

집에서 보고 싶은 영화 찾고싶니
그럼 사용해봐

영화 찾기 프로그램



CONTENTS

01

프로젝트 설명

02

프로젝트 일정

03

ER 다이어그램

04

프로젝트 흐름

05

주요 기능

06

화면 구현

01

프로젝트 설명

영화를 보고 싶을때 볼만한 영화를 찾는데 시간이 상당히 걸리고 그 영화의 정보를 보고 싶을 때 이러한 불편함을 개선하고 싶어 프로그램을 제작하였다. **api**를 이용하여 보고 싶은 영화를 찾기 위한 프로젝트이다.

개발 환경

02

프로젝트 일정

프로젝트 일정

2021-06-19

2021-06-21

2021-06-23

2021-06-25

2021-06-27

2021-06-29

2021-07-01

2021-07-03

프로젝트 계획

2021-06-24

2

2021-06-25

DB설계

2021-06-25

2

2021-06-26

화면설계

2021-06-27

2

2021-06-28

기능 및 구현 테스트

2021-06-28

3

2021-06-30

화면 구현 및 테스트

2021-06-29

2

2021-06-30

최종점검

2021-06-30

1

2021-06-30

03

ER 다이어그램

member_t						
멤버테이블						
PK	AI	FK	Null	Logical Name	Name	Type
✓		+		회원id	mem_id	VARCHAR2(20)
		+		회원password	mem_password	VARCHAR2(20)
		+		회원이름	mem_name	VARCHAR2(20)
		+		회원전화번호	mem_tel	VARCHAR2(20)
		+		회원생년월일	mem_birth	VARCHAR2(20)
		+		좋아하는장르	mem_Mgenre	VARCHAR2(20)
		+		싫어하는장르	mem_Hgenre	VARCHAR2(20)

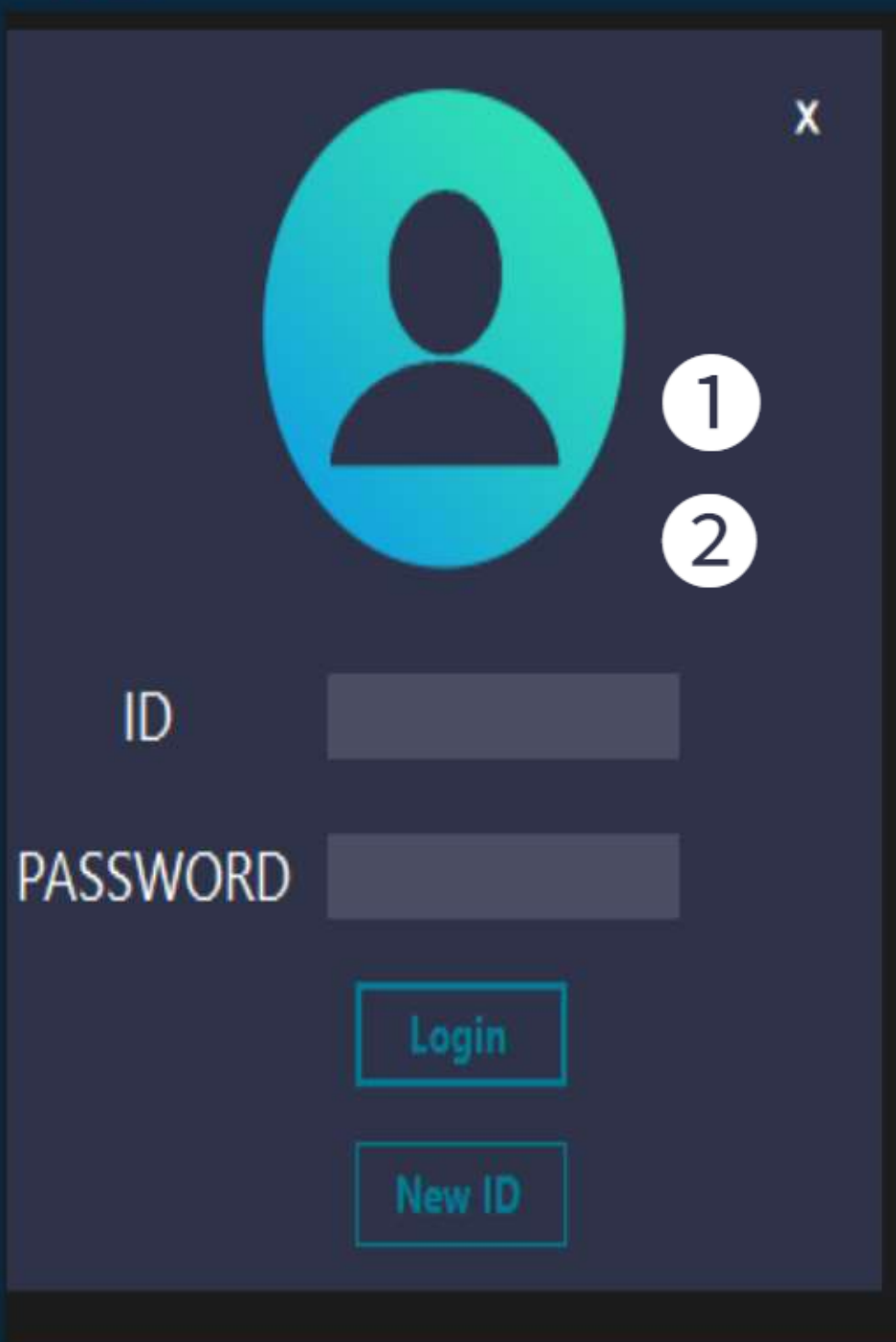
movie_t						
무비테이블						
PK	AI	FK	Null	Logical Name	Name	Type
		+		영화제목	movie_title	VARCHAR2(100)
		+		평점	starpoint	INT
		+	✓	감독	diretor	VARCHAR2(20)
		+	✓	배우	actor	VARCHAR2(20)
✓	✓	+		영화id	movie_id	INT

starpoint_t						
Table logical name						
PK	AI	FK	Null	Logical Name	Name	Type
✓	✓	+		별점id	starpoint_id	INT
		✓		회원id	mem_id	VARCHAR2(20)
		✓		영화id	movie_id	INT
		+		별점	starpoint	INT
		+		한줄평	evolation	VARCHAR2(20)

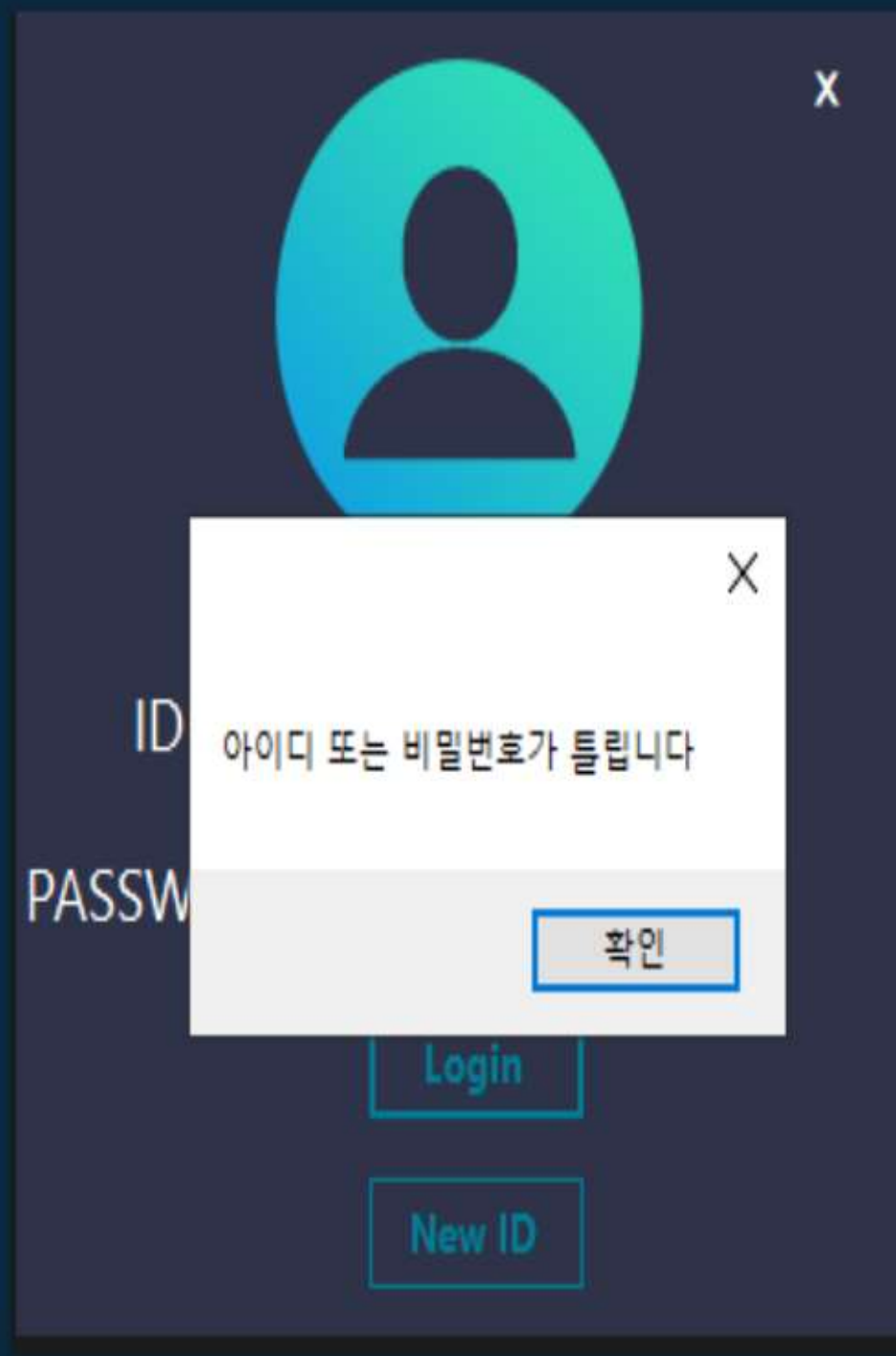
04

프로젝트 흐름

04 프로그램 흐름



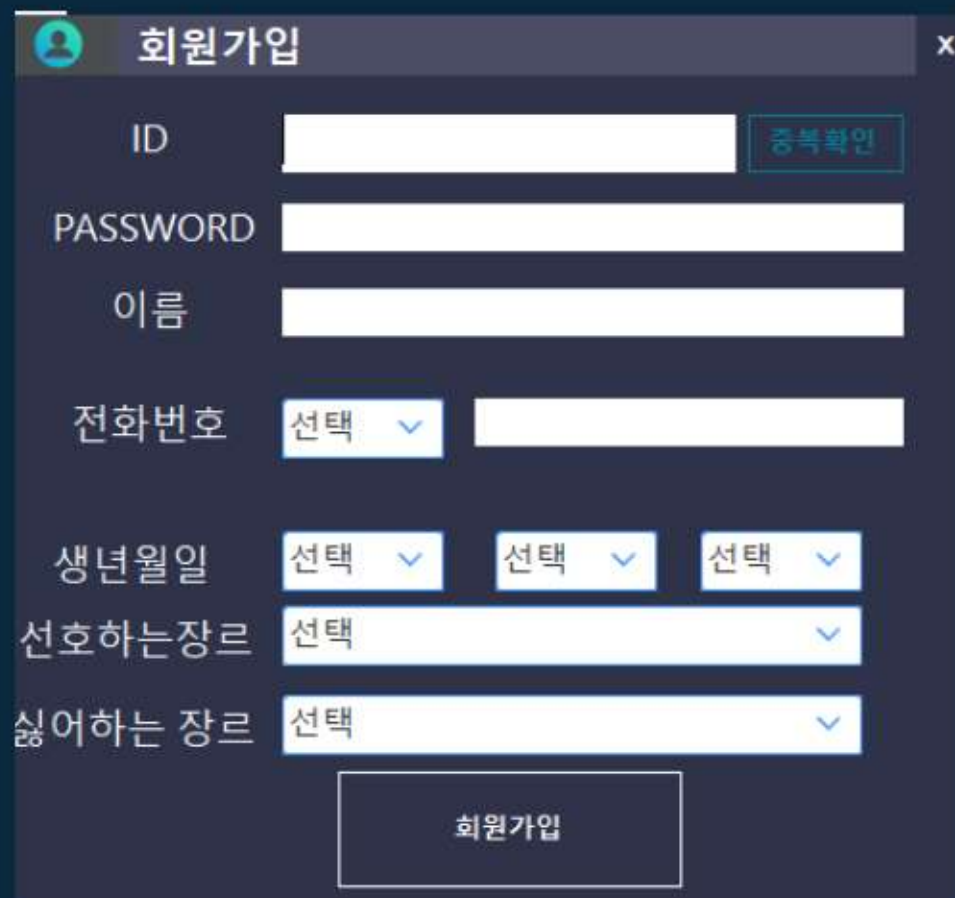
The image shows a login form with a dark blue background. At the top left is a circular profile picture placeholder. To its right are two white circles with black numbers '1' and '2'. Below the profile picture is an 'ID' label and a text input field. Below that is a 'PASSWORD' label and another text input field. At the bottom are two buttons: 'Login' and 'New ID'. A small 'X' icon is in the top right corner of the form.



The image shows the same login form as the previous one, but with a white error message dialog box in the center. The dialog box has a title bar with an 'X' icon and contains the text '아이디 또는 비밀번호가 틀립니다' (ID or password is incorrect). Below the text is a button labeled '확인' (Confirm). The background form is dimmed.

프로그램을 시작하면 로그인 창이 뜨는데 1번과 2번 텍스트 박스에 정확한 아이디와 비밀번호를 텍스트 박스에 넣고 로그인 버튼을 누르면 로그인이 됩니다. 맞지 않은 텍스트를 넣으면 메시지 박스가 나옵니다.

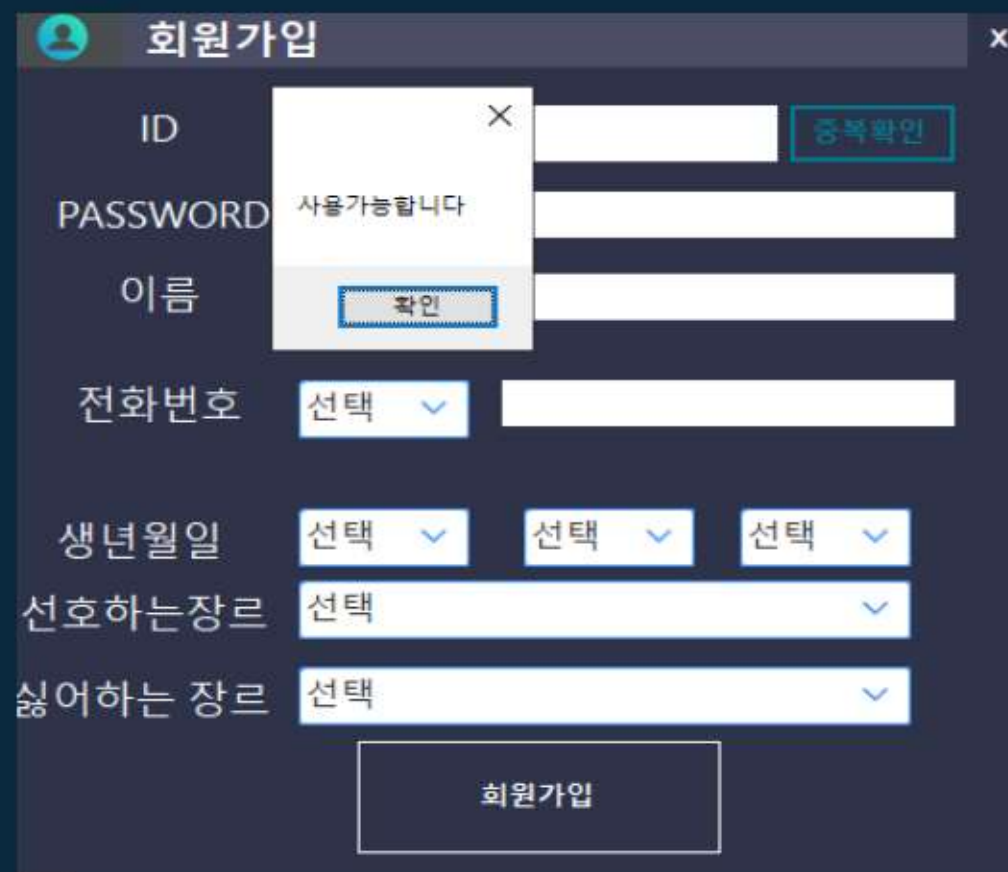
04 프로그램 흐름



회원가입 폼의 초기 상태. ID, PASSWORD, 이름, 전화번호, 생년월일, 선호하는장르, 싫어하는 장르 입력 필드가 있고 중복확인 버튼이 ID 옆에 있습니다.



회원가입 폼에서 ID 중복 확인 시 나타나는 경고 메시지. ID 입력란 옆에 '중복확인' 버튼이 있고, ID 텍스트에 '중복된 ID입니다'라는 메시지가 표시되어 있습니다.



회원가입 폼에서 ID 중복 확인 시 나타나는 경고 메시지. ID 입력란 옆에 '중복확인' 버튼이 있고, ID 텍스트에 '사용가능합니다'라는 메시지가 표시되어 있습니다.

로그인 창에서 newID버튼을 누르면 회원가입 창이 나오고 아이디 텍스트에 텍스트를 입력하고 중복확인 버튼을 누르면 DB에 저장된 아이디가 있으면 중복된 ID라는 메시지가 나오고 없으면 사용가능합니다 라는 메시지가 나옵니다

04 프로그램 흐름

회원가입

ID: rhkrndndud22 중복확인

PASSWORD: rhkr1218

이름: 곽우영

전화번호: 010 21916995

생년월일: 1996 12 18

선호하는장르: 코미디

싫어하는 장르: 공포

회원가입

회원가입

ID: rhkrndndud22 중복확인

PASSWORD: rhkr1218

이름: 곽우영

전화번호: 010

생년월일: 1996 12 18

선호하는장르: 코미디

싫어하는 장르: 공포

회원가입

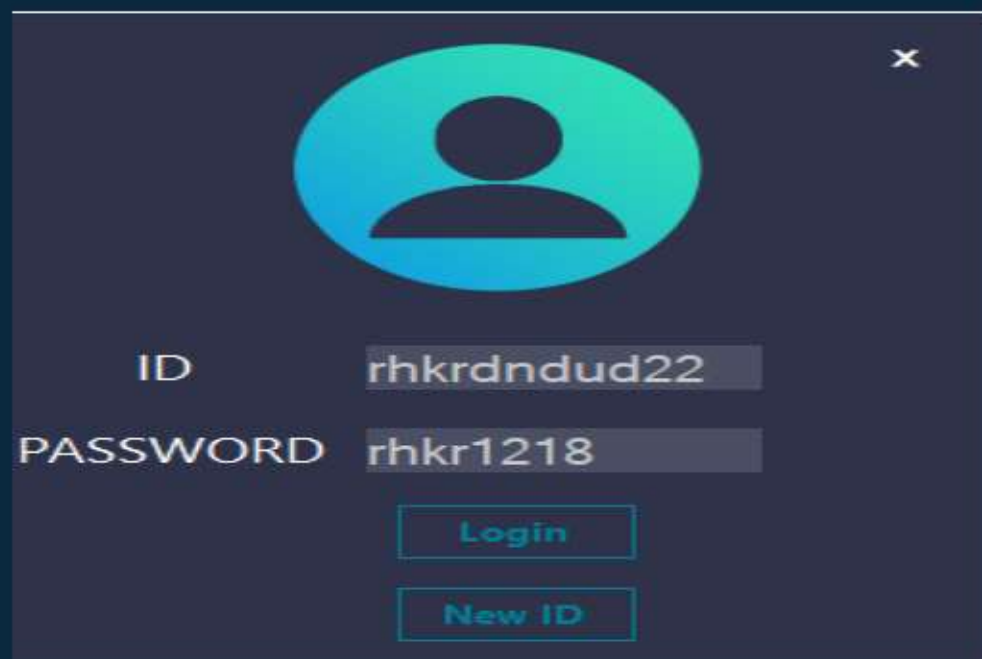
가입 해주셔서 감사합니다

확인

텍스트 박스를 빠짐없이 입력하고 회원가입 버튼을 누르면 메시지 박스가 나오면서 DB로 들어가게 됩니다.

	MEM_ID	MEM_PASSWORD	MEM_NAME	MEM_TEL	MEM_BIRTH	MEM_MGENRE	MEM_HGENRE
1	rhkrndndud22	rhkr1218	곽우영	01021916995	19961218	코미디	공포

04 프로그램 흐름



A login form with a dark blue background. At the top is a light blue circular icon with a white person silhouette. Below it, the text 'ID' is followed by a text input field containing 'rhkrdndud22'. Below that, 'PASSWORD' is followed by a text input field containing 'rhkr1218'. At the bottom are two buttons: 'Login' and 'New ID'.



정확한 아이디와 비밀번호를 입력하고 로그인 버튼을 누르면 홈화면이 나오고 1번 텍스트가 회원가 입창에 입력한 이름으로 설정 됩니다.

04 프로그램 흐름

영화제목

제목 감독이름
 부제목 출연진
 개봉일 평균평점

별점 별점과 한줄평 등록
 한줄평

영화제목	부제목	개봉일	감독	배우	평점
------	-----	-----	----	----	----

영화제목

제목 감독이름
 부제목 출연진
 개봉일 평균평점

별점 별점과 한줄평 등록
 한줄평

영화제목	부제목	개봉일	감독	배우	평점
설국열차	Snowpiercer	2013	봉준호	크리스 에반스...	7.98

설국열차

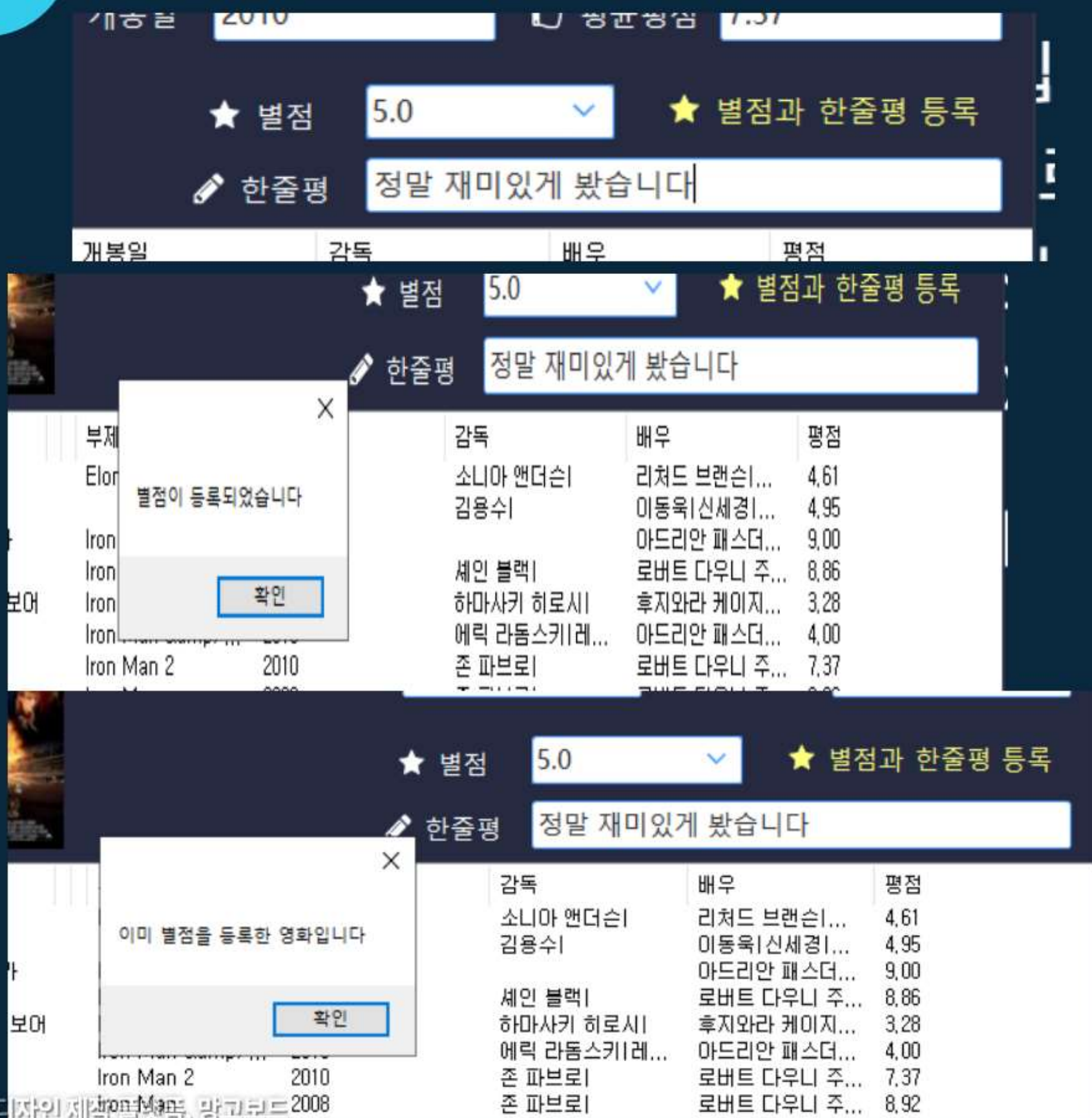
8 봉준호

크리스

1

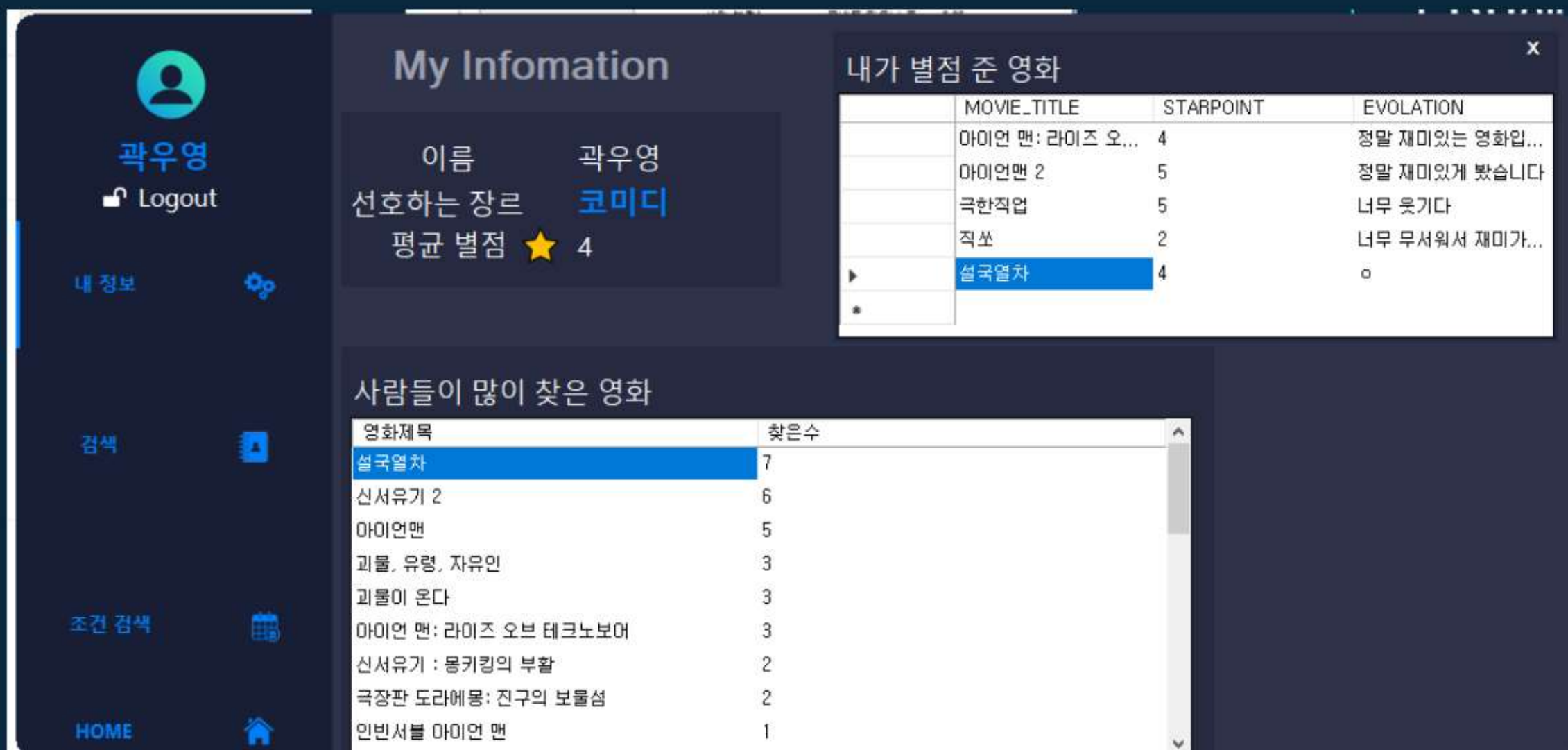
검색 버튼을 누르면 검색 폼으로 변환이 되고 버튼에 색이 바뀝니다. 영화제목 텍스트에 영화 제목을 입력하고 찾기 버튼을 누르면 그리드뷰에 영화가 나오고 누르면 이미지와 정보가 텍스트 박스에 들어가고 DB에 들어가게됩니다.

04 프로그램 흐름



별점 콤보 박스와 한줄
평 텍스트에 텍스트를
넣고 등록 버튼을 누르
게 되면 별점이 등록되
고 DB에 들어가게 되고
이미 등록한 영화면 이
미 등록한 영화라는 메
세지가 나옵니다

04 프로그램 흐름



내 정보 버튼을 누르면
내가 별점 준 영화들과
사람들이 많이 찾은 영
화들을 그리드 뷰에서
볼 수 있고 나의 평균 별
점과 좋아하는 장르도
나온다.

04 프로그램 흐름

조건 검색 버튼을 누르면
폼이 다시 바뀌고 텍스트에
조건을 다 주고
찾기 버튼을 누르면

The screenshot shows a web application interface for searching movies. On the left is a dark sidebar with a user profile icon and name '곽우영', a 'Logout' button, and three menu items: '내 정보' (My Info), '검색' (Search), and '조건 검색' (Conditional Search). The main content area has a search form at the top with fields for '연관 검색' (Associated Search), '장르' (Genre), '나라' (Country), '연도' (Year), and '연도' (Year) with a range selector '에서' (from) to '까지' (to). A blue '찾기' (Find) button is next to the form. Below the form is a table with columns: '영화제목' (Movie Title), '링..', '이..', '부제목' (Subtitle), '개봉일' (Release Date), '감독' (Director), '배우' (Actor), and '평점' (Rating). The table is currently empty.

This screenshot shows the same search interface as the previous one, but with filters applied. The '장르' (Genre) dropdown is set to '5: 로맨스' (5: Romance). The '나라' (Country) dropdown is set to 'KR 한국' (KR Korea). The '연도' (Year) dropdown is set to '1950' and the range selector is set to '2020'. The '찾기' (Find) button is still present. The table below remains empty.

04 프로그램 흐름

조건 검색 버튼을 누르면 폼이 다시 바뀌고 텍스트에 조건을 다 적어 준다.

The screenshot shows a web application interface for searching movies. On the left is a dark sidebar with a user profile icon and name '곽우영', a 'Logout' button, and three menu items: '내 정보' (My Info), '검색' (Search), and '조건 검색' (Conditional Search). The main content area has a search form at the top with fields for '연관 검색' (Associated Search), '장르' (Genre), '나라' (Country), '연도' (Year), and '연도' (Year) with a range selector '에서' (from) to '까지' (to). A blue '찾기' (Find) button is next to the form. Below the form is a table with columns: '영화제목' (Movie Title), '링..', '이..', '부제목' (Subtitle), '개봉일' (Release Date), '감독' (Director), '배우' (Actor), and '평점' (Rating). The table is currently empty.

The screenshot shows the same web application interface, but the search form is now filled out with specific conditions. The '장르' (Genre) dropdown is set to '5: 로맨스' (5: Romance). The '나라' (Country) dropdown is set to 'KR 한국' (KR Korea). The '연도' (Year) dropdown is set to '1950' and the range selector is set to '2020'. The '찾기' (Find) button is still present. The table below the form is still empty.

04 프로그램 흐름



찾기 버튼을 누르면 그
거에 맞는 검색 결과와
이미지가 나타나고 조건
을 바꿔주면 다른 영화
들이 나타나는 것을 볼
수 있다.



05

프로젝트 설명

05 프로그램 설명

```
class OracleHandler
{
    public static string movie_id;
    const string ORADB =
        "Data Source=(DESCRIPTION=(ADDRESS_LIST=" +
        "(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)" +
        "(PORT=1521)))" +
        "(CONNECT_DATA=(SERVER=DEDICATED)" +
        "(SERVICE_NAME=XE)))";
    "User Id=MODEL;Password=1234;";
    OracleConnection conn;
    OracleCommand cmd;

    public OracleHandler()
    {
        conn = new OracleConnection(ORADB);
        cmd = new OracleCommand();
        dbConnect();
    }

    public void dbConnect()
    {
        try
        {
            conn.Open();
        }
        catch (OracleException )
        {
        }
    }
}
```

먼저 오라클을 연결해주는 OracleHandler라는 클래스를 만들어 주고 이 클래스에서 OracleConnection을 이용하여 오라클DB를 연결해줍니다. 그러면 DB에 MODEL 사용자와 연결 되어 테이블을 사용할 수 있게 됩니다.

05 프로그램 설명

1

```
adapter.addMember(new Member(id, password, name, telF + telB,  
    (birthY + birthM + birthD), Mgenre, Hgenre));  
  
adapter.addReceiptDb();
```

2

```
namespace Dashboard.handler  
{  
    class MemberAdapter  
    {  
        private List<Movie> movieList = new List<Movie>();  
        private List<Star> starList = new List<Star>();  
        private List<Member> memberList = new List<Member>();  
        private OracleHandler ora;  
  
        public MemberAdapter(OracleHandler ora)  
        {  
            this.ora = ora;  
        }  
        public void addMember(Member member)  
        {  
            memberList.Add(member);  
        }  
    }  
}
```

4

```
public void insertdb(Member member)  
{  
    try  
    {  
        string query = string.Format("insert into member_t values" +  
            "('{0}','{1}','{2}','{3}','{4}','{5}','{6}']", member.Id, member.Pasword,  
            member.Name, member.Tel, member.Birth, member.Lgenre, member.Hgenre);  
        cmd.Connection = conn;  
        cmd.CommandText = query;  
        cmd.ExecuteNonQuery();  
        MessageBox.Show("가입 해주셔서 감사합니다");  
    }  
    catch(OracleException )  
    {  
        MessageBox.Show("아이디가 중복되어 가입이 불가능합니다");  
    }  
}
```

3

```
public void addReceiptDb()  
{  
    for (int i = 0; i < memberList.Count; i++)  
    {  
        ora.insertdb(memberList[i]);  
    }  
    memberList.Clear();  
}
```

그 다음 오라클을 연결해주는 adapter를 만들어 줍니다. 이곳에 DB테이블 정보들을 담은 리스트들을 만들어 주고 클래스들의 정보를 리스트에 넣는 메서드도 만들어 줍니다. 그리고 input 폼에서 정보를 받은 뒤 이것을 핸들러로 넘겨서 insert 문으로 테이블에 정보를 넣어줍니다.

1

```
private void Duplicate_Click(object sender, EventArgs e)
{
    string name = memID.Text;
    adapter.selectdb(name);
}
```

2

```
public void selectdb(string id)
{
    ora.selectid(id);
}
```

3

```
public void selectid(string id)
{
    string query = "select mem_id FROM member_T where mem_id='" + id + "'";

    cmd.Connection = conn;
    cmd.CommandText = query;

    OracleDataReader dr = cmd.ExecuteReader();

    if (dr.HasRows)
    {
        MessageBox.Show("중복된 ID입니다");
    }
    else
    {
        MessageBox.Show("사용가능합니다");
    }
}
```

아이디 중복 체크는 중복체크 버튼을 누르면 ID 텍스트를 변수에 넣어주고 이것을 어댑터 클래스에 넘겨주고 이것을 핸들러로 넘겨준 후 select문을 이용해 DB에서 찾아서 행이 있으면 중복된 ID라 가입을 못하고 없으면 사용가능하다는 메시지가 나오게 만들었습니다.

```

string query = "https://openapi.naver.com/v1/search/movie.xml?query=" + movieSearch.Text ;

string client_id = "XXXXXXXXXXXXXXXXXXXX";
string client_secret = "XXXXXXXXXXXX";
string your_client_id = client_id;
string your_client_secret = client_secret;

WebRequest wr = WebRequest.Create(query);
wr.Method = "GET";
wr.Headers.Add("X-Naver-Client-Id", your_client_id);
wr.Headers.Add("X-Naver-Client-Secret", your_client_secret);
WebResponse wrs = wr.GetResponse();
Stream s = wrs.GetResponseStream();
StreamReader sr = new StreamReader(s);
string response = sr.ReadToEnd();
XmlDocument xd = new XmlDocument();
xd.LoadXml(response);
XmlNode xn = xd["rss"]["channel"];

```

```

listView1.Items.Clear();
for (int i = 7; i < xn.ChildNodes.Count; i++)
{
    ListViewItem lvi = new ListViewItem();
    lvi.Text = xn.ChildNodes[i]["title"].InnerText.Replace("<b>", "").Replace("</b>", "");

    lvi.SubItems.Add(xn.ChildNodes[i]["link"].InnerText);
    lvi.SubItems.Add(xn.ChildNodes[i]["image"].InnerText);
    lvi.SubItems.Add(xn.ChildNodes[i]["subtitle"].InnerText);
    lvi.SubItems.Add(xn.ChildNodes[i]["pubDate"].InnerText);
    lvi.SubItems.Add(xn.ChildNodes[i]["director"].InnerText);
    lvi.SubItems.Add(xn.ChildNodes[i]["actor"].InnerText);
    lvi.SubItems.Add(xn.ChildNodes[i]["userRating"].InnerText);

    listView1.Items.Add(lvi);
    //richTextBox1.Text += xn.ChildNodes[i]["title"].InnerText.Replace("<b>", "").Replace("</b>", "") + "\n";
}

```

영화 검색 정보를 보여주기
 위해 네이버 검색 API 코드
 를 받아 온 뒤 텍스트 박스에
 입력 된 값을 찾은 후 그 정
 보를 리스트 뷰에 뿌려서 보
 여지게 작성하였다.


```
public DataTable selectpoint(string id)
{
    string query =
        "SELECT MOVIE_T.MOVIE_TITLE,Starpoint_T.Starpoint,Starpoint_t.evolution AS FROM MOVIE_T INNER JOIN STARPOINT_T ON MOVIE_T.MOVIE_ID =
        STARPOINT_t.MOVIE_ID WHERE starpoint_t.mem_ID='" + id + "'";

    cmd.Connection = conn;
    cmd.CommandText = query;
    OracleDataAdapter ada = new OracleDataAdapter();
    ada.SelectCommand = new OracleCommand(query, conn);
    DataTable dt = new DataTable();
    ada.Fill(dt);
    return dt;
}
```

```
DataTable dt=adapter.selectmovie();
Movieview.DataSource = dt;
```

```
DataTable da = adapter.selectpoint(name);
StarpointView.DataSource = da;
```

```
public DataTable selectmovie()
{
    string query = "select movie_title as 영화제목 ,count(movie_title) as 찾은수 from movie_T Group By movie_title order by 찾은수 desc";
    cmd.Connection = conn;
    cmd.CommandText = query;
    OracleDataAdapter ada = new OracleDataAdapter();
    ada.SelectCommand = new OracleCommand(query,conn);
    DataTable dt = new DataTable();
    ada.Fill(dt);
    return dt;
}
```

Info폼에 그리드 뷰
에 정보를 뿌리기 위
해 쿼리문을 이용하
여 정보를 찾은 뒤
그 쿼리를 데이터 테
이블에 넣은 뒤 리턴
하여 그 정보를 리턴
하여 정보들을 보여
주게 만들었다.

```

string title = movietitle.Text;
string evol = evol.Text;
if (title != "")
{
    [지역 변수] string title

    if (evol != "")
    {
        if (starPBox1.SelectedText != "")
        {
            double point = Convert.ToDouble(starPBox1.SelectedText);

            string id = star_id;
            adapter.addstar(new Star(id, title, point, evol));
            adapter.addstardb(id);
        }
        else
        {
            MessageBox.Show("별점 선택해주세요");
        }
    }
    else
    {
        MessageBox.Show("한줄평을 적어주세요");
    }
}
else
{
    MessageBox.Show("영화를 선택해주세요");
}

```

```

public void addstardb(string name)
{
    for (int i = 0; i < starList.Count; i++)
    {
        ora.insertstarpoint(starList[i], name);
    }

    starList.Clear();
}

```

별점 또한 영화 제목
과 한줄평과 별점을
선택하지 않으면
예외처리를 하게 작
성하였고 다 작성을
하면 어댑터로 넘겨
준 뒤 그 정보들을
담은 star클래스를
핸들러로 넘겨줍니
다.


```
public void insertstarpoint(Star star, string id)
{
    string query = string.Format("SELECT movie_id FROM movie_T where movie_title = '{0}'"+
        "INTERSECT SELECT movie_id FROM starpoint_t where mem_id = '{1}'", movie_id, id);
    cmd.Connection = conn;
    cmd.CommandText = query;
    OracleDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows)
    {
        MessageBox.Show("이미 별점을 등록한 영화입니다");
    }
    else
    {
        string query1 = string.Format("insert into starpoint_t values(STARPOINT_T_SEQ.NEXTVAL," + "'{0}'," +
            + "(select movie_id from movie_t where movie_title = '{1}' and rownum=1)," + " {2}, '{3}']",
            star.Mem_id, movie_id, star.Point, star.Evaluation);
        cmd.Connection = conn;
        cmd.CommandText = query1;

        cmd.ExecuteNonQuery();
        MessageBox.Show("별점이 등록되었습니다");
    }
}
```

핸들러에서 먼저
select문을 이용
하여 DB에 같은
ID로 저장된 영화
가 있는지 찾은 후
없으면 DB에 등
록하게 작성하였
습니다.

06

화면 구현

```

139 cmd.Connection = conn;
140 cmd.CommandText = query;
141 OracleDataAdapter ada = new OracleDataAdapter();
142 ada.SelectCommand = new OracleCommand(query, conn);
143 DataTable dt = new DataTable();
144 ada.Fill(dt);
145 return dt;
146 }
147 public void loginid(string id,string password)
148 {
149
150     string query= string.Format("select mem_id,mem_password from member_t where mem_id = '{0}' and mem_password = '{1}'",id,password);
151
152     cmd.Connection = conn;
153     cmd.CommandText = query;
154     OracleDataReader dr = cmd.ExecuteReader();
155
156     if (dr.HasRows)
157     {
158         Form1 f1 = new Form1();
159         f1.Show();
160     }
161     else
162     {
163         MessageBox.Show("아이디 또는 비밀번호가 틀립니다")
164     }
165 }
166
167 public void insertmovie(Movie movie)
168 {
169     string query = string.Format("insert into movie_t va
170     "('{0}','{1}','{2}','{3}',", movie.Title, mc
171     cmd.Connection = conn;
172     cmd.CommandText = query;
173     cmd.ExecuteNonQuery();
174

```

OWIE_T_SEQ.NEXTVAL)";

문제가 검색되지 않음

감사합니다.