들어있는 텍스트 파일을 읽어 들여 해당 ID 가 이미 사용 중인지를 검사하고 에러 메시지를 다시 클라이언트에게 Packet 으로 전송한다.

```
if (_isError)
    break;
else
{
    _fileWriter = new StreamWriter(_storage + "member.txt", true);
    _fileWriter.WriteLine(_joinMember._id);
    _fileWriter.WriteLine(_joinMember._password);
    _fileWriter.Flush();
    _fileWriter.Close();
    _fileWriter.Dispose();
    _fileWriter = null;
    this.Invoke(new MethodInvoker(delegate ()
    {
        txtLog.AppendText(_joinMember._id + " >> Join\n");
    }));
    LoadMemberList();

    Msg msg = new Msg();
    msg._text = "회원가입이 정상적으로 완료되었습니다 !";
    msg.Type = (int)PacketType.MSG;
    Packet.Serialize(msg).CopyTo(_sendBuffer, 0);
    Send();
    break;
}
```

회원가입에 이상이 없다면, 계정 정보 텍스트 파일에 새 계정 정보를 입력하고 Log txtBox 에 회원가입을 알리는 텍스트를 쓴다. 그리고 처음에 호출하였던 ListView 를 로드하는 메소드를 호출시켜 계정 목록을 갱신한 뒤, 회원가입이 완료되었다는 메시지를 클라이언트에게 보낸다.

### 3. 2. 로그인

```
if (btnLogin.Text == "로그인")
{
    if (!_isConnect)
        return;

    Login login = new Login();
    login.Type = (int)PacketType.LOGIN;
    login._id = txtID.Text;
    login._password = txtPassword.Text;
    _myPost = new List<Post>();

    Packet.Serialize(login).CopyTo(_sendBuffer, 0);
    Send();
}
```

클라이언트 쪽에서 로그인 버튼을 클릭하면 입력된 계정 정보를 Packet 에 담아 서버로 전송한다. Login 은 Packet 의 상속을 받은 클래스이다.

```
try
{
    _isError = true;
    _loginMember = (Login)Packet.Desserialize(_readBuffer);
    _fileReader = new StreamReader(_storage + "member.txt");
    while (_fileReader.Peek() >= 0)
    {
        if (_loginMember._id.Equals(_fileReader.ReadLine()))
        {
            if (_loginMember._password.Equals(_fileReader.ReadLine()))
            {
                _isError = false;
                break;
            }
        }
        else
            _fileReader.ReadLine();
    }
    _fileReader.Close();
    _fileReader.Dispose();
    _fileReader = null;

    if (_isError)
    {
        Msg msg = new Msg();
        msg._text = "ID 또는 PW가 잘못되었습니다.ㄸ계정이 없다면 회원가입 버튼을 통해 계정을 만드십시오!";
        msg.Type = (int)PacketType.MSG;
        Packet.Serialize(msg).CopyTo(_sendBuffer, 0);
        Send();
        break;
    }
}
```

Packet 을 받은 서버는 계정 정보 텍스트 파일을 참고하여 ID 와 비밀번호를 비교하고 둘 다 맞다면 에러가 없음을, 틀린 것이 있다면 에러가 있다고 표시하고 에러메시지를 클라이언트에게 보낸다.



```
DirectoryInfo memberDi = new DirectoryInfo(_storage + _loginMember._id);
if (memberDi.Exists == false)
    memberDi.Create();
_loginMember._postnum = memberDi.GetFiles().Length / 2;

Msg msg = new Msg();
msg._text = "LogIn";
msg._memo = _loginMember._postnum.ToString();
msg.Type = (int)PacketType.MSG;
Packet.Serialize(msg).CopyTo(_sendBuffer, 0);
Send();
this.Invoke(new MethodInvoker(delegate ()
{
    txtLog.AppendText(_loginMember._id + " >> LogIn\r\n");
}));
```

여러가 없다면 그 계정의 정보를 담은 디렉토리를 서버 저장소에 만들어 주고 클라이언트에게 정상 로그인 됐음을 알려준다. 그리고 Log txtBox 에도 텍스트를 써준다.

#### 4. 1. Search Icon Picture Box Click – 가입된 전체 계정 List 보여주기

```
Member memberList = new Member();
string otherMember;
_fileReader = new StreamReader(_storage + "member.txt");
while (_fileReader.Peek() >= 0)
{
    otherMember = _fileReader.ReadLine();
    if (!_loginMember._id.Equals(otherMember))
        memberList._list += otherMember + "\r\n";
    _fileReader.ReadLine();
}
memberList.Type = (int)PacketType.MEMBER;
Packet.Serialize(memberList).CopyTo(_sendBuffer, 0);
Send();
_fileReader.Close();
_fileReader.Dispose();
_fileReader = null;
```

로그인과 동시에 본인 계정을 뺀 나머지 계정들의 ID를 담은 문자열을 패킷에 담아 클라이언트에 보낸다.

```
_memberList = (Member)Packet.Desserialize(_readBuffer);
_members = _memberList._list.Split('\r\n');
this.Invoke(new MethodInvoker(delegate ()
{
    foreach (var member in _members)
        lstSearch.Items.Add(member);
}));
break;
```

클라이언트는 이를 listBox 에 추가한다.

#### 4. 1. Search Icon Picture Box Click – 검색어로 계정 검색하기

```
private void btnSearch_Click(object sender, EventArgs e)
{
    lstSearch.Items.Clear();
    foreach(var member in _members)
    {
        if (member.Contains(txtSearch.Text))
            lstSearch.Items.Add(member);
    }
}
```

찾기 버튼을 누르면 입력된 문자열의 포함여부를 판단하여 listBox 를 새로 구성한다.

#### 5. Upload Icon Picture Box Click – 게시물 업로드하기

```
private void btnFind_Click(object sender, EventArgs e)
{
    openFileDialog.InitialDirectory = "C:\\\\";

    if(openFileDialog.ShowDialog() == DialogResult.OK)
    {
        txtPicturePath.Text = openFileDialog.FileName;
        picUploadPicture.ImageLocation = openFileDialog.FileName;
    }
}
```

```
private void btnUpload_Click(object sender, EventArgs e)
{
    if (txtPicturePath.Text == "" || picUploadPicture.ImageLocation == "")
        MessageBox.Show("업로드할 사진을 선택하세요.");
    else
    {
        Post post = new Post();
        post.Type = (int)PacketType.POST;
        post._image = picUploadPicture.Image.Clone();
        post._text = txtUploadText.Text;

        try
        {
            Packet.Serialize(post).CopyTo(_sendBuffer, 0);
            Send();
        }
        catch
        {
            MessageBox.Show("파일 용량이 너무 큼니다. (512KB 이하)");
        }
        _isNew = true;
    }
}
```

다이얼로그를 통해 파일을 선택하고 텍스트박스나 Picture 박스에 담긴 것이 없으면 오류 메시지박스를 띄우고 아니라면 해당 내용을 서버에 보낸다.

## 6. 1. MyPage Icon Picture Box Click – 프로필 정보 (프로필 사진, 자기소개, 게시물 개수)

```
pnIPost.Controls.Clear();

lblPostCounting.Text = _postnum.ToString();

_profileThumbnail = new PictureBox[_postnum];
_profileName = new Label[_postnum];
_postPicture = new PictureBox[_postnum];
_postText = new TextBox[_postnum];
```

마이페이지 아이콘을 누르면 새로 갱신할 것이 있을 경우에만 패널을 초기화 시키고 현재 서버 저장소의 파일 개수를 가져와 게시물 개수를 변경해준다.

## 6. 2. MyPage Icon Picture Box Click – View 기능 구현

```
int y = 0;
for (int i = _postnum - 1; i >= 0; i--)
{
    _profileThumbnail[i] = new PictureBox();
    _profileThumbnail[i].Image = Properties.Resources.profile_img;
    _profileThumbnail[i].Size = new Size(30, 30);
    _profileThumbnail[i].SizeMode = PictureBoxSizeMode.StretchImage;
    _profileThumbnail[i].Location = new Point(0, y);
    pnIPost.Controls.Add(_profileThumbnail[i]);

    _profileName[i] = new Label();
    _profileName[i].Text = txtID.Text;
    _profileName[i].Location = new Point(40, y);
    pnIPost.Controls.Add(_profileName[i]);

    try
    {
        _postPicture[i] = new PictureBox();
        _postPicture[i].Image = (Image)(_myPost[i]._image);
        _postPicture[i].Size = new Size(270, 180);
        _postPicture[i].SizeMode = PictureBoxSizeMode.Zoom;
        _postPicture[i].Location = new Point(40, y + 50);
        pnIPost.Controls.Add(_postPicture[i]);

        _postText[i] = new TextBox();
        _postText[i].Text = _myPost[i]._text;
        _postText[i].Size = new Size(270, 180);
        _postText[i].Location = new Point(40, y + 250);
        pnIPost.Controls.Add(_postText[i]);
    }
}
```

서버로부터 게시글의 정보를 받아 각각을 List 컬렉션에 담고 picture 박스와 텍스트 박스를 동적으로 생성하여 이들 담고 위치 및 크기를 조절해준다.

## 고찰.

너무도 어려운 과제였다. 패킷 통신 하는 부분에 있어 런타임 에러가 자주 발생하여 애를 먹었다. 그리고 파일을 쓰고 읽는 부분 또한 예상치 못한 에러가 계속 발생하여 디버깅을 하는데 많은 시간을 쏟아야 했다. 심지어는 디버깅할 때는 감지되지 않는 에러들이 막상 런타임시에, 그것도 간헐적으로 발생하여 원인을 찾지 못해 시간을 많이 허비하였다. 그럼에도 배운 내용들을 총망라하여 정리하고 디버깅 실력 또한 기를 수 있던 좋은 기회였다. 시간이 조금 더 있었으면 하는 아쉬움도 남는다.