

```

/* Jacob Smith
 * Main WinForm.
 */

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Drawing.Drawing2D;
using System.Runtime.InteropServices;
using System.IO;
using System.Drawing.Printing;

namespace Fbla_Lib
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            SetPlaceholder(textBox2, "Search Book Titles!");
            SetPlaceholder(textBox1, "Search Names or Ids!");
        }

        #region Search_Asset
        /// <summary>
        /// Lets textboxes to have placeholder text.
        /// </summary>
        /// <param name="control"></param>
        /// <param name="PlaceholderText"></param>
        public void SetPlaceholder(Control control, string PlaceholderText)
        {
            control.Text = PlaceholderText;
            control.GotFocus += delegate (object sender, EventArgs args)
            {
                if (control.Text == PlaceholderText)
                {
                    control.Text = "";
                }
            };
            control.LostFocus += delegate (object sender, EventArgs args)
            {
                if (control.Text.Length == 0)
                {
                    control.Text = PlaceholderText;
                }
            };
        }
    }
}

```

```

#endregion
#region Saving Pathing
string filepath = @"C:\Program Files\WisdomHighSchoolLib\";
string bookPath = @"C:\Program Files\WisdomHighSchoolLib\books.xml";
string memberPath = @"C:\Program Files\WisdomHighSchoolLib\members.xml";
DateTime date = DateTime.Today;
//
AutoCompleteStringCollection memberIds = new AutoCompleteStringCollection();
AutoCompleteStringCollection memberNames = new AutoCompleteStringCollection();

private void Form1_Load(object sender, EventArgs e)
{
    if (File.Exists(memberPath))
    {
        try
        {
            DataSet ds = new DataSet();
            ds.ReadXml(memberPath);
            foreach (DataRow item in ds.Tables["members"].Rows)
            {
                int n = dataGridView1.Rows.Add();
                #region Rows
                dataGridView1.Rows[n].Cells[0].Value = item["Id"].ToString();
                dataGridView1.Rows[n].Cells[1].Value = item["FirstName"].ToString();
                dataGridView1.Rows[n].Cells[2].Value = item["LastName"].ToString();
                dataGridView1.Rows[n].Cells[3].Value = item["MemberType"].ToString();
                dataGridView1.Rows[n].Cells[4].Value = item["NumberBooksOut"].ToString();
                dataGridView1.Rows[n].Cells[5].Value = item["MaxBooksOut"].ToString();
                dataGridView1.Rows[n].Cells[6].Value = item["DateNextBook"].ToString();
                dataGridView1.Rows[n].Cells[7].Value = item["bookidOut"].ToString();
                dataGridView1.Rows[n].Cells[8].Value = item["bookFee"].ToString();
                #endregion
            }
        }
        catch (Exception ex)
        {
            // MessageBox.Show("Members Missing", ex.ToString());
        }
        try
        {
            DataSet ds = new DataSet();
            ds.ReadXml(bookPath);
            foreach (DataRow item in ds.Tables["books"].Rows)
            {
                int n = dataGridView2.Rows.Add();
                #region Rows
                dataGridView2.Rows[n].Cells[0].Value = item["Data_bookID"].ToString();
                dataGridView2.Rows[n].Cells[1].Value = item["booktitle"].ToString();
                dataGridView2.Rows[n].Cells[2].Value = item["bookAuthor"].ToString();
                dataGridView2.Rows[n].Cells[3].Value = item["Isout"].ToString();
                dataGridView2.Rows[n].Cells[4].Value = item["whenBack"].ToString();
                #endregion
            }
        }
    }
}

```

```

    }
}
catch (Exception ex)
{
    //MessageBox.Show( ex.ToString());
}

AutoCompleteStringCollection source = new AutoCompleteStringCollection();
foreach (DataGridViewRow item in dataGridView2.Rows)
{
    if (item.Cells[1].Value != null) //value is not null
    {
        source.Add(item.Cells[1].Value.ToString());
    }
}
textBox2.AutoCompleteSource = AutoCompleteSource.CustomSource;
textBox1.AutoCompleteSource = AutoCompleteSource.CustomSource;
textBox2.AutoCompleteCustomSource = source;
textBox1.AutoCompleteCustomSource = memberIds;
//-----
//-----
foreach (DataGridViewRow item in dataGridView1.Rows)
{
    if (item.Cells[0].Value != null)
    {
        memberIds.Add(item.Cells[0].Value.ToString());
    }
}
foreach (DataGridViewRow item in dataGridView1.Rows)
{
    if (item.Cells[1].Value != null)
    {
        memberNames.Add(item.Cells[1].Value.ToString());
    }
}
}
if (!File.Exists(memberPath))
{
    Directory.CreateDirectory(filepath);
    File.Create(bookPath);
    File.Create(memberPath);
    System.Diagnostics.Process.Start(Application.ExecutablePath); // to start new instance of application
    this.Close(); //to turn off current app
    MessageBox.Show("Created at: " + filepath);
}

//Adds sum for all missing books
try
{
    List<string> list = new List<string>();
    foreach (DataGridViewRow item in dataGridView1.Rows)
    {

```

```

        if (item.Cells[6].Value != "0")
        {
            list.Add(item.Cells[0].Value.ToString() + "#" + item.Cells[6].Value.ToString() + "#" +
item.Cells[8].Value.ToString());
            lb_owed.Items.Add("Name: " + item.Cells[1].Value.ToString() + " " + item.Cells[2].Value.ToString() +
" Books Id: " + item.Cells[7].Value.ToString() + " Amount Owed: " + item.Cells[8].Value.ToString());

        }

        foreach (string person in list)
        {
            if (Convert.ToInt32(person.Split('/')[1]) <= Convert.ToInt32(date.Day))
            {
                string money = person.Split('$')[1].ToString();
                int whole = Convert.ToInt32(money.Split('.')[0]);
                int change = Convert.ToInt32(money.Split('.')[1]);
                if (change >= 70)
                {
                    whole++;
                    change = 00;
                }
                if (change < 70)
                {
                    change = change + 30;
                }
                string final = "$" + whole.ToString() + "." + change.ToString();

                string searchValue = person.Split('#')[0];

                dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
                bool valueResult = false;
                foreach (DataGridViewRow row in dataGridView1.Rows)
                {
                    for (int i = 0; i < row.Cells.Count; i++)
                    {
                        if (row.Cells[i].Value != null && row.Cells[i].Value.ToString().Equals(searchValue))
                        {
                            int rowIndex = row.Index;
                            dataGridView1.Rows[rowIndex].Selected = true;
                            valueResult = true;
                            break;
                        }
                    }
                }

                item.Cells[8].Value = final;

            }

        }

    }
}

```

```

    }
    catch(Exception ex)
    {

    }

    try
    {
        List<string> list = new List<string>();
        foreach (DataGridViewRow item in dataGridView2.Rows)
        {
            if(item.Cells[3].Value.ToString() == "False")
                lb_Books.Items.Add("Book ID: " + item.Cells[0].Value.ToString() + " Book Title: " +
item.Cells[1].Value.ToString() + " Return Date: "+ item.Cells[4].Value.ToString());
        }

    }
    catch(Exception ex)
    {

    }
}

```

```

private void buttonAddMember_Click(object sender, EventArgs e)

```

```

{
    List<string> list = new List<string>();
    foreach (DataGridViewRow item in dataGridView1.Rows)
    {
        if (item.Cells[1].Value != null)
        {
            list.Add(item.Cells[0].Value.ToString());
        }
    }
    if (list.Contains(tb_Id.Text))
    {
        MessageBox.Show("This Member is already in DataBase.");
        tb_Id.Clear();
        tb_Id.Focus();
        goto skip;
    }
}

```

```

int n = dataGridView1.Rows.Add();
dataGridView1.Rows[n].Cells[0].Value = tb_Id.Text;
dataGridView1.Rows[n].Cells[1].Value = tb_firstname.Text;
dataGridView1.Rows[n].Cells[2].Value = tb_lastname.Text;
if (rB_Student.Checked == true)
{
    dataGridView1.Rows[n].Cells[3].Value = "Student";
    dataGridView1.Rows[n].Cells[5].Value = "5";
}
if (rB_Staff.Checked == true)
{
    dataGridView1.Rows[n].Cells[3].Value = "Staff";
    dataGridView1.Rows[n].Cells[5].Value = "10";
}

```

```

}
dataGridView1.Rows[n].Cells[4].Value = "0";

dataGridView1.Rows[n].Cells[6].Value = "0";
dataGridView1.Rows[n].Cells[7].Value = "";
dataGridView1.Rows[n].Cells[8].Value = "$0.00";
// saving
ds = new DataSet();
dt = new DataTable();
dt.TableName = "members";
#region Columns
dt.Columns.Add("Id");
dt.Columns.Add("FirstName");
dt.Columns.Add("LastName");
dt.Columns.Add("MemberType");
dt.Columns.Add("NumberBooksOut");
dt.Columns.Add("MaxBooksOut");
dt.Columns.Add("DateNextBook");
dt.Columns.Add("bookidout");
dt.Columns.Add("bookFee");
#endregion
ds.Tables.Add(dt);
try
{
    foreach (DataGridViewRow r in dataGridView1.Rows)
    {
        DataRow row1 = ds.Tables["members"].NewRow();
        #region Rows
        row1["Id"] = r.Cells[0].Value.ToString();
        row1["FirstName"] = r.Cells[1].Value.ToString();
        row1["LastName"] = r.Cells[2].Value.ToString();
        row1["MemberType"] = r.Cells[3].Value.ToString();
        row1["NumberBooksOut"] = r.Cells[4].Value.ToString();
        row1["MaxBooksOut"] = r.Cells[5].Value.ToString();
        row1["DateNextBook"] = r.Cells[6].Value.ToString();
        row1["bookidout"] = r.Cells[7].Value.ToString();
        row1["bookFee"] = r.Cells[8].Value.ToString();
        #endregion
        ds.Tables["members"].Rows.Add(row1);
    }
}
catch (Exception ex)
{
    //Catches Null when converting to xml from dataGridView
    //MessageBox.Show(ex.ToString());
}

ds.WriteXml(memberPath);
skip:
// skipped
Console.WriteLine("Skipped Process");
}

```

```

DataSet ds;
DataTable dt;
private void buttonMemberSave_Click(object sender, EventArgs e)
{
    // saving
    ds = new DataSet();
    dt = new DataTable();
    dt.TableName = "members";
    #region Columns
    dt.Columns.Add("Id");
    dt.Columns.Add("FirstName");
    dt.Columns.Add("LastName");
    dt.Columns.Add("MemberType");
    dt.Columns.Add("NumberBooksOut");
    dt.Columns.Add("MaxBooksOut");
    dt.Columns.Add("DateNextBook");
    dt.Columns.Add("bookidout");
    dt.Columns.Add("bookFee");
    #endregion
    ds.Tables.Add(dt);
    try
    {
        foreach (DataGridViewRow r in dataGridView1.Rows)
        {
            DataRow row1 = ds.Tables["members"].NewRow();
            #region Rows
            row1["Id"] = r.Cells[0].Value.ToString();
            row1["FirstName"] = r.Cells[1].Value.ToString();
            row1["LastName"] = r.Cells[2].Value.ToString();
            row1["MemberType"] = r.Cells[3].Value.ToString();
            row1["NumberBooksOut"] = r.Cells[4].Value.ToString();
            row1["MaxBooksOut"] = r.Cells[5].Value.ToString();
            row1["DateNextBook"] = r.Cells[6].Value.ToString();
            row1["bookidout"] = r.Cells[7].Value.ToString();
            row1["bookFee"] = r.Cells[8].Value.ToString();
            #endregion
            ds.Tables["members"].Rows.Add(row1);
        }
    }
    catch (Exception ex)
    {
        //Catches Null when converting to xml from dataViewGrid
        //MessageBox.Show(ex.ToString());
    }

    ds.WriteXml(memberPath);
}
/// <summary>
/// Add Book To DB
/// </summary>
/// <param name="sender"></param>

```

```

/// <param name="e"></param>
private void buttonAddBook_Click(object sender, EventArgs e)
{
    List<string> list = new List<string>();
    foreach (DataGridViewRow item in dataGridView2.Rows)
    {
        if (item.Cells[1].Value != null)
        {
            list.Add(item.Cells[0].Value.ToString());
        }
    }
    if (list.Contains(textBoxBookId.Text))
    {
        MessageBox.Show("This book is already in DataBase.");
        textBoxBookId.Clear();
        textBoxBookId.Focus();
    }
    else
    {
        int n = dataGridView2.Rows.Add();
        dataGridView2.Rows[n].Cells[0].Value = textBoxBookId.Text;
        dataGridView2.Rows[n].Cells[1].Value = textBoxBookTitle.Text;
        dataGridView2.Rows[n].Cells[2].Value = textBoxBookAuthor.Text;
        dataGridView2.Rows[n].Cells[3].Value = "True";
        dataGridView2.Rows[n].Cells[4].Value = "0";
        // saving
        ds = new DataSet();
        dt = new DataTable();
        dt.TableName = "books";
        #region Columns
        dt.Columns.Add("Data_bookID");
        dt.Columns.Add("booktitle");
        dt.Columns.Add("bookAuthor");
        dt.Columns.Add("Isout");
        dt.Columns.Add("whenBack");
        #endregion
        ds.Tables.Add(dt);
        try
        {
            foreach (DataGridViewRow r in dataGridView2.Rows)
            {
                DataRow row1 = ds.Tables["books"].NewRow();
                #region Rows
                row1["Data_bookID"] = r.Cells[0].Value.ToString();
                row1["booktitle"] = r.Cells[1].Value.ToString();
                row1["bookAuthor"] = r.Cells[2].Value.ToString();
                row1["Isout"] = r.Cells[3].Value.ToString();
                row1["whenBack"] = r.Cells[4].Value.ToString();
                #endregion
                ds.Tables["books"].Rows.Add(row1);
            }
        }
    }
}

```



```

        catch (Exception ex)
        {
            //Catches Null when converting to xml from dataViewGrid
            // MessageBox.Show(ex.ToString());

        }

        ds.WriteXml(bookPath);
        textBoxBookId.Clear();
        textBoxBookTitle.Clear();
        textBoxBookAuthor.Clear();
    }
}
/// <summary>
/// Creates Book Key
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    List<int> list = new List<int>();
    foreach (DataGridViewRow item in dataGridView2.Rows)
    {
        if (item.Cells[1].Value != null)
        {
            list.Add(Convert.ToInt32(item.Cells[0].Value));
        }
    }
    try
    {
        list.Sort();
        int idCurrent = list.Max() + 1;
        textBoxBookId.Text = idCurrent.ToString();
    }
    catch (Exception ex)
    {
        textBoxBookId.Text = "0";
    }
}
/// <summary>
/// Creates Member id
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void linkLabel2_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    List<int> list = new List<int>();
    foreach (DataGridViewRow item in dataGridView1.Rows)
    {
        if (item.Cells[1].Value != null)
        {
            list.Add(Convert.ToInt32(item.Cells[0].Value));
        }
    }
}

```

```

try
{
    list.Sort();
    int idCurrent = list.Max() + 1;
    tb_Id.Text = idCurrent.ToString();
}
catch(Exception ex)
{
    tb_Id.Text = "0";
}
}
#endregion
#region Menus
private void textBox1_Click(object sender, EventArgs e)
{
    //if(radioButton_memberId.Checked == true)
    //{
    //    textBox1.AutoCompleteCustomSource = memberIds;
    //}
    //if(radioButton_membername.Checked == true)
    //{
    //    textBox1.AutoCompleteCustomSource = memberNames;
    //}
}
/// <summary>
/// Search By
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void radioButton_memberId_CheckedChanged(object sender, EventArgs e)
{
    if(radioButton_memberId.Checked == true)
    {
        textBox1.AutoCompleteCustomSource = memberIds;
    }
    if(radioButton_memberId.Checked == false)
    {
        textBox1.AutoCompleteCustomSource = memberNames;
    }
}
/// <summary>
/// Search Members filter
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void button6_Click(object sender, EventArgs e)
{
    string searchValue = textBox1.Text;
    dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
    try
    {
        bool valueResult = false;
        foreach (DataGridViewRow row in dataGridView1.Rows)
        {

```

```

        for (int i = 0; i < row.Cells.Count; i++)
        {
            if (row.Cells[i].Value != null && row.Cells[i].Value.ToString().Equals(searchValue))
            {
                int rowIndex = row.Index;
                dataGridView1.Rows[rowIndex].Selected = true;
                valueResult = true;
                break;
            }
        }
    }
    if (!valueResult)
    {
        MessageBox.Show("Unable to find " + textBox1.Text, "Not Found");
        return;
    }
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message);
}
}
/// <summary>
/// Search for book
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void button_BookSearch_Click(object sender, EventArgs e)
{
    string searchValue = textBox2.Text;
    dataGridView2.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
    try
    {
        bool valueResult = false;
        foreach (DataGridViewRow row in dataGridView2.Rows)
        {
            for (int i = 0; i < row.Cells.Count; i++)
            {
                if (row.Cells[i].Value != null && row.Cells[i].Value.ToString().Equals(searchValue))
                {
                    int rowIndex = row.Index;
                    dataGridView2.Rows[rowIndex].Selected = true;
                    valueResult = true;
                    break;
                }
            }
        }
    }
    if (!valueResult)
    {
        MessageBox.Show("Unable to find " + textBox2.Text, "Not Found");
        return;
    }
}

```

```

    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.Message);
    }
}
/// <summary>
/// Get Checkout Ready
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void buttonCheckOut_Click(object sender, EventArgs e)
{
    tP_CheckOut.BringToFront();
    tP_CheckOut.Show();
    textBoxMemberId.Focus();
}
/// <summary>
/// Return books
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void buttonReturn_Click(object sender, EventArgs e)
{
    tP_Return.BringToFront();
    tP_Return.Show();
    textBoxBookId.Focus();
}
/// <summary>
/// Checkout Process
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void buttonCheckOut_continue_Click(object sender, EventArgs e)
{
    //Check if they can get a book
    int value2 = 0;
    int value1 = 0;
    string searchValue = textBoxMemberId.Text;
    dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
    try
    {
        bool valueResult = false;
        foreach (DataGridViewRow row in dataGridView1.Rows)
        {
            for (int i = 0; i < row.Cells.Count; i++)
            {
                if (row.Cells[i].Value != null && row.Cells[i].Value.ToString().Equals(searchValue))
                {
                    int rowIndex = row.Index;
                    dataGridView1.Rows[rowIndex].Selected = true;
                    valueResult = true;
                    break;
                }
            }
        }
    }
}

```

```

    }

    }
    if (!valueResult)
    {
        MessageBox.Show("Unable to find " + textBoxMemberId.Text, "Not Found");
        return;
    }
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message);
}
foreach (DataGridViewRow row in dataGridView1.SelectedRows)
{
    // Gets Books out
    value1 = Convert.ToInt32(row.Cells[4].Value);
    // Gets max books
    value2 = Convert.ToInt32(row.Cells[5].Value);
}
if(value1 >= value2)
{
    MessageBox.Show("Member at Book Limit! \n");
    textBoxMemberId.Clear();
}
else
{
    CheckOutContinue.Enabled = true;
    //Do check out.
    textBoxCheckOutBookId.Focus();
}

}

}
/// <summary>
/// Submits Check out info.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void buttonSubmitCheckOut_Click(object sender, EventArgs e)
{
    string currentBook = "";
    string memberName = "";
    string memberid = "";
    string bookid = textBoxCheckOutBookId.Text;

    string searchValue = textBoxCheckOutBookId.Text;
    dataGridView2.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
    try
    {
        bool valueResult = false;
        foreach (DataGridViewRow row in dataGridView2.Rows)

```

```

{
    for (int i = 0; i < row.Cells.Count; i++)
    {
        if (row.Cells[i].Value != null && row.Cells[i].Value.ToString().Equals(searchValue))
        {
            int rowIndex = row.Index;
            dataGridView2.Rows[rowIndex].Selected = true;
            valueResult = true;
            break;
        }
    }
}
if (!valueResult)
{
    MessageBox.Show("Unable to find " + textBoxCheckOutBookId.Text, "Not Found");
    return;
}
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message);
}

foreach (DataGridViewRow item in dataGridView2.Rows)
{
    if (item.Cells[3].Value != null)
    {
        if (item.Cells[3].Value == "False")
        {
            MessageBox.Show("Book Out");
            goto skip;
        }
    }
}
//Gets book name
foreach (DataGridViewRow row in dataGridView2.SelectedRows)
{
    currentBook = row.Cells[1].Value.ToString();
}
//Gets Member Name
foreach (DataGridViewRow row in dataGridView1.SelectedRows)
{
    memberName = row.Cells[1].Value.ToString() + " " + row.Cells[2].Value.ToString();
    memberid = row.Cells[0].Value.ToString();
}
CheckoutInfo.Items.Add("Name: " + memberName);
CheckoutInfo.Items.Add("id: " + memberid);
CheckoutInfo.Items.Add("Book Title: " + currentBook);
CheckoutInfo.Items.Add("Date: " + date.ToShortDateString().ToString());
CheckoutInfo.Items.Add("Return Date: " + date.AddDays(14).ToShortDateString());
skip:
Console.WriteLine("Skipped");
}

```

```

/// <summary>
/// Cancel Checkout
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void buttonCancel_Click(object sender, EventArgs e)
{
    CheckOutInfo.Items.Clear();
    textBoxCheckOutBookId.Clear();
    CheckOutContinue.Enabled = false;
    textBoxMemberId.Clear();
    tP_Clear.BringToFront();
    tP_Clear.Show();
}

/// <summary>
/// CheckOut
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void buttonOk_Click(object sender, EventArgs e)
{
    Checkout_ok(false);
    // saving
    ds = new DataSet();
    dt = new DataTable();
    dt.TableName = "members";
    #region Columns
    dt.Columns.Add("Id");
    dt.Columns.Add("FirstName");
    dt.Columns.Add("LastName");
    dt.Columns.Add("MemberType");
    dt.Columns.Add("NumberBooksOut");
    dt.Columns.Add("MaxBooksOut");
    dt.Columns.Add("DateNextBook");
    dt.Columns.Add("bookidout");
    dt.Columns.Add("bookFee");
    #endregion
    ds.Tables.Add(dt);
    try
    {
        foreach (DataGridViewRow r in dataGridView1.Rows)
        {
            DataRow row1 = ds.Tables["members"].NewRow();
            #region Rows
            row1["Id"] = r.Cells[0].Value.ToString();
            row1["FirstName"] = r.Cells[1].Value.ToString();
            row1["LastName"] = r.Cells[2].Value.ToString();
            row1["MemberType"] = r.Cells[3].Value.ToString();
            row1["NumberBooksOut"] = r.Cells[4].Value.ToString();
            row1["MaxBooksOut"] = r.Cells[5].Value.ToString();
            row1["DateNextBook"] = r.Cells[6].Value.ToString();
            row1["bookidout"] = r.Cells[7].Value.ToString();
            row1["bookFee"] = r.Cells[8].Value.ToString();

```

```

        #endregion
        ds.Tables["members"].Rows.Add(row1);
    }

}
catch (Exception ex)
{
    //Catches Null when converting to xml from dataViewGrid
    //MessageBox.Show(ex.ToString());

}

ds.WriteXml(memberPath);

// saving
ds = new DataSet();
dt = new DataTable();
dt.TableName = "books";
#region Columns
dt.Columns.Add("Data_bookID");
dt.Columns.Add("booktitle");
dt.Columns.Add("bookAuthor");
dt.Columns.Add("Isout");
dt.Columns.Add("whenBack");
#endregion
ds.Tables.Add(dt);
try
{
    foreach (DataGridViewRow r in dataGridView2.Rows)
    {
        DataRow row1 = ds.Tables["books"].NewRow();
        #region Rows
        row1["Data_bookID"] = r.Cells[0].Value.ToString();
        row1["booktitle"] = r.Cells[1].Value.ToString();
        row1["bookAuthor"] = r.Cells[2].Value.ToString();
        row1["Isout"] = r.Cells[3].Value.ToString();
        row1["whenBack"] = r.Cells[4].Value.ToString();
        #endregion
        ds.Tables["books"].Rows.Add(row1);
    }

}
catch (Exception ex)
{
    //Catches Null when converting to xml from dataViewGrid
    //MessageBox.Show(ex.ToString());

}

ds.WriteXml(bookPath);
}

```



```

/// <summary>
/// Checking Out Book
/// </summary>
/// <param name="print"></param>
private void Checkout_ok(bool print)
{
    // Get All Info
    string bookid = "";
    string memberid = "";
    //Gets Member ID
    foreach (DataGridViewRow row in dataGridView1.SelectedRows)
    {
        memberid = row.Cells[0].Value.ToString();
    }
    //Gets book ID
    foreach (DataGridViewRow row in dataGridView2.SelectedRows)
    {
        bookid = row.Cells[0].Value.ToString();
    }
    //Set New Data members
    foreach (DataGridViewRow row in dataGridView1.SelectedRows)
    {
        int booksout = Convert.ToInt32(row.Cells[4].Value);
        booksout++;
        row.Cells[4].Value = booksout;
        string today = date.AddDays(14).Day.ToString();
        string NextDueDate = row.Cells[6].Value.ToString();

        try
        {
            NextDueDate = NextDueDate.Split('/')[1];
        }
        catch(Exception ex)
        {
            //Do nothing
        }
        try
        {
            if (Convert.ToInt32(today) >= (Convert.ToInt32(NextDueDate)) || NextDueDate == null)
            {
                row.Cells[6].Value = date.ToShortDateString();
            }
        }
        catch(Exception ex)
        {
            MessageBox.Show("> " + ex.ToString());
        }
        string out1 = "";
        try
        {
            out1 = row.Cells[7].Value.ToString();
        }
        catch(Exception ex)
    }
}

```

```

    {
        //do nothing
    }

    if (out1 == "")
    {
        out1 = out1 + "" + bookid;
    }
    else
    {
        out1 = out1 + ", " + bookid;
    }
    row.Cells[7].Value = out1;
}
//Set New Data Books
foreach (DataGridViewRow row in dataGridView2.SelectedRows)
{
    row.Cells[3].Value = "False";
    row.Cells[4].Value = date.AddDays(14).ToShortDateString();

}

//Clear and Reset
if(print == false)
{
    CheckOutInfo.Items.Clear();
    textBoxCheckOutBookId.Clear();
    CheckOutContinue.Enabled = false;
    textBoxMemberId.Clear();
    tP_Clear.BringToFront();
    tP_Clear.Show();
    MessageBox.Show("Book Checked Out!");
}
if(print == true)
{

}

}

/// <summary>
/// Saving Book Data
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void button7_Click(object sender, EventArgs e)
{
    // saving
    ds = new DataSet();
    dt = new DataTable();
    dt.TableName = "books";
    #region Columns
    dt.Columns.Add("Data_bookID");
    dt.Columns.Add("booktitle");

```

```

dt.Columns.Add("bookAuthor");
dt.Columns.Add("Isout");
dt.Columns.Add("whenBack");
#endregion
ds.Tables.Add(dt);
try
{
    foreach (DataGridViewRow r in dataGridView2.Rows)
    {
        DataRow row1 = ds.Tables["books"].NewRow();
        #region Rows
        row1["Data_bookID"] = r.Cells[0].Value.ToString();
        row1["booktitle"] = r.Cells[1].Value.ToString();
        row1["bookAuthor"] = r.Cells[2].Value.ToString();
        row1["Isout"] = r.Cells[3].Value.ToString();
        row1["whenBack"] = r.Cells[4].Value.ToString();
        #endregion
        ds.Tables["books"].Rows.Add(row1);
    }

}
catch (Exception ex)
{
    //Catches Null when converting to xml from dataViewGrid
    // MessageBox.Show(ex.ToString());

}

ds.WriteXml(bookPath);
}
#endregion
#region Book_Return

private void buttonReturnBook_Click(object sender, EventArgs e)
{
    groupBoxReturn.Enabled = true;
}

private void buttonReturned_Click(object sender, EventArgs e)
{
    // Get All Info
    string bookid = "";
    string memberid = "";
    //Gets Member ID
    foreach (DataGridViewRow row in dataGridView1.SelectedRows)
    {
        memberid = row.Cells[0].Value.ToString();
    }
    //Gets book ID
    foreach (DataGridViewRow row in dataGridView2.SelectedRows)
    {
        bookid = row.Cells[0].Value.ToString();
    }
    //Set New Data members
    foreach (DataGridViewRow row in dataGridView1.SelectedRows)

```

```

{
    int booksout = Convert.ToInt32(row.Cells[4].Value);
    booksout--;
    row.Cells[4].Value = booksout;
    string NextDueDate = row.Cells[6].Value.ToString();
    try
    {
        row.Cells[6].Value = "0";
    }
    catch (Exception ex)
    {
        MessageBox.Show("> " + ex.ToString());
    }
    string out1 = "";
    try
    {
        out1 = row.Cells[7].Value.ToString().Replace(bookid, "");
    }
    catch (Exception ex)
    {
        //do nothing
    }

    if (out1.Contains(bookid))
    {
    }
    row.Cells[7].Value = out1;
}
//Set New Data Books
foreach (DataGridViewRow row in dataGridView2.SelectedRows)
{
    row.Cells[3].Value = "True";
    row.Cells[4].Value = "0";
}

//Clear and Reset
textBoxReturnID.Clear();
textBoxReturnMember.Clear();
textBoxOwed.Clear();
groupBoxReturn.Enabled = false;
textBoxMemberId.Clear();
tP_Clear.BringToFront();
tP_Clear.Show();
MessageBox.Show("Book Returned!");
}
#endregion
#region MenuButtons
private void buttonHome_Click(object sender, EventArgs e)
{
    tabHome.BringToFront();
    tabHome.Show();
}

```

```

}

private void buttonMembers_Click(object sender, EventArgs e)
{
    tabMembers.BringToFront();
    tabMembers.Show();
}

private void buttonBooks_Click(object sender, EventArgs e)
{
    tabBooks.BringToFront();
    tabBooks.Show();
}

private void buttonReports_Click(object sender, EventArgs e)
{
    tabReports.BringToFront();
    tabReports.Show();
}

}
/// <summary>
/// Closes Program Safely
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void button1_Click(object sender, EventArgs e)
{
    Application.Exit();
}
#endregion
#region Printing
private void buttonPrintOk_Click(object sender, EventArgs e)
{
    Checkout_ok(true);
}
private void buttonPrintOwe_Click(object sender, EventArgs e)
{
    try
    {
        PrintDocument pd = new PrintDocument();
        pd.PrintPage += new PrintPageEventHandler(pd_PrintPage);
        pd.Print();
    }
    catch (Exception ex)
    {
    }
}

private void pd_PrintPage(object sender, PrintPageEventArgs e)
{
    int i = 13;
    int x = lb_owed.Items.Count;
    //Header

```

```

e.Graphics.DrawString("Wisdom", new Font(dataGridView1.Font, FontStyle.Bold),
    Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -
    e.Graphics.MeasureString("Wisdom", new Font(dataGridView1.Font,
        FontStyle.Bold), e.MarginBounds.Width).Height - 13);

String strDate = DateTime.Now.ToLongDateString() + " " + DateTime.Now.ToShortTimeString();
//Date
e.Graphics.DrawString(strDate, new Font(dataGridView1.Font, FontStyle.Bold),
    Brushes.Black, e.MarginBounds.Left + (e.MarginBounds.Width -
    e.Graphics.MeasureString(strDate, new Font(dataGridView1.Font,
        FontStyle.Bold), e.MarginBounds.Width).Width), e.MarginBounds.Top -
    e.Graphics.MeasureString("Wisdom", new Font(new Font(dataGridView1.Font,
        FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);
foreach (string line in lb_owed.Items)
{
    e.Graphics.DrawString(line, new Font(dataGridView1.Font, FontStyle.Bold),
        Brushes.Black, e.MarginBounds.Left + 10, e.MarginBounds.Top + i -
        e.Graphics.MeasureString(line, new Font(dataGridView1.Font,
            FontStyle.Bold), e.MarginBounds.Width).Height - 13);
    i += 13;
}
}

private void buttonPrintOutBooks_Click(object sender, EventArgs e)
{
    try
    {
        PrintDocument pd = new PrintDocument();
        pd.PrintPage += new PrintPageEventHandler(pd_PrintPageBook);
        pd.Print();
    }
    catch (Exception ex)
    {
    }
}

private void pd_PrintPageBook(object sender, PrintPageEventArgs e)
{
    int i = 13;
    int x = lb_Books.Items.Count;
    //Header
    e.Graphics.DrawString("Wisdom", new Font(dataGridView1.Font, FontStyle.Bold),
        Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -
        e.Graphics.MeasureString("Wisdom", new Font(dataGridView1.Font,
            FontStyle.Bold), e.MarginBounds.Width).Height - 13);

    String strDate = DateTime.Now.ToLongDateString() + " " + DateTime.Now.ToShortTimeString();
    //Date
    e.Graphics.DrawString(strDate, new Font(dataGridView1.Font, FontStyle.Bold),
        Brushes.Black, e.MarginBounds.Left + (e.MarginBounds.Width -
        e.Graphics.MeasureString(strDate, new Font(dataGridView1.Font,
            FontStyle.Bold), e.MarginBounds.Width).Width), e.MarginBounds.Top -
        e.Graphics.MeasureString("Wisdom", new Font(new Font(dataGridView1.Font,
            FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);
}

```

```
        FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);
foreach (string line in lb_Books.Items)
{
    e.Graphics.DrawString(line, new Font(dataGridView1.Font, FontStyle.Bold),
        Brushes.Black, e.MarginBounds.Left + 10, e.MarginBounds.Top + i -
        e.Graphics.MeasureString(line, new Font(dataGridView1.Font,
            FontStyle.Bold), e.MarginBounds.Width).Height - 13);
    i += 13;
}
}
```

```
#endregion
```

```
}
}
```