Web Services Stage 2 Report

Banner IDs: B00294711 B00294646

## Prerequisite

In order to use our Web Services System, the following things need to be done in advanced:

1. Install MySQL version 5.6 or above.

2. Install Visual Studio 2010 or above, with Microsoft Chart Controls for Microsoft .NET Framework 3.5(The [installation package](http://www.baidu.com/link?url=4g_a4x5zYooFk4TEDRNWR3CHUNjbhjn-3c1HrokKGcRgNyawaQYis5diKRuqwneGxAe-rKbBEkFuGh9VacesGb0iqHO6-fbg_Zay3o61lM_BHm_ecenw0y8y-cawFfox) is available among the files attached).

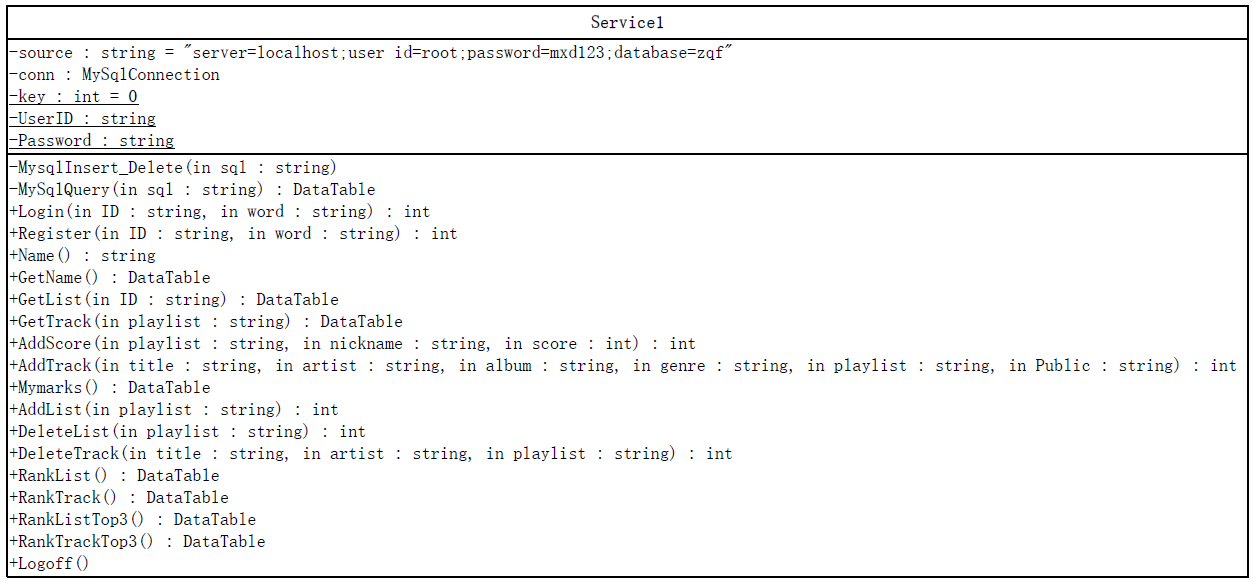
3. [Configure](http://www.baidu.com/link?url=oy0In_XhArWRQuzot58ahTrshnvvtgq-8vu5g_n1erDjyEuz8qD_fkb7KBhcQYucaD2B5q3kWHvpyzf007Ct4OVN_5dZsb7y1TBwOhnN6Du) MySQL as what the attached document file “mysql configuration.txt“ wrote.

## Server

The server in our Web Services System has two missions. The first one is to convey information from client to MySQL databases and vice versa. The second one is to protect information in the databases. To be more specific, our server protect users’ privilege and privacy.

## Class diagrams

The class diagram of our server is below:



Members marked as “–“ are Private methods, for internal class use.

Members marked as “+ “are Public web methods.

## Service Code

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Services;

using MySql.Data.MySqlClient;// to load this namespace, MySql.Data.dll is needed.

using System.Data;

namespace WebService5

{

[WebService(Namespace = "http://functional.com")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

// [System.Web.Script.Services.ScriptService]

public class Service1 : System.Web.Services.WebService

{

private string source = "server=localhost;user id=root;password=mxd123;database=zqf";

// The class which is used to connect mysql

private MySqlConnection conn;

//static variable, which can be used through out the process.

private static int key = 0;

//storage Users' identity information

private static String UserID;

private static String Password;

//Private method by which server can insert or delete data from mysql databases

private void MysqlInsert\_Delete(String sql)

{

try

{

//creae MySqlConnection object;

conn = new MySqlConnection(source);

conn.Open();

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataReader reader = cmd.ExecuteReader();

conn.Close();

}

catch (MySql.Data.MySqlClient.MySqlException ex)

{

throw ex;

}

}

//Private method by which server can query data from mysql databases

private DataTable MySqlQuery(String sql)

{

// string source = "server=localhost;user id=root;password=mxd123;database=zqf";

DataTable dt = new DataTable();

try

{

conn = new MySqlConnection(source);

conn.Open();

MySqlCommand cmd = new MySqlCommand(sql, conn);

MySqlDataReader reader = cmd.ExecuteReader();

dt.Load(reader);

conn.Close();

}

catch (MySql.Data.MySqlClient.MySqlException ex)

{

throw ex;

}

return dt;

}

//This method check whether users had registered or not, and give respond. if Yes, storage users' userid and password in order to authorize users.

[WebMethod]

public int Login(String ID, String word)

{

String str = "select \* from register where UserID ='" + ID + "' and Password='" + word + "';";

DataTable dt = MySqlQuery(str);

if (dt.Rows.Count == 1&&ID!=""&&word!="")

{

key = 1;

UserID = ID;

Password = word;

}

return key;

}

//register a new account,and authorize users. If the ID has been occupied, return 0.

[WebMethod]

public int Register(String ID, String word)

{

String str1 = "select \* from register where UserID ='" + ID+"';";

DataTable dt = MySqlQuery(str1);

int num = dt.Rows.Count;

if (num == 1||ID==""||word=="")

{

return 0;

}

else

{

String str2 = "insert into register values('" + ID + "','" + word + "');";

MysqlInsert\_Delete(str2);

UserID = ID;

Password = word;

key = 1;

return key;

}

}

//if tbe users have been authorized, return users' ID.

[WebMethod]

public String Name()

{

if (key == 1)

{

return UserID;

}

else

{

return "";

}

}

//return the list of all users.

[WebMethod]

public DataTable GetName()

{

DataTable dt = new DataTable();

if (key == 1)

{

String str = "select UserID from register;";

dt = MySqlQuery(str);

}

return dt;

}

//return the list of all playlists.

[WebMethod]

public DataTable GetList(String ID)

{

DataTable dt = new DataTable();

if (key == 1)

{

String str = "select \* from list where nickname='"+ID+"';";

dt = MySqlQuery(str);

}

return dt;

}

//return the list of tracks in certurn playlist.

[WebMethod]

public DataTable GetTrack(String playlist)

{

DataTable dt = new DataTable();

if (key == 1)

{

String str1 = "select title,artist,album,genre from track where playlist='" + playlist + "';";

String str2 = "select title,artist,album,genre from track where playlist='" + playlist + "' and public='Y';";

String str = "select \* from list where nickname='"+UserID+"' and playlist='"+playlist+"';";

DataTable dt\_test = MySqlQuery(str);

if (dt\_test.Rows.Count == 0)

{

dt = MySqlQuery(str2);

}

else

{

dt = MySqlQuery(str1);

}

}

return dt;

}

// Add a score of playlist into databases

[WebMethod]

public int AddScore(String playlist, String nickname, int score)

{

if (key == 1)

{

String str1 = "select \* from playlist\_score where playlist='" + playlist + "' and nickname='" + nickname + "' and scoring\_person='" + UserID + "';";

DataTable dt = MySqlQuery(str1);

if (dt.Rows.Count == 0)

{

String str2 = "insert into playlist\_score values('" + playlist + "','" + nickname + "','" + UserID + "'," + score.ToString() + ");";

MysqlInsert\_Delete(str2);

return 1;

}

else

{

return 0;

}

}

else

{

return 0;

}

}

//Add a track into databases

[WebMethod]

public int AddTrack(String title, String artist, String album,String genre,String playlist,String Public)

{

if (key == 1)

{

String str1 = "select \* from track where title='" + title + "' and artist='" + artist + "' and playlist='" + playlist + "';";

DataTable dt = MySqlQuery(str1);

if (dt.Rows.Count == 0)

{

String str2 = "insert into track values('" + title + "','" + artist + "','" + album + "','" + genre + "','" + playlist + "','" + Public + "');";

MysqlInsert\_Delete(str2);

return 1;

}

else

{

return 0;

}

}

else

{

return 0;

}

}

//return all the mark the authorized users made

[WebMethod]

public DataTable Mymarks()

{

DataTable dt = new DataTable();

if (key == 1)

{

String str = "select playlist,nickname,score from playlist\_score where scoring\_person='"+UserID+"';";

dt = MySqlQuery(str);

}

return dt;

}

//add a new playlist

[WebMethod]

public int AddList(String playlist)

{

String nickname = UserID;

if (key == 1)

{

String str1 = "select \* from list where playlist='" + playlist + "' and nickname='" + nickname + "';";

DataTable dt = MySqlQuery(str1);

if (dt.Rows.Count == 0)

{

String str2 = "insert into list values('" + playlist + "','" + nickname + "');";

MysqlInsert\_Delete(str2);

return 1;

}

else

{

return 0;

}

}

else

{

return 0;

}

}

//delete one whole playlist

[WebMethod]

public int DeleteList(String playlist)

{

String nickname = UserID;

if (key == 1)

{

String str1 = "select \* from list where playlist='" + playlist + "' and nickname='" + nickname + "';";

DataTable dt = MySqlQuery(str1);

if (dt.Rows.Count != 0)

{

String str2 = "delete from list where playlist='" + playlist + "' and nickname='" + nickname + "';";

MysqlInsert\_Delete(str2);

return 1;

}

else

{

return 0;

}

}

else

{

return 0;

}

}

[WebMethod]

//delete one track

public int DeleteTrack(String title, String artist, String playlist)

{

if (key == 1)

{

String str1 = "select \* from track where title='" + title + "' and artist='" + artist + "' and playlist='" + playlist + "';";

DataTable dt = MySqlQuery(str1);

if (dt.Rows.Count != 0)

{

String str2 = "delete from track where title='" + title + "' and artist='" + artist + "' and playlist='" + playlist + "';";

MysqlInsert\_Delete(str2);

return 1;

}

else

{

return 0;

}

}

else

{

return 0;

}

}

//return the ranking list of playlist.

[WebMethod]

public DataTable RankList()

{

DataTable dt = new DataTable();

if (key == 1)

{

String str = "select playlist,AVG(score) as rank from playlist\_score group by playlist order by AVG(score) desc;";

dt = MySqlQuery(str);

}

return dt;

}

//return the ranking list of tracks

[WebMethod]

public DataTable RankTrack()

{

DataTable dt = new DataTable();

if (key == 1)

{

String str = "select title,count(playlist) as rank from track group by title order by count(playlist) desc;";

dt = MySqlQuery(str);

}

return dt;

}

//return TOP 3 of playlists

[WebMethod]

public DataTable RankListTop3()

{

DataTable dt = new DataTable();

if (key == 1)

{

String str = "select playlist,AVG(score) as rank from playlist\_score group by playlist order by AVG(score) desc limit 3;";

dt = MySqlQuery(str);

}

return dt;

}

//return TOP 3 of tracks

[WebMethod]

public DataTable RankTrackTop3()

{

DataTable dt = new DataTable();

if (key == 1)

{

String str = "select title,count(playlist) as rank from track group by title order by count(playlist) desc limit 3;";

dt = MySqlQuery(str);

}

return dt;

}

//Log off and clear the local data.

[WebMethod]

public void Logoff()

{

key=0;

UserID="";

Password = "";

}

}

}

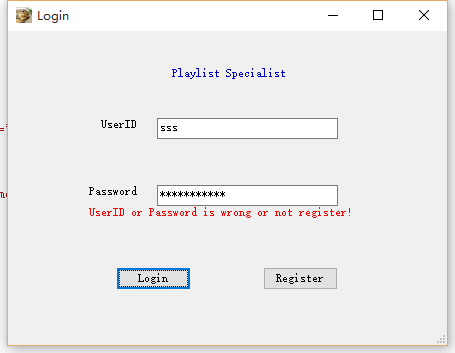
1. Client

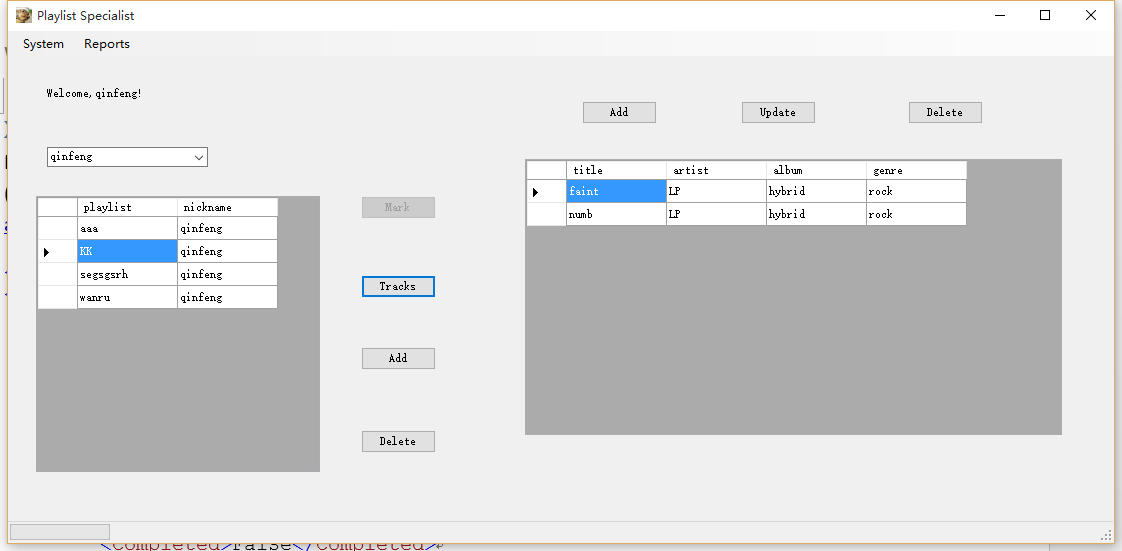
Our Client is versatile. It can not only provide users with a convenient user interface to manage their information, or giving users intuitive comprehension by graphic chart, but also limit unsafe manipulations which may let the system break down.

## Client User Interface

We design a multi-page client user interface.

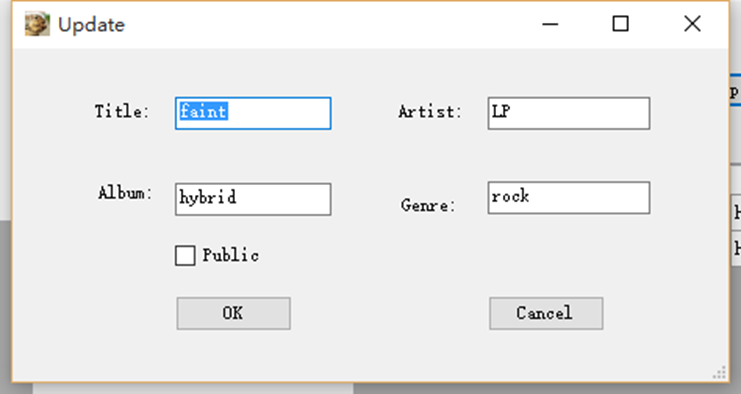
In the Login page, the users can use exist UserID to enter into the System, or register a new one. Even if the user had logged in, he or she can log off, and use other UserID to log in or Register a new one.



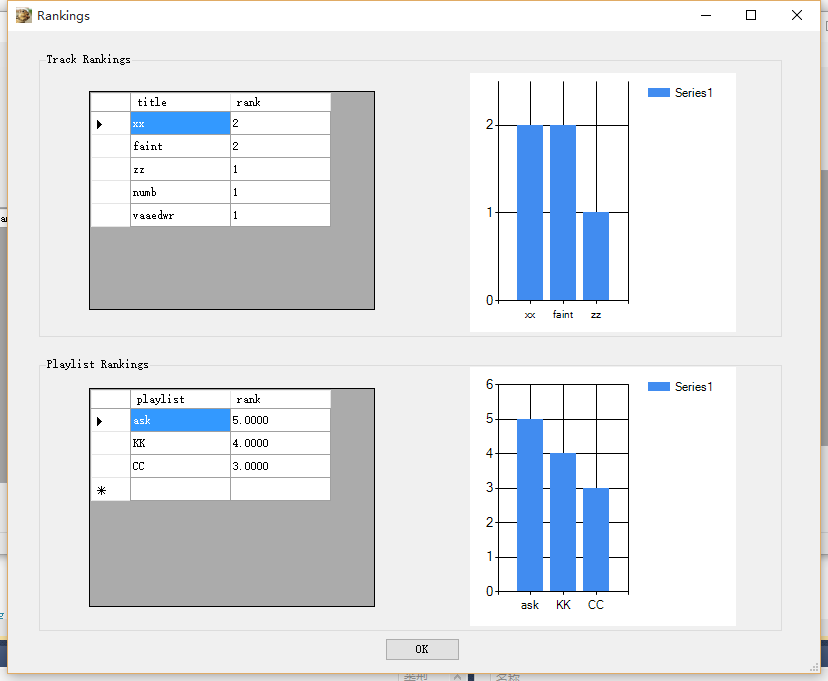


The main form connects all the functions the System offered. In the main form, people can check information from the two DataGridViews. The left one is the list of playlist, while the right one is the list of tracks. People can freely check others’ information, only if those information is set to be public by the owner. The buttons on this form are only enabled on the correct occasions.

People can update their tracks, and the old information will appear on the Unpadate Forms.



Our System calculates the rankings information MySQL databases and show it in a graphic manner. The left two table shows the whole rankings, despite the tracks or lists which no one are willing to mark. The right histograms show the TOP 3 in tracks and playlists respectively.



They are not all about our System. For more about our system, you could contact us or simply try yourself.

## Client Code

The code of our Client is extremely long (It includes 8 forms). So we only copy the Main Form code there:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class Playlist\_Specialist : Form

{

//define object abc

public myservice.Service1SoapClient abc = new myservice.Service1SoapClient();

public Playlist\_Specialist()

{

Form1 form1 = new Form1();

//Load form1

if (form1.ShowDialog() != DialogResult.OK)

{

Environment.Exit(Environment.ExitCode);

return;

}

InitializeComponent();

label1.Text = "Welcome," + abc.Name() + "!";

DataTable name = abc.GetName();

//ergodic process,which loads all the usersID.

foreach (DataRow row in name.Rows)

{

foreach (DataColumn column in name.Columns)

{

this.namelist.Items.Add(row[column]);

}

}

DataTable dt1 = abc.GetList(abc.Name());

dataGridView1.DataSource = dt1;

// if the list is vacant, some buttons are not available.

if (dt1.Rows.Count ==0)

{

Tracks.Enabled = false;

Delete\_List.Enabled = false;

}

this.namelist.Text = abc.Name();

}

private void label1\_Click(object sender, EventArgs e)

{

}

public void button1\_Click(object sender, EventArgs e)

{

}

private void closeToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

private void logOffToolStripMenuItem\_Click(object sender, EventArgs e)

{

// log off process

abc.Logoff();

Form1 form2 = new Form1();

this.Hide();

if (form2.ShowDialog() != DialogResult.OK)

{

Environment.Exit(Environment.ExitCode);

return;

}

label1.Text = "Welcome," + abc.Name() + "!";

dataGridView1.DataSource = abc.GetList(abc.Name());

this.namelist.Text = abc.Name();

DataTable dt = new DataTable();

dataGridView2.DataSource = dt;

this.Show();

}

private void label1\_Click\_1(object sender, EventArgs e)

{

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

DataTable dt1=abc.GetList(this.namelist.SelectedItem.ToString());

dataGridView1.DataSource = dt1;

// if the selection is changed, some buttons are not available.

if (this.namelist.SelectedItem.ToString() == abc.Name() || dt1.Rows.Count==0)

{

Mark.Enabled = false;

}

else

{

Mark.Enabled = true;

}

if (dt1.Rows.Count ==0)

{

Tracks.Enabled = false;

}

else

{

Tracks.Enabled = true;

}

if (this.namelist.SelectedItem.ToString() == abc.Name())

{

Add\_List.Enabled = true;

}

else

{

Add\_List.Enabled = false;

}

if (this.namelist.SelectedItem.ToString() != abc.Name() || dt1.Rows.Count == 0)

{

Delete\_List.Enabled = false;

}

else

{

Delete\_List.Enabled = true;

}

if (dt1.Rows.Count == 0 || this.namelist.SelectedItem.ToString() != abc.Name())

{

Update\_Track.Enabled = false;

Delete\_Track.Enabled = false;

}

else

{

Update\_Track.Enabled = true;

Delete\_Track.Enabled = true;

}

if (this.namelist.SelectedItem.ToString() != abc.Name())

{

Add\_Track.Enabled = false;

}

else

{

Add\_Track.Enabled = true;

}

}

private void statusStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

//add a playlist

Add\_Playlist form6 = new Add\_Playlist(abc);

if (form6.ShowDialog() == DialogResult.OK)

{

DataTable dt1 = abc.GetList(abc.Name());

dataGridView1.DataSource = dt1;

MessageBox.Show("Add successfully!");

}

}

private void Tracks\_Click(object sender, EventArgs e)

{

// check the track of a selected playlist

String ID = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["playlist"].Value.ToString();

DataTable dt1 = abc.GetTrack(ID);

dataGridView2.DataSource = dt1;

if (dt1.Rows.Count == 0 || this.namelist.SelectedItem.ToString() != abc.Name())

{

Update\_Track.Enabled = false;

Delete\_Track.Enabled = false;

}

else

{

Update\_Track.Enabled = true;

Delete\_Track.Enabled = true;

}

if (this.namelist.SelectedItem.ToString() != abc.Name())

{

Add\_Track.Enabled = false;

}

else

{

Add\_Track.Enabled = true;

}

}

private void Delete\_List\_Click(object sender, EventArgs e)

{

// delete selected list.

String ID = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["playlist"].Value.ToString();

int k = abc.DeleteList(ID);

DataTable dt1 = abc.GetList(abc.Name());

dataGridView1.DataSource = dt1;

if (k == 1)

{

MessageBox.Show("Delete successfully!");

}

else

{

MessageBox.Show("Fail to delete or non-exist!");

}

}

private void Mark\_Click(object sender, EventArgs e)

{

// mark the playlist

Mark form4 = new Mark();

if (form4.ShowDialog() == DialogResult.OK)

{

String playlist = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["playlist"].Value.ToString();

String nickname = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["nickname"].Value.ToString();

if (abc.AddScore(playlist, nickname, form4.score) == 1)

{

MessageBox.Show("Mark successfully!");

}

else

{

MessageBox.Show("Fail to mark or had marked!");

}

}

}

private void myMarksToolStripMenuItem\_Click(object sender, EventArgs e)

{

DataTable dt = abc.Mymarks();

MyMarks form5 = new MyMarks(dt);

form5.ShowDialog();

}

private void Add\_Track\_Click(object sender, EventArgs e)

{

String ID = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["playlist"].Value.ToString();

//convey the object abc and ID as parameters

Add\_Track form7 = new Add\_Track(abc,ID);

if (form7.ShowDialog() == DialogResult.OK)

{

MessageBox.Show("Add successfully!");

DataTable dt1 = abc.GetTrack(ID);

dataGridView2.DataSource = dt1;

}

}

private void Delete\_Track\_Click(object sender, EventArgs e)

{

String ID = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["playlist"].Value.ToString();

String title = dataGridView2.Rows[dataGridView2.CurrentRow.Index].Cells["title"].Value.ToString();

String artist = dataGridView2.Rows[dataGridView2.CurrentRow.Index].Cells["artist"].Value.ToString();

int k = abc.DeleteTrack(title,artist,ID);

DataTable dt1 = abc.GetTrack(ID);

dataGridView2.DataSource = dt1;

if (k == 1)

{

MessageBox.Show("Delete successfully!");

}

else

{

MessageBox.Show("Fail to delete or non-exist!");

}

}

private void Update\_Track\_Click(object sender, EventArgs e)

{

String ID = dataGridView1.Rows[dataGridView1.CurrentRow.Index].Cells["playlist"].Value.ToString();

String title = dataGridView2.Rows[dataGridView2.CurrentRow.Index].Cells["title"].Value.ToString();

String artist = dataGridView2.Rows[dataGridView2.CurrentRow.Index].Cells["artist"].Value.ToString();

String genre = dataGridView2.Rows[dataGridView2.CurrentRow.Index].Cells["genre"].Value.ToString();

String album = dataGridView2.Rows[dataGridView2.CurrentRow.Index].Cells["album"].Value.ToString();

Update form8 = new Update(abc,ID,title,artist,album,genre);

if (form8.ShowDialog() == DialogResult.OK)

{

MessageBox.Show("Update successfully!");

DataTable dt1 = abc.GetTrack(ID);

dataGridView2.DataSource = dt1;

}

}

private void scoresToolStripMenuItem\_Click(object sender, EventArgs e)

{

Rankings form9 = new Rankings(abc);

form9.ShowDialog();

}

private void Playlist\_Specialist\_Load(object sender, EventArgs e)

{

}

}

}