

```

App_Code\CSVExporter.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.SqlClient;
using System.Web.Configuration;

/// <summary>
/// Summary description for CSVExporter
/// </summary>
public class CSVExporter
{
    public CSVExporter()
    {
        //
        // TODO: Add constructor logic here
        //
    }

    public static void writeDelimitedData(DataTable dataTable, string fileName, string delimiter)
    {
        HttpContext.Current.Response.Clear();
        HttpContext.Current.Response.ContentType = "text/csv";
        HttpContext.Current.Response.Charset = "utf-8";
        HttpContext.Current.Response.ContentEncoding = System.Text.Encoding.GetEncoding("windows-
1250");
        HttpContext.Current.Response.AppendHeader("Content-Disposition",
string.Format("attachment; filename={0}", fileName));

        for (int i = 0; i < dataTable.Columns.Count; i++)
        {
            HttpContext.Current.Response.Write(dataTable.Columns[i].ColumnName);
            HttpContext.Current.Response.Write((i < dataTable.Columns.Count - 1) ? delimiter :
Environment.NewLine);
        }

        foreach (DataRow row in dataTable.Rows)
        {
            for (int i = 0; i < dataTable.Columns.Count; i++)
            {
                HttpContext.Current.Response.Write(row[i].ToString());
                HttpContext.Current.Response.Write((i < dataTable.Columns.Count - 1) ? delimiter :
Environment.NewLine);
            }
        }

        HttpContext.Current.Response.End();
    }

    public static DataTable getDatabaseCopy()
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[get_database_copy]";
        cmd.Connection = con;

        DataTable dataTable = new DataTable();

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();

            dataTable.Load(rdr);
        }
    }
}

```

```

        catch (SqlException ex)
        {
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }

        return dataTable;
    }
}

App_Code\DatabaseProcedures.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Configuration;
using System.Data.SqlClient;
using System.Data;
using System.Globalization;

/// <summary>
/// Summary description for DatabaseProcedures
/// </summary>
public class DatabaseProcedures
{
    public static string SERVER = "TPPServer";

    public DatabaseProcedures()
    {
        //
        // TODO: Add constructor logic here
        //
    }

    public static Dictionary<string, string> getEnumerationString(string table, string attribute)
    {
        SqlConnection con = new
        SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[get_enumeration_varchar]";
        cmd.Parameters.Add("@table_name", SqlDbType.VarChar, 30).Value = table;
        cmd.Parameters.Add("@attr_name", SqlDbType.VarChar, 50).Value = attribute;
        cmd.Connection = con;

        Dictionary<string, string> enumeration = new Dictionary<string, string>();

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();

            while (rdr.Read())
            {
                string value = (string) rdr["Value"];
                string label = (string) rdr["Label"];

                enumeration.Add(value, label);
            }
        }
        catch (SqlException ex)
        {
        }
    }
}

```

```

        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }

        return enumeration;
    }

    public static Dictionary<byte, string> getEnumerationByte(string table, string attribute)
    {
        SqlConnection con = new
        SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[get_enumeration_int]";
        cmd.Parameters.Add("@table_name", SqlDbType.VarChar, 30).Value = table;
        cmd.Parameters.Add("@attr_name", SqlDbType.VarChar, 50).Value = attribute;
        cmd.Connection = con;

        Dictionary<byte, string> enumeration = new Dictionary<byte, string>();

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();

            while (rdr.Read())
            {
                byte value = (byte)rdr["Value"];
                string label = (string)rdr["Label"];

                enumeration.Add(value, label);
            }
        }
        catch (SqlException ex)
        {
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }

        return enumeration;
    }

    public static Dictionary<byte, string> getEnumerationByteWithNoData(string table, string
attribute, byte noData)
    {
        Dictionary<byte, string> dictionary = getEnumerationByte(table, attribute);
        dictionary.Add(noData, "");

        return dictionary;
    }

    public static Dictionary<decimal, string> getEnumerationDecimal(string table, string
attribute)
    {
        SqlConnection con = new
        SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();

```

```

cmd.CommandType = CommandType.StoredProcedure;
cmd.CommandText = "[dbo].[get_enumeration_decimal]";
cmd.Parameters.Add("@table_name", SqlDbType.VarChar, 30).Value = table;
cmd.Parameters.Add("@attr_name", SqlDbType.VarChar, 50).Value = attribute;
cmd.Connection = con;

Dictionary<decimal, string> enumeration = new Dictionary<decimal, string>();

try
{
    con.Open();
    SqlDataReader rdr = cmd.ExecuteReader();

    while (rdr.Read())
    {
        decimal value = (decimal)rdr["Value"];
        string label = (string)rdr["Label"];

        enumeration.Add(value, label);
    }
}
catch (SqlException ex)
{
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}

return enumeration;
}

public static Dictionary<decimal, string> getEnumerationDecimalWithNoData(string table, string
attribute, decimal noData)
{
    Dictionary<decimal, string> dictionary = getEnumerationDecimal(table, attribute);
    dictionary.Add(noData, "");

    return dictionary;
}

public static object getByteOrNull(string value)
{
    return getByteOrNullWithNoData(value, "");
}

public static object getByteOrNullWithNoData(string value, string noData)
{
    byte b;
    if (value == noData || !byte.TryParse(value, out b))
    {
        return DBNull.Value;
    }
    else
    {
        return b;
    }
}

public static object getShortOrNull(string value)
{
    short s;
    if (value == "" || !short.TryParse(value, out s))
    {

```

```

        return DBNull.Value;
    }
    else
    {
        return s;
    }
}

public static object getBitOrNull(string value)
{
    return (value == "2") ? DBNull.Value : (object)Convert.ToBoolean(int.Parse(value));
}

public static object getStringOrNull(string value)
{
    return (value == "") ? DBNull.Value : (object)value;
}

public static object getDecimalOrNull(string value)
{
    return getDecimalOrNullWithNoData(value, "");
}

/// <summary>
/// Accepts a point or a comma for decimals.
/// </summary>
public static object getDecimalOrNullWithNoData(string value, string noData)
{
    value = value.Replace(',', '.');
    decimal d;
    if (value == noData || !decimal.TryParse(value, NumberStyles.Any,
CultureInfo.InvariantCulture, out d))
    {
        return DBNull.Value;
    }
    else
    {
        return d;
    }
}

public static string getDropYesNoValue(object value)
{
    return value == DBNull.Value ? "2" : ((byte)value).ToString();
}

public static string getDropBitValue(object value)
{
    if (value == DBNull.Value)
    {
        return "2";
    }
    else if ((bool)value == false)
    {
        return "0";
    }
    else if ((bool)value == true)
    {
        return "1";
    }
    else
    {
        return "2";
    }
}

public static string getDropMultiValue(object value)
{
    return value == DBNull.Value ? "1" : ((byte)value).ToString();
}

```

```

    }

    public static string getDropMultiValueWithNoData(object value, string noData)
    {
        return value == DBNull.Value ? noData : ((byte)value).ToString();
    }

    public static string getDropDecimalValue(object value)
    {
        return value == DBNull.Value ? "2.0" : ((decimal)value).ToString();
    }

    public static string getDropDecimalValueWithNoData(object value, string noData)
    {
        return value == DBNull.Value ? noData : ((decimal)value).ToString();
    }

    public static string getTextDecimalValue(object value)
    {
        return value == DBNull.Value ? "" : ((decimal)value).ToString();
    }

    public static string getTextStringValue(object value)
    {
        return value == DBNull.Value ? "" : value.ToString();
    }

    public static string getTextByteValue(object value)
    {
        return value == DBNull.Value ? "" : ((byte)value).ToString();
    }

    public static string getTextShortValue(object value)
    {
        return value == DBNull.Value ? "" : ((short)value).ToString();
    }

    public static void saveMultiChoice(List<string> selectedValues, string attributeName, int appointmentId, string actor)
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[update_exam_int_attribute]";
        cmd.Parameters.Add("@IdWizyta", SqlDbType.Int).Value = appointmentId;
        cmd.Parameters.Add("@attr_name", SqlDbType.VarChar, 50).Value = attributeName;

        DataTable dataTable = new DataTable();
        DataColumn dataColumn = new DataColumn();
        dataColumn = new DataColumn("Value", typeof(System.Int32));
        dataTable.Columns.Add(dataColumn);
        foreach (string value in selectedValues)
        {
            DataRow dataRow = dataTable.NewRow();
            dataRow[0] = value;
            dataTable.Rows.Add(dataRow);
        }
        SqlParameter param = cmd.Parameters.AddWithValue("@values", dataTable);
        cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = actor;
        cmd.Parameters.Add("@result", SqlDbType.Int);
        cmd.Parameters["@result"].Direction = ParameterDirection.Output;
        cmd.Connection = con;

        int success = 0;
        try
        {
            con.Open();
            cmd.ExecuteNonQuery();

```

```

        success = (int)cmd.Parameters["@result"].Value;
    }
    catch (SqlException ex)
    {
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }
}

public static List<string> getMultiChoice(string attributeName, int appointmentId)
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select * from dbo.get_exam_multi_values('" + attributeName + "', " +
appointmentId + ")";
    cmd.Connection = con;

    List<string> selectedValues = new List<string>();
    try
    {
        con.Open();
        SqlDataReader rdr = cmd.ExecuteReader();
        while (rdr.Read())
        {
            string value = rdr["Wartosc"].ToString();
            selectedValues.Add(value);
        }
    }
    catch (SqlException ex)
    {
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }

    return selectedValues;
}
}

```

App_Code\Utils.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI.WebControls;

```

/// <summary>

/// Summary description for Utils

/// </summary>

```
public class Utils
```

```
{
```

```
    public Utils()
```

```
    {
```

```
        //
```

```
        // TODO: Add constructor logic here
```

```

        }
    }

    public static void setSelectedCheckListItems(List<string> selectedValues, CheckBoxList
checkList)
    {
        foreach (string value in selectedValues)
        {
            ListItem listItem = checkList.Items.FindByValue(value);
            if (listItem != null)
            {
                listItem.Selected = true;
            }
        }
    }

    public static List<string> getSelectedCheckListItems(CheckBoxList checkList)
    {
        List<string> selectedValues = new List<string>();
        foreach (ListItem listItem in checkList.Items)
        {
            if (listItem.Selected)
            {
                selectedValues.Add(listItem.Value);
            }
        }

        return selectedValues;
    }
}

```

About.aspx

```

<%@ Page Title="About Us" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="About.aspx.cs" Inherits="About" %>

```

```

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <h2>
        0 stronie
    </h2>
    <p>
        Opis.
    </p>
</asp:Content>

```

About.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class About : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}

```

AppointmentForm.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="AppointmentForm.aspx.cs" Inherits="AppointmentForm" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">

```



```

<link href="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/themes/base/jquery-ui.css"
rel="stylesheet" type="text/css" />
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js"
type="text/javascript"></script>
<script src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8/jquery-ui.min.js"
type="text/javascript"></script>
<script
src="http://ajax.googleapis.com/ajax/libs/jqueryui/1.8.1/i18n/jquery.ui.datepicker-pl.min.js"
type="text/javascript"></script>
<script type="text/javascript" language="javascript">
$(document).ready(function () {
    $("#<%= textDateIn.ClientID %>").datepicker({
        showOn: 'button',
        buttonImage: 'img/calendar_datepicker.png',
        buttonImageOnly: true,
        changeMonth: true,
        changeYear: true,
        showAnim: 'slideDown',
        duration: 'fast'
    });
});
</script>
<script type="text/javascript" language="javascript">
$(document).ready(function () {
    $("#<%= textDateSurgery.ClientID %>").datepicker({
        showOn: 'button',
        buttonImage: 'img/calendar_datepicker.png',
        buttonImageOnly: true,
        changeMonth: true,
        changeYear: true,
        showAnim: 'slideDown',
        duration: 'fast'
    });
});
</script>
<script type="text/javascript" language="javascript">
$(document).ready(function () {
    $("#<%= textDateOut.ClientID %>").datepicker({
        showOn: 'button',
        buttonImage: 'img/calendar_datepicker.png',
        buttonImageOnly: true,
        changeMonth: true,
        changeYear: true,
        showAnim: 'slideDown',
        duration: 'fast'
    });
});
</script>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
<h2>Dane demograficzne i objawy ruchowe</h2>
<br />
<table ID="Table1" runat="server">
<tr>
<td>
<table ID="Table2" runat="server">
<tr>
<td><asp:Label ID="labelPatientNumber" runat="server"
Text=""></asp:Label></td>
</tr>
<tr>
<td>Rodzaj wizyty:</td>
<td>
<asp:DropDownList ID="dropAppointmentType" runat="server">
</asp:DropDownList>
</td>
</tr>
<tr>
<td>Data przyjęcia:</td>

```

```

        <td>
            <asp:TextBox ID="textDateIn" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>Data wypisu:</td>
        <td>
            <asp:TextBox ID="textDateOut" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="labelDateSurgery" runat="server" Text="Data
operacji:"></asp:Label>
        </td>
        <td>
            <asp:TextBox ID="textDateSurgery" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>Masa ciała:</td>
        <td>
            <asp:TextBox ID="textMasaCiala" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="labelEducation" runat="server"
Text="Wykształcenie:"></asp:Label>
        </td>
        <td>
            <asp:DropDownList ID="dropEducation" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="labelFamily" runat="server"
Text="Rodzinność:"></asp:Label>
        </td>
        <td>
            <asp:DropDownList ID="dropFamily" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Miesiąc i rok zachorowania:</td>
        <td>
            <asp:DropDownList ID="dropMiesiacZachorowania" runat="server">
            </asp:DropDownList>
            <asp:DropDownList ID="dropRokZachorowania" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Miesiąc i rok badania:</td>
        <td>
            <asp:DropDownList ID="dropMiesiacBadania" runat="server">
            </asp:DropDownList>
            <asp:DropDownList ID="dropRokBadania" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>

```

```

objaw:"></asp:Label>
        <asp:Label ID="labelSymptom" runat="server" Text="Pierwszy
    </td>
    <td>
        <asp:DropDownList ID="dropSymptom" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Drżenie:</td>
    <td>
        <asp:DropDownList ID="dropDrzenie" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Sztywność:</td>
    <td>
        <asp:DropDownList ID="dropSztywnosc" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Spowolnienie:</td>
    <td>
        <asp:DropDownList ID="dropSpowolnienie" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Objawy inne:</td>
    <td>
        <asp:DropDownList ID="dropObjawy" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Objawy inne. Jakież?:</td>
    <td>
        <asp:TextBox ID="textObjawy" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>
        <asp:Label ID="labelTimeSymptom" runat="server" Text="Czas od początku
objawów do<br />włączenia LDopy [mies.]:"></asp:Label>
    </td>
    <td>
        <asp:TextBox ID="textTimeSymptom" runat="server"></asp:TextBox>
        <!--
        <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server"
            ErrorMessage="*"
ControlToValidate="textTimeSymptom"></asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator Runat="server"
ID="RegularExpressionValidator1"
            ControlToValidate="textTimeSymptom"

```

```

                ErrorMessage="Proszę podać liczbę od 0 do 255."
                ValidationExpression="^([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])$" ></asp:RegularExpressionValidator>
                --%>
            </td>
        </tr>
        <tr>
            <td>Dyskinezy obecnie:</td>
            <td>
                <asp:DropDownList ID="dropDiskinesia" runat="server"
                    onselectedindexchanged="dropDiskinesia_SelectedIndexChanged"
                    AutoPostBack="True">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="1">tak</asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Czas dyskinez od lat:</td>
            <td>
                <asp:TextBox ID="textTimeDiskinesia" runat="server"></asp:TextBox>
                <%--
                <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server"
                    ErrorMessage="*"
ControlToValidate="textTimeDiskinesia"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator ID="RegularExpressionValidator2"
runat="server"
                    ErrorMessage="Proszę podać liczbę od 0 do 99.9."
                    ValidationExpression="^[0-9]{1,2}(?:\.[0-9]{1})? $"
ControlToValidate="textTimeDiskinesia"></asp:RegularExpressionValidator>
                --%>
            </td>
        </tr>
        <tr>
            <td>Fluktuacje obecnie:</td>
            <td>
                <asp:DropDownList ID="dropFluctuations" runat="server"
                    onselectedindexchanged="dropFluctuations_SelectedIndexChanged"
                    AutoPostBack="True">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="1">tak</asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Fluktuacje od lat:</td>
            <td>
                <asp:TextBox ID="textYearsFluctuations" runat="server"></asp:TextBox>
                <%--
                <asp:RequiredFieldValidator ID="RequiredFieldValidator3"
runat="server"
                    ErrorMessage="*"
ControlToValidate="textYearsFluctuations"></asp:RequiredFieldValidator>
                <asp:RegularExpressionValidator ID="RegularExpressionValidator3"
runat="server"
                    ErrorMessage="Proszę podać liczbę od 0 do 99.9."
                    ValidationExpression="^[0-9]{1,2}(?:\.[0-9]{1})? $"
ControlToValidate="textYearsFluctuations"></asp:RegularExpressionValidator>
                --%>
            </td>
        </tr>
        <tr>
            <td>Czas dyskinez:</td>
            <td>

```

```

        <asp:TextBox ID="textCzasDyskinez" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Czas OFF:</td>
    <td>
        <asp:TextBox ID="textCzasOFF" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Poprawa po LDopie:</td>
    <td>
        <asp:DropDownList ID="dropPoprawa" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
</table>
</td>
<td align="center" valign="top" style="padding-left: 100px;">
    <table ID="Table3" runat="server">
        <tr>
            <td>
                <asp:Label ID="labelAppointment" runat="server" Text=""></asp:Label>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Button ID="buttonPartB" runat="server" Text="Objawy i
epidemiologia"
                Width="265px" onclick="buttonPartB_Click" CausesValidation="false"
            />
            </td>
        </tr>
        <tr>
            <td>
                <asp:Button ID="buttonPartC" runat="server" Text="Przyjmowane leki"
                Width="265px" onclick="buttonPartC_Click" CausesValidation="false"
            />
            </td>
        </tr>
        <tr>
            <td>
                <asp:Button ID="buttonPartE" runat="server" Text="Kwestionariusz
zaburzeń autonomicznych"
                Width="265px" onclick="buttonPartE_Click" CausesValidation="false"
            />
            </td>
        </tr>
        <tr>
            <td>
                <asp:Button ID="buttonPartF" runat="server" Text="UPDRS i skale oceny
klinicznej"
                Width="265px" onclick="buttonPartF_Click" CausesValidation="false"
            />
            </td>
        </tr>
        <tr>
            <td>
                <asp:Button ID="buttonPartG" runat="server" Text="Badania i testy
psychologiczne"
                Width="265px" onclick="buttonPartG_Click" CausesValidation="false"
            />
            </td>
        </tr>
    </table>
</td>
</tr>
</table>

```

```

        <asp:Button ID="buttonParth" runat="server" Text="Badania dodatkowe"
            Width="265px" onclick="buttonParth_Click" CausesValidation="false"
        />
    </td>
</tr>
</table>
</td>
</tr>
</table>
<asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
<br />
<asp:Button ID="buttonOK" runat="server" Text="Zatwierdź"
    onclick="buttonOK_Click" />
<asp:Button ID="buttonCancel" runat="server" Text="Powrót"
    CausesValidation="False" onclick="buttonCancel_Click" />
</asp:Content>

```

AppointmentForm.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

public partial class AppointmentForm : System.Web.UI.Page
{
    private bool update = false;
    private string patientNumber;
    private static string DATE_FORMAT = "yyyy-MM-dd";
    private static byte NO_DATA = 100;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["AppointmentType"] != null && Session["PatientNumber"] != null &&
            Session["Update"] != null)
        {
            patientNumber = Session["PatientNumber"].ToString();
            labelPatientNumber.Text = "Numer pacjenta: " + patientNumber;

            dropAppointmentType.SelectedValue = Session["AppointmentType"].ToString();

            update = (bool)Session["Update"];
            if (!IsPostBack)
            {
                initControls();
                if (update)
                {
                    loadAppointment((int)Session["AppointmentId"]);
                }
                else
                {
                    initAppointment();
                }
            }
        }
        else
        {
            Response.Redirect("~/Main.aspx");
        }
    }

    private void initControls()
    {
        textDateIn.Text = DateTime.Now.ToString(DATE_FORMAT);
        textDateSurgery.Text = DateTime.Now.ToString(DATE_FORMAT);
    }
}

```

```

textDateOut.Text = DateTime.Now.ToString(DATE_FORMAT);
// date not setting when disabled textbox?
//textDateIn.Enabled = false;
//textDateSurgery.Enabled = false;
//textDateOut.Enabled = false;

dropAppointmentType.DataSource = DatabaseProcedures.getEnumerationDecimal("Wizyta",
"RodzajWizyty");
dropAppointmentType.DataTextField = "Value";
dropAppointmentType.DataValueField = "Key";
dropAppointmentType.DataBind();
dropAppointmentType.Enabled = false;
toggleButtons(update);

int minYear = 1920;
int currentYear = DateTime.Now.Year;
for (int i = minYear; i <= currentYear; i++)
{
    dropRokZachorowania.Items.Add(new ListItem(" " + i, " " + i));
}
dropRokZachorowania.SelectedIndex = currentYear - minYear - 10;

int startYear = 2013;
for (int i = startYear; i <= currentYear; i++)
{
    dropRokBadania.Items.Add(new ListItem(" " + i, " " + i));
}
dropRokBadania.SelectedIndex = currentYear - startYear;

for (int i = 1; i <= 12; i++)
{
    dropMiesiacZachorowania.Items.Add(new ListItem(" " + i, " " + i));
    dropMiesiacBadania.Items.Add(new ListItem(" " + i, " " + i));
}
dropMiesiacBadania.SelectedIndex = DateTime.Now.Month - 1;

dropEducation.DataSource = DatabaseProcedures.getEnumerationByteWithNoData("Wizyta",
"Wyksztalcenie", NO_DATA);
dropEducation.DataTextField = "Value";
dropEducation.DataValueField = "Key";
dropEducation.DataBind();

dropSymptom.DataSource = DatabaseProcedures.getEnumerationByteWithNoData("Wizyta",
"PierwszyObjaw", NO_DATA);
dropSymptom.DataTextField = "Value";
dropSymptom.DataValueField = "Key";
dropSymptom.DataBind();

if ((Session["AppointmentType"].ToString() != "0.0" &&
Session["PatientNumber"].ToString().Contains("POP") == false) ||
(Session["AppointmentType"].ToString() != "0.0" &&
Session["AppointmentType"].ToString() != "0.5" &&
Session["PatientNumber"].ToString().Contains("POP") == true))
{
    textDateSurgery.Visible = false;
    labelDateSurgery.Visible = false;
    dropEducation.Visible = false;
    labelEducation.Visible = false;
    dropFamily.Visible = false;
    labelFamily.Visible = false;
    dropSymptom.Visible = false;
    labelSymptom.Visible = false;
    textTimeSymptom.Visible = false;
    labelTimeSymptom.Visible = false;
}
}

private void saveAppointment()
{

```

```

        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
SqlCommand cmd = new SqlCommand();
cmd.CommandType = CommandType.StoredProcedure;
cmd.CommandText = "[dbo].[update_examination_questionnaire_partA]";
cmd.Parameters.Add("@NumerPacjenta", SqlDbType.VarChar, 20).Value = patientNumber;
SqlParameter examinationTypeDecimal = new SqlParameter("@RodzajWizyty",
SqlDbType.Decimal);
examinationTypeDecimal.Precision = 2;
examinationTypeDecimal.Scale = 1;
examinationTypeDecimal.Value = decimal.Parse(dropAppointmentType.Selected.Value);
cmd.Parameters.Add(examinationTypeDecimal);
cmd.Parameters.Add("@DataPrzyjecia", SqlDbType.Date).Value =
DateTime.Parse(textDateIn.Text);
cmd.Parameters.Add("@DataOperacji", SqlDbType.Date).Value =
DateTime.Parse(textDateSurgery.Text);
cmd.Parameters.Add("@DataWypisu", SqlDbType.Date).Value =
DateTime.Parse(textDateOut.Text);
SqlParameter masaCialaDecimal = new SqlParameter("@MasaCiala", SqlDbType.Decimal);
masaCialaDecimal.Precision = 4;
masaCialaDecimal.Scale = 1;
masaCialaDecimal.Value = DatabaseProcedures.getDecimalOrNull(textMasaCiala.Text);
cmd.Parameters.Add(masaCialaDecimal);
cmd.Parameters.Add("@Wyksztalcenie", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNullWithNoData(dropEducation.Selected.Value, NO_DATA.ToString());
cmd.Parameters.Add("@Rodzinnosc", SqlDbType.TinyInt).Value =
byte.Parse(dropFamily.Selected.Value);
cmd.Parameters.Add("@RokZachorowania", SqlDbType.SmallInt).Value =
short.Parse(dropRokZachorowania.Selected.Value);
cmd.Parameters.Add("@MiesiacZachorowania", SqlDbType.TinyInt).Value =
byte.Parse(dropMiesiacZachorowania.Selected.Value);
cmd.Parameters.Add("@RokBadania", SqlDbType.SmallInt).Value =
short.Parse(dropRokBadania.Selected.Value);
cmd.Parameters.Add("@MiesiacBadania", SqlDbType.TinyInt).Value =
byte.Parse(dropMiesiacBadania.Selected.Value);
cmd.Parameters.Add("@PierwszyObjaw", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNullWithNoData(dropSymptom.Selected.Value, NO_DATA.ToString());
cmd.Parameters.Add("@Drzenie", SqlDbType.TinyInt).Value =
byte.Parse(dropDrzenie.Selected.Value);
cmd.Parameters.Add("@Sztynosc", SqlDbType.TinyInt).Value =
byte.Parse(dropSztynosc.Selected.Value);
cmd.Parameters.Add("@Spowolnienie", SqlDbType.TinyInt).Value =
byte.Parse(dropSpowolnienie.Selected.Value);
cmd.Parameters.Add("@ObjawyInne", SqlDbType.TinyInt).Value =
byte.Parse(dropObjawy.Selected.Value);
cmd.Parameters.Add("@ObjawyInneJakie", SqlDbType.VarChar, 80).Value =
textObjawy.Text.ToString();
cmd.Parameters.Add("@CzasOdPoczObjDoWlDopy", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textTimeSymptom.Text);
cmd.Parameters.Add("@DyskinezyObecnie", SqlDbType.TinyInt).Value =
byte.Parse(dropDiskinesia.Selected.Value);
SqlParameter dyskinesiaDecimal = new SqlParameter("@DyskinezyOdLat", SqlDbType.Decimal);
dyskinesiaDecimal.Precision = 3;
dyskinesiaDecimal.Scale = 1;
dyskinesiaDecimal.Value = DatabaseProcedures.getDecimalOrNull(textTimeDiskinesia.Text);
cmd.Parameters.Add(dyskinesiaDecimal);
cmd.Parameters.Add("@FluktuacjeObecnie", SqlDbType.TinyInt).Value =
byte.Parse(dropFluctuations.Selected.Value);
SqlParameter fluctuationsDecimal = new SqlParameter("@FluktuacjeOdLat",
SqlDbType.Decimal);
fluctuationsDecimal.Precision = 3;
fluctuationsDecimal.Scale = 1;
fluctuationsDecimal.Value =
DatabaseProcedures.getDecimalOrNull(textYearsFluctuations.Text);
cmd.Parameters.Add(fluctuationsDecimal);
SqlParameter czasDyskinezDecimal = new SqlParameter("@CzasDyskinez", SqlDbType.Decimal);
czasDyskinezDecimal.Precision = 3;
czasDyskinezDecimal.Scale = 1;
czasDyskinezDecimal.Value = DatabaseProcedures.getDecimalOrNull(textCzasOFF.Text);

```



```

cmd.Parameters.Add(czasDyskinezDecimal);
SqlParameter czasOFFDecimal = new SqlParameter("@CzasOFF", SqlDbType.Decimal);
czasOFFDecimal.Precision = 3;
czasOFFDecimal.Scale = 1;
czasOFFDecimal.Value = DatabaseProcedures.getDecimalOrNull(textCzasOFF.Text);
cmd.Parameters.Add(czasOFFDecimal);
cmd.Parameters.Add("@PoprawaPoLDopie", SqlDbType.TinyInt).Value =
byte.Parse(dropPoprawa.SelectedText);
cmd.Parameters.Add("@allow_update_existing", SqlDbType.Bit).Value = update;
cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
cmd.Parameters.Add("@result", SqlDbType.Int);
cmd.Parameters["@result"].Direction = ParameterDirection.Output;
cmd.Parameters.Add("@visit_id", SqlDbType.Int);
cmd.Parameters["@visit_id"].Direction = ParameterDirection.Output;
cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
cmd.Parameters["@message"].Direction = ParameterDirection.Output;
cmd.Connection = con;

int success = 0;
try
{
    con.Open();
    cmd.ExecuteNonQuery();
    success = (int)cmd.Parameters["@result"].Value;

    if (success == 0)
    {
        if (!update)
        {
            toggleButtons(true);
            Session["AppointmentId"] = (int)cmd.Parameters["@visit_id"].Value;
            Session["Update"] = true;
        }
    }
    else
    {
        labelMessage.Text = (string)cmd.Parameters["@message"].Value;
    }
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}
}

private void loadAppointment(int appointmentId)
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
SqlCommand cmd = new SqlCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "select DataPrzyjecia, DataOperacji, DataWypisu, Wykształcenie,
MasaCiała, Rodzinnosc, RokZachorowania, MiesiacZachorowania, RokBadania, MiesiacBadania, " +
"PierwszyObjaw, Drzenie, Sztynosc, Spowolnienie, ObjawyInne, ObjawyInneJakie,
CzasOdPoczObjDoWLLDopy, DyskinezyObecnie, DyskinezyOdLat, FluktuacjeObecnie, FluktuacjeOdLat,
CzasDyskinez, CzasOFF, PoprawaPoLDopie " +
" from Wizyta where IdWizyta = " + appointmentId;
cmd.Connection = con;

try
{

```

```

con.Open();
SqlDataReader rdr = cmd.ExecuteReader();
while (rdr.Read())
{
    textDateIn.Text = ((DateTime)rdr["DataPrzyjecia"]).ToString(DATE_FORMAT);
    textDateSurgery.Text = ((DateTime)rdr["DataOperacji"]).ToString(DATE_FORMAT);
    textDateOut.Text = ((DateTime)rdr["DataWypisu"]).ToString(DATE_FORMAT);
    textMasaCiala.Text = DatabaseProcedures.getTextDecimalValue(rdr["MasaCiala"]);
    dropEducation.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Wykształcenie"], NO_DATA.ToString());
    dropFamily.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["Rodzinnosc"]);
    dropRokZachorowania.SelectedValue = ((short)rdr["RokZachorowania"]).ToString();
    dropMiesiacZachorowania.SelectedValue =
((byte)rdr["MiesiacZachorowania"]).ToString();
    dropRokBadania.SelectedValue = ((short)rdr["RokBadania"]).ToString();
    dropMiesiacBadania.SelectedValue = ((byte)rdr["MiesiacBadania"]).ToString();
    dropSymptom.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["PierwszyObjaw"], NO_DATA.ToString());
    dropDrzenie.SelectedValue = DatabaseProcedures.getDropYesNoValue(rdr["Drzenie"]);
    dropSztywnosc.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["Sztywnosc"]);
    dropSpowolnienie.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["Spowolnienie"]);
    dropObjawy.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["ObjawyInne"]);
    textObjawy.Text = DatabaseProcedures.getTextStringValue(rdr["ObjawyInneJakie"]);
    textTimeSymptom.Text =
DatabaseProcedures.getTextByteValue(rdr["CzasOdPoczObjDoWlLDopy"]);
    dropDiskinesia.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["DyskinezyObecnie"], NO_DATA.ToString());
    if (dropDiskinesia.SelectedValue == "0")
    {
        textTimeDiskinesia.Enabled = false;
        //RequiredFieldValidator2.Enabled = false;
    }
    else
    {
        textTimeDiskinesia.Text =
DatabaseProcedures.getTextDecimalValue(rdr["DyskinezyOdLat"]);
        //RequiredFieldValidator2.Enabled = true;
    }
    dropFluctuations.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["FluktuacjeObecnie"], NO_DATA.ToString());
    if (dropFluctuations.SelectedValue == "0")
    {
        textYearsFluctuations.Enabled = false;
        //RequiredFieldValidator3.Enabled = false;
    }
    else
    {
        textYearsFluctuations.Text =
DatabaseProcedures.getTextDecimalValue(rdr["FluktuacjeOdLat"]);
        //RequiredFieldValidator3.Enabled = true;
    }
    textCzasDyskinez.Text =
DatabaseProcedures.getTextDecimalValue(rdr["CzasDyskinez"]);
    textCzasOFF.Text = DatabaseProcedures.getTextDecimalValue(rdr["CzasOFF"]);
    dropPoprawa.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["PoprawaPoLDopie"]);
}
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
}

```

```

        if (con != null)
        {
            con.Close();
        }
    }
}

private void initAppointment()
{
    dropEducation.SelectedValue = NO_DATA.ToString();
    dropSymptom.SelectedValue = NO_DATA.ToString();
    textTimeDiskinesia.Enabled = false;
    //RequiredFieldValidator2.Enabled = false;
    textYearsFluctuations.Enabled = false;
    //RequiredFieldValidator3.Enabled = false;
}

protected void dropDiskinesia_SelectedIndexChanged(object sender, EventArgs e)
{
    if (byte.Parse(dropDiskinesia.SelectedValue) == 1)
    {
        textTimeDiskinesia.Enabled = true;
        //RequiredFieldValidator2.Enabled = true;
    }
    else
    {
        textTimeDiskinesia.Enabled = false;
        textTimeDiskinesia.Text = "";
        //RequiredFieldValidator2.Enabled = false;
    }
}

protected void dropFluctuations_SelectedIndexChanged(object sender, EventArgs e)
{
    if (byte.Parse(dropFluctuations.SelectedValue) == 1)
    {
        textYearsFluctuations.Enabled = true;
        //RequiredFieldValidator3.Enabled = true;
    }
    else
    {
        textYearsFluctuations.Enabled = false;
        textYearsFluctuations.Text = "";
        //RequiredFieldValidator3.Enabled = false;
    }
}

protected void buttonOK_Click(object sender, EventArgs e)
{
    saveAppointment();
}

protected void buttonCancel_Click(object sender, EventArgs e)
{
    Response.Redirect("~/AppointmentList.aspx");
}

protected void buttonPartB_Click(object sender, EventArgs e)
{
    Response.Redirect("~/PartBForm.aspx");
}

protected void buttonPartC_Click(object sender, EventArgs e)
{
    Response.Redirect("~/PartCForm.aspx");
}

protected void buttonPartE_Click(object sender, EventArgs e)
{
    Response.Redirect("~/PartEForm.aspx");
}

protected void buttonPartF_Click(object sender, EventArgs e)

```

```

    {
        Response.Redirect("~/PartFForm.aspx");
    }
    protected void buttonPartG_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/PartGForm.aspx");
    }
    protected void buttonPartH_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/PartHForm.aspx");
    }
}

private void toggleButtons(bool enable)
{
    if (enable)
    {
        Session["AppointmentName"] = dropAppointmentType.SelectedItem.ToString();
        labelAppointment.Text = "Badania dla wizyty: " + Session["AppointmentName"];
    }
    else
    {
        labelAppointment.Text = "";
    }
    buttonPartB.Enabled = enable;
    buttonPartC.Enabled = enable;
    buttonPartE.Enabled = enable;
    buttonPartF.Enabled = enable;
    buttonPartG.Enabled = enable;
    buttonPartH.Enabled = enable;
}
}

```

AppointmentList.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="AppointmentList.aspx.cs" Inherits="AppointmentList" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <asp:Label ID="labelPatientNumber" runat="server" Text="labelPatientNumber"></asp:Label>
    <asp:Table ID="tableAppointments" runat="server" Width="450px">
        <asp:TableHeaderRow>
        </asp:TableHeaderRow>
    </asp:Table>
    <asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
    <asp:Button ID="buttonBack" runat="server" Text="Powrót" onclick="buttonBack_Click" />
</asp:Content>

```

AppointmentList.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

public partial class AppointmentList : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["PatientNumber"] != null)
        {
            labelPatientNumber.Text = "Numer pacjenta: " + Session["PatientNumber"].ToString();
            Dictionary<decimal, string> appointmentTypes =
            DatabaseProcedures.getEnumerationDecimal("Wizyta", "RodzajWizyty");

```

```

        List<AppointmentSelection> existingAppointments =
getAppointments(Session["PatientNumber"].ToString(), appointmentTypes);

        TableHeaderRow header = new TableHeaderRow();
        TableHeaderCell headerCell11 = new TableHeaderCell();
        TableHeaderCell headerCell12 = new TableHeaderCell();
        header.Cells.Add(headerCell11);
        header.Cells.Add(headerCell12);
        tableAppointments.Rows.Add(header);
        headerCell11.Text = "Data";
        headerCell12.Text = "Typ wizyty";

        foreach (KeyValuePair<decimal, string> appointmentType in appointmentTypes)
        {
            TableRow row = new TableRow();
            TableCell cell11 = new TableCell();
            TableCell cell12 = new TableCell();
            TableCell cell13 = new TableCell();
            row.Cells.Add(cell11);
            row.Cells.Add(cell12);
            row.Cells.Add(cell13);
            tableAppointments.Rows.Add(row);

            bool exists = false;
            foreach (AppointmentSelection existingAppointment in existingAppointments)
            {
                if (existingAppointment.typeKey == appointmentType.Key)
                {
                    exists = true;
                    cell11.Controls.Add(existingAppointment.labelDate);
                    cell12.Controls.Add(existingAppointment.labelType);
                    cell13.Controls.Add(existingAppointment.buttonEdit);
                    cell13.Controls.Add(existingAppointment.buttonDelete);
                }
            }

            if (!exists)
            {
                AppointmentSelection appointment = new
AppointmentSelection(appointmentType.Value, appointmentType.Key, this);
                cell12.Controls.Add(appointment.labelType);
                cell13.Controls.Add(appointment.buttonNew);
            }
        }
    }

    private List<AppointmentSelection> getAppointments(string patientNumber, Dictionary<decimal,
string> appointmentTypes)
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "select IdWizyta, DataPrzyjecia, RodzajWizyty from Wizyta where exists
(select IdPacjent from Pacjent where Wizyta.IdPacjent = Pacjent.IdPacjent and
Pacjent.NumerPacjenta = '" + patientNumber + "')";
        cmd.Connection = con;

        List<AppointmentSelection> list = new List<AppointmentSelection>();

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();

            while (rdr.Read())
            {
                decimal typeKey = (decimal)rdr["RodzajWizyty"];

```

```

        string typeValue = "";
        appointmentTypes.TryGetValue(typeKey, out typeValue);
        AppointmentSelection appointment = new AppointmentSelection((int)rdr["IdWizyta"],
(DateTime)rdr["DataPrzyjecia"], typeValue, typeKey, this);
        list.Add(appointment);
    }
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}

return list;
}

private class AppointmentSelection
{
    private int idAppointment;
    public decimal typeKey;
    public Label labelDate;
    public Label labelType;
    public Button buttonNew;
    public Button buttonEdit;
    public Button buttonDelete;
    private AppointmentList page;

    public AppointmentSelection(string typeValue, decimal typeKey, AppointmentList page)
    {
        this.page = page;
        labelType = new Label();
        labelType.Text = typeValue;
        this.typeKey = typeKey;
        buttonNew = new Button();
        buttonNew.Text = "Utwórz";
        buttonNew.Click += new System.EventHandler(buttonNew_Click);
    }

    public AppointmentSelection(int id, DateTime date, string typeValue, decimal typeKey,
AppointmentList page)
    {
        this.page = page;
        idAppointment = id;
        labelDate = new Label();
        labelDate.Text = date.ToString("yyyy-MM-dd");
        labelType = new Label();
        labelType.Text = typeValue;
        this.typeKey = typeKey;
        buttonEdit = new Button();
        buttonEdit.Text = "Edytuj";
        buttonEdit.Click += new System.EventHandler(buttonEdit_Click);
        buttonDelete = new Button();
        buttonDelete.Text = "Usuń";
        buttonDelete.Click += new System.EventHandler(buttonDelete_Click);
    }

    protected void buttonNew_Click(object sender, EventArgs e)
    {
        page.Session["PatientNumber"] = page.Session["PatientNumber"];
        page.Session["AppointmentType"] = typeKey;
        page.Session["Update"] = false;
        page.Response.Redirect("~/AppointmentForm.aspx");
    }
}

```

```

    }

    protected void buttonShowExaminations_Click(object sender, EventArgs e)
    {
        page.Session["AppointmentId"] = idAppointment;
        page.Response.Redirect("~/ExaminationList.aspx");
    }

    protected void buttonEdit_Click(object sender, EventArgs e)
    {
        page.Session["PatientNumber"] = page.Session["PatientNumber"];
        page.Session["AppointmentType"] = typeKey;
        page.Session["Update"] = true;
        page.Session["AppointmentId"] = idAppointment;
        page.Response.Redirect("~/AppointmentForm.aspx");
    }

    protected void buttonDelete_Click(object sender, EventArgs e)
    {
        SqlConnection con = new
        SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "delete from Wizyta where IdWizyta = " + idAppointment + ";delete
        from Badanie where IdWizyta = " + idAppointment;
        cmd.Connection = con;

        List<string> list = new List<string>();

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
        }
        catch (SqlException ex)
        {
            page.LabelMessage.Text = ex.Message;
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
            page.Response.Redirect(page.Request.RawUrl);
        }
    }

    protected void buttonBack_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/Main.aspx");
    }
}

```

Default.aspx

```

<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeFile="Default.aspx.cs" Inherits="_Default" %>

```

```

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <h2>
        Strona Powitalna
    </h2>
    <p>
        <a href="Main.aspx" title="Main.aspx">Lista pacjentów</a>
    </p>

```

```

        <asp:Button ID="buttonExportCSV" runat="server" Text="Pobierz plik CSV"
            onclick="buttonExportCSV_Click" />
    </asp:Content>

```

Default.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void buttonExportCSV_Click(object sender, EventArgs e)
    {
        DataTable dataTable = CSVExporter.getDatabaseCopy();
        CSVExporter.writeDelimitedData(dataTable, "BD_" + DateTime.Now.ToString("yyyy-MM-dd") +
            ".csv", ";");
    }
}

```

ExaminationList.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeFile="ExaminationList.aspx.cs" Inherits="ExaminationList" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <asp:Label ID="labelAppointmentId" runat="server" Text="labelPatientNumber"></asp:Label>
    <asp:Table ID="tableExaminations" runat="server">
        <asp:TableHeaderRow>
        </asp:TableHeaderRow>
    </asp:Table>
    <asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
    <br />
    <asp:Button ID="buttonNewUPDRS" runat="server" Text="Dodaj nowe badanie - UPDRS"
        onclick="buttonNewUPDRS_Click" />
    <asp:Button ID="buttonBack" runat="server" Text="Powrót" onclick="buttonBack_Click" />
</asp:Content>

```

ExaminationList.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

public partial class ExaminationList : System.Web.UI.Page
{
    private int appointmentId;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["AppointmentId"] != null)
        {
            appointmentId = (int)Session["AppointmentId"];
            labelAppointmentId.Text = "Wizyta numer: " + appointmentId;

            List<ExaminationSelection> existingExaminations = getExaminations(appointmentId);

```



```

        TableHeaderRow header = new TableHeaderRow();
        TableHeaderCell headerCell1 = new TableHeaderCell();
        TableHeaderCell headerCell2 = new TableHeaderCell();
        header.Cells.Add(headerCell1);
        header.Cells.Add(headerCell2);
        tableExaminations.Rows.Add(header);
        headerCell1.Text = "Badania";

        bool exists = false;
        foreach (ExaminationSelection existingExamination in existingExaminations)
        {
            exists = true;
            TableRow row = new TableRow();
            TableCell cell1 = new TableCell();
            TableCell cell2 = new TableCell();
            row.Cells.Add(cell1);
            row.Cells.Add(cell2);
            tableExaminations.Rows.Add(row);
            cell1.Controls.Add(existingExamination.labelExamination);
            cell2.Controls.Add(existingExamination.buttonEdit);
            cell2.Controls.Add(existingExamination.buttonDelete);
        }

        if (!exists)
        {
            TableRow row = new TableRow();
            TableCell cell1 = new TableCell();
            row.Cells.Add(cell1);
            tableExaminations.Rows.Add(row);
            Label labelNoExaminations = new Label();
            labelNoExaminations.Text = "Brak badań dla tej wizyty";
            cell1.Controls.Add(labelNoExaminations);
        }
    }

    private List<ExaminationSelection> getExaminations(int appointmentId)
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "select IdBadanie from Badanie where exists (select IdWizyta from Wizyta
where Badanie.IdWizyta = Wizyta.IdWizyta and Wizyta.IdWizyta = " + appointmentId + ")";
        cmd.Connection = con;

        List<ExaminationSelection> list = new List<ExaminationSelection>();

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();

            while (rdr.Read())
            {
                ExaminationSelection appointment = new ExaminationSelection((int)rdr["IdBadanie"],
this);
                list.Add(appointment);
            }
        }
        catch (SqlException ex)
        {
            labelMessage.Text = ex.Message;
        }
        finally
        {
            cmd.Dispose();
            if (con != null)

```

```

        {
            con.Close();
        }
    }

    return list;
}

protected void buttonNewUPDRS_Click(object sender, EventArgs e)
{
    Response.Redirect("~/UPDRSForm.aspx");
}

private class ExaminationSelection
{
    private int idBadanie;
    private ExaminationList page;
    public Label labelExamination;
    public Button buttonEdit;
    public Button buttonDelete;

    public ExaminationSelection(int id, ExaminationList page)
    {
        idBadanie = id;
        this.page = page;
        labelExamination = new Label();
        labelExamination.Text = "Badanie nr " + id;
        buttonEdit = new Button();
        buttonEdit.Text = "Edytuj";
        buttonEdit.Click += new System.EventHandler(buttonEdit_Click);
        buttonDelete = new Button();
        buttonDelete.Text = "Usuń";
        buttonDelete.Click += new System.EventHandler(buttonDelete_Click);
    }

    protected void buttonEdit_Click(object sender, EventArgs e)
    {
        //page.Session["PatientNumber"] = page.Session["PatientNumber"];
        //page.Session["AppointmentType"] = typeKey;
        //page.Session["Update"] = true;
        //page.Session["AppointmentId"] = idAppointment;
        page.Response.Redirect("~/UPDRSForm.aspx");
    }

    protected void buttonDelete_Click(object sender, EventArgs e)
    {
        SqlConnection con = new
        SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "delete from Badanie where IdBadanie = " + idBadanie;
        cmd.Connection = con;

        List<string> list = new List<string>();

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
        }
        catch (SqlException ex)
        {
            page.labelMessage.Text = ex.Message;
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {

```

```

        con.Close();
    }
    page.Response.Redirect(page.Request.RawUrl);
}
}

protected void buttonBack_Click(object sender, EventArgs e)
{
    Response.Redirect("~/AppointmentList.aspx");
}
}

Login.aspx
<%@ Page Title="Zaloguj do TPP" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true" CodeFile="Login.aspx.cs" Inherits="Login" %>

<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <asp:Label ID="lblMessage" runat="server" Text=""></asp:Label>
    <div style="text-align: center;">
        <div style="width: 400px; margin-left: auto; margin-right: auto;">
            <asp:Login ID="AppLogin" runat="server"
                DestinationPageUrl="~/Main.aspx"

                TitleText="Proszę podać nazwę użytkownika i hasło."
                onauthenticate="AppLogin_Authenticate" DisplayRememberMe="False"
                UserNameLabelText="Użytkownik:" PasswordLabelText="Hasło:"
                LoginButtonText="Zaloguj"
                FailureText="Próba zalogowania nie powiodła się. <br />Proszę spróbować
ponownie.">
            </asp:Login>
        </div>
    </div>
</asp:Content>

```

```

Login.aspx.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
using System.Web.Configuration;
using System.Web.Security;

public partial class Login : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void AppLogin_Authenticate(object sender, AuthenticateEventArgs e)
    {
        string user = AppLogin.UserName;
        string password = AppLogin.Password;

        if (user != string.Empty && password != string.Empty)
        {
            if (Validate_Login(user, password))
            {
                FormsAuthentication.RedirectFromLoginPage(user, false);
                //Response.Redirect(AppLogin.DestinationPageUrl);
            }
        }
    }
}

```

```

private bool Validate_Login(String Username, String Password)
{
    //test/pass
    SqlConnection con = new
    SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();

    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "[dbo].[validate_password]";
    cmd.Parameters.Add("@login", SqlDbType.VarChar, 30).Value = Username;
    cmd.Parameters.Add("@pass", SqlDbType.VarChar, 25).Value = Password;
    cmd.Parameters.Add("@res", SqlDbType.Bit);
    cmd.Parameters["@res"].Direction = ParameterDirection.Output;
    cmd.Connection = con;

    bool success = false;
    try
    {
        con.Open();
        cmd.ExecuteNonQuery();
        success = (bool) cmd.Parameters["@res"].Value;
    }
    catch (SqlException ex)
    {
        lblMessage.Text = ex.Message;
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }

    return success;
}
}

```

Main.aspx

```

<%@ Page Title="Strona główna" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="Main.aspx.cs" Inherits="Main" %>

```

```

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <h2>
        Strona główna
    </h2>
    <p>
        Pacjenci.
    </p>
    <table ID="Table1" runat="server">
        <tr>
            <td>BMT</td>
            <td>DBS</td>
            <td>POP</td>
        </tr>
        <tr>
            <td>
                <asp:ListBox ID="listPatientsBMT" runat="server" Rows="10" Width="200px"
                    AutoPostBack="True"
OnSelectedIndexChanged="listPatients_SelectedIndexChanged"></asp:ListBox>
            </td>
            <td>
                <asp:ListBox ID="listPatientsDBS" runat="server" Rows="10" Width="200px"
                    AutoPostBack="True"
OnSelectedIndexChanged="listPatients_SelectedIndexChanged"></asp:ListBox>
            </td>
            <td>

```

```

        <asp:ListBox ID="listPatientsPOP" runat="server" Rows="10" Width="200px"
            AutoPostBack="True"
            OnSelectedIndexChanged="listPatients_SelectedIndexChanged"></asp:ListBox>
    </td>
    <td>
        <table ID="Table2" runat="server">
            <tr>
                <td>
                    <asp:Button ID="buttonShowAppointments" runat="server" Text="Wyświetl
listę wizyt"
                        Width="200px" onclick="buttonShowAppointments_Click" />
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Button ID="buttonNewPatient" runat="server" Text="Wprowadź nowego
pacjenta"
                        onclick="buttonNewPatient_Click" Width="200px" />
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Button ID="buttonEditPatient" runat="server" Text="Edytuj dane
pacjenta"
                        Width="200px" onclick="buttonEditPatient_Click" />
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Button ID="buttonDeletePatient" runat="server" Text="Usuń dane
pacjenta"
                        Width="200px" onclick="buttonDeletePatient_Click" />
                </td>
            </tr>
        </table>
    </td>
</tr>
</table>
<asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
</asp:Content>

```

```

Main.aspx.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
using System.Web.Configuration;

public partial class Main : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            listPatientsBMT.DataSource = getPatients("BMT");
            listPatientsBMT.DataBind();
            listPatientsDBS.DataSource = getPatients("DBS");
            listPatientsDBS.DataBind();
            listPatientsPOP.DataSource = getPatients("POP");
            listPatientsPOP.DataBind();
            toggleButtons(false);
        }
    }
}

```

```

private List<string> getPatients(string group)
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select NumerPacjenta from Pacjent where NazwaGrupy = '" + group + "'";
    cmd.Connection = con;

    List<string> list = new List<string>();

    try
    {
        con.Open();
        SqlDataReader rdr = cmd.ExecuteReader();

        while (rdr.Read())
        {
            string numerPacjenta = (string) rdr["NumerPacjenta"];

            list.Add(numerPacjenta);
        }
    }
    catch (SqlException ex)
    {
        labelMessage.Text = ex.Message;
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }

    return list;
}

protected void buttonNewPatient_Click(object sender, EventArgs e)
{
    Session.Remove("PatientNumber");
    Response.Redirect("~/PatientForm.aspx");
}
// todo: remove else if's
protected void buttonDeletePatient_Click(object sender, EventArgs e)
{
    if (listPatientsBMT.SelectedIndex >= 0)
    {
        deletePatient(listPatientsBMT.SelectedValue);
        Response.Redirect(Request.RawUrl);
    }
    else if (listPatientsDBS.SelectedIndex >= 0)
    {
        deletePatient(listPatientsDBS.SelectedValue);
        Response.Redirect(Request.RawUrl);
    }
    else if (listPatientsPOP.SelectedIndex >= 0)
    {
        deletePatient(listPatientsPOP.SelectedValue);
        Response.Redirect(Request.RawUrl);
    }
}

private void deletePatient(string patientNumber)
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());

```

```

SqlCommand cmd = new SqlCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "delete from Pacjent where NumerPacjenta = '" + patientNumber + "'";
cmd.Connection = con;

try
{
    con.Open();
    SqlDataReader rdr = cmd.ExecuteReader();
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}
}

protected void buttonEditPatient_Click(object sender, EventArgs e)
{
    if (listPatientsBMT.SelectedIndex >= 0)
    {
        Session["PatientNumber"] = listPatientsBMT.SelectedValue;
        Response.Redirect("~/PatientForm.aspx");
    }
    else if (listPatientsDBS.SelectedIndex >= 0)
    {
        Session["PatientNumber"] = listPatientsDBS.SelectedValue;
        Response.Redirect("~/PatientForm.aspx");
    }
    else if (listPatientsPOP.SelectedIndex >= 0)
    {
        Session["PatientNumber"] = listPatientsPOP.SelectedValue;
        Response.Redirect("~/PatientForm.aspx");
    }
}

protected void buttonShowAppointments_Click(object sender, EventArgs e)
{
    if (listPatientsBMT.SelectedIndex >= 0)
    {
        Session["PatientNumber"] = listPatientsBMT.SelectedValue;
        Response.Redirect("~/AppointmentList.aspx");
    }
    else if (listPatientsDBS.SelectedIndex >= 0)
    {
        Session["PatientNumber"] = listPatientsDBS.SelectedValue;
        Response.Redirect("~/AppointmentList.aspx");
    }
    else if (listPatientsPOP.SelectedIndex >= 0)
    {
        Session["PatientNumber"] = listPatientsPOP.SelectedValue;
        Response.Redirect("~/AppointmentList.aspx");
    }
}

protected void listPatients_SelectedIndexChanged(object sender, System.EventArgs e)
{
    if (sender.Equals(listPatientsBMT))
    {
        listPatientsDBS.ClearSelection();
        listPatientsPOP.ClearSelection();
    }
}

```

```

        else if (sender.Equals(listPatientsDBS))
        {
            listPatientsBMT.ClearSelection();
            listPatientsPOP.ClearSelection();
        }
        else if (sender.Equals(listPatientsPOP))
        {
            listPatientsBMT.ClearSelection();
            listPatientsDBS.ClearSelection();
        }
        if (listPatientsBMT.SelectedIndex >= 0 || listPatientsDBS.SelectedIndex >= 0 ||
listPatientsPOP.SelectedIndex >= 0)
        {
            toggleButtons(true);
        }
    }

    private void toggleButtons(bool enable)
    {
        buttonEditPatient.Enabled = enable;
        buttonDeletePatient.Enabled = enable;
        buttonShowAppointments.Enabled = enable;
    }
}

```

PartBForm.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="PartBForm.aspx.cs" Inherits="PartBForm" %>

```

```

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <h2>Objawy i epidemiologia</h2>
    <br />
    <asp:Label ID="labelAppointment" runat="server" Text=""></asp:Label>
    <table ID="Table1" runat="server">
        <tr>
            <td>Przebyte leczenie operacyjne:</td>
            <td>
                <asp:DropDownList ID="dropPrzebyteLeczenieOperacyjne" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Papierosy:</td>
            <td>
                <asp:DropDownList ID="dropCigarettes" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Kawa:</td>
            <td>
                <asp:DropDownList ID="dropCoffee" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Zielona herbata:</td>
            <td>
                <asp:DropDownList ID="dropGreenTea" runat="server">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="1">tak</asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Alkohol:</td>

```



```

        <td>
            <asp:DropDownList ID="dropAlcohol" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Liczba zabiegów w znieczuleniu<br />ogólnym przed rozpoznaniem PD:</td>
        <td>
            <asp:TextBox ID="textTreatmentNumber" runat="server"></asp:TextBox>
            <!--
            <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server"
                ErrorMessage="*"
                ControlToValidate="textTreatmentNumber"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator Runat="server" ID="RegularExpressionValidator4"
                ControlToValidate="textTreatmentNumber"
                ErrorMessage="Proszę podać liczbę od 0 do 255."
                ValidationExpression="^([0-9]|[1-9][0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-
5])$"></asp:RegularExpressionValidator>
            --%>
        </td>
    </tr>
    <tr>
        <td>Zamieszkanie:</td>
        <td>
            <asp:DropDownList ID="dropZamieszkanie" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Narażenie na toksyczność:</td>
        <td>
            <asp:CheckBoxList ID="checkListToxic" runat="server">
            </asp:CheckBoxList>
        </td>
    </tr>
    <tr>
        <td>Uwagi:</td>
        <td>
            <asp:TextBox ID="textNotes" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>Nadciśnienie:</td>
        <td>
            <asp:DropDownList ID="dropNadcisnienie" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Blokery kanału wapniowego:</td>
        <td>
            <asp:DropDownList ID="dropBlokery" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Dominujący objaw obecnie:</td>
        <td>
            <asp:DropDownList ID="dropDominujacyObjawObecnie" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>

```

```

        <td>Dominujący objaw uwagi:</td>
        <td>
            <asp:TextBox ID="textDominujacyObjawUwagi" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>Objawy autonomiczne:</td>
        <td>
            <asp:CheckBoxList ID="checkListObjawyAutonomiczne" runat="server">
            </asp:CheckBoxList>
        </td>
    </tr>
    <tr>
        <td>RLS:</td>
        <td>
            <asp:DropDownList ID="dropRLS" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Objawy psychotyczne:</td>
        <td>
            <asp:DropDownList ID="dropPsychotyczne" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Depresja:</td>
        <td>
            <asp:DropDownList ID="dropDepresja" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Otępienie:</td>
        <td>
            <asp:DropDownList ID="dropOtepienie" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Dyzartia:</td>
        <td>
            <asp:DropDownList ID="dropDyzartia" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Dysfagia objaw:</td>
        <td>
            <asp:DropDownList ID="dropDysfagiaObjaw" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>

```

```

<tr>
    <td>RBD:</td>
    <td>
        <asp:DropDownList ID="dropRBD" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Zaburzenia ruchomości gałek ocznych:</td>
    <td>
        <asp:DropDownList ID="dropZaburzeniaGalek" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Apraksja:</td>
    <td>
        <asp:DropDownList ID="dropApraksja" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Test klaskania:</td>
    <td>
        <asp:DropDownList ID="dropKlaskanie" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="1">prawidłowy</asp:ListItem>
            <asp:ListItem Value="0">nieprawidłowy</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Zaburzenia węchowe:</td>
    <td>
        <asp:DropDownList ID="dropZaburzeniaWechowe" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
</tr>
</table>
<asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
<br />
<asp:Button ID="buttonOK" runat="server" Text="Zatwierdź"
    onclick="buttonOK_Click" />
<asp:Button ID="buttonCancel" runat="server" Text="Anuluj"
    CausesValidation="False" onclick="buttonCancel_Click" />
</asp:Content>

```

```

PartBForm.aspx.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

```

```

public partial class PartBForm : System.Web.UI.Page
{
    private static byte NO_DATA = 100;
    private static decimal NO_DATA_DECIMAL = 100;
    private List<Tuple<CheckBox, byte>> toxicChoiceList = new List<Tuple<CheckBox, byte>>();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["PatientNumber"] != null && Session["AppointmentId"] != null &&
Session["AppointmentName"] != null)
        {
            labelAppointment.Text = "Pacjent: " + Session["PatientNumber"] + "<br />Wizyta: " +
Session["AppointmentName"];
            if (!IsPostBack)
            {
                initControls();
                loadPartB();
            }
        }
        else
        {
            Response.Redirect("~/Main.aspx");
        }
    }

    private void initControls()
    {
        dropPrzebyteLeczenieOperacyjne.DataSource =
DatabaseProcedures.getEnumerationByteWithNoData("Wizyta", "PrzebyteLeczenieOperacyjnePD",
NO_DATA);
        dropPrzebyteLeczenieOperacyjne.DataTextField = "Value";
        dropPrzebyteLeczenieOperacyjne.DataValueField = "Key";
        dropPrzebyteLeczenieOperacyjne.DataBind();

        dropCigarettes.DataSource = DatabaseProcedures.getEnumerationByteWithNoData("Wizyta",
"Papierosy", NO_DATA);
        dropCigarettes.DataTextField = "Value";
        dropCigarettes.DataValueField = "Key";
        dropCigarettes.DataBind();

        dropCoffee.DataSource = DatabaseProcedures.getEnumerationByteWithNoData("Wizyta", "Kawa",
NO_DATA);
        dropCoffee.DataTextField = "Value";
        dropCoffee.DataValueField = "Key";
        dropCoffee.DataBind();

        dropAlcohol.DataSource = DatabaseProcedures.getEnumerationByteWithNoData("Wizyta",
"Alkohol", NO_DATA);
        dropAlcohol.DataTextField = "Value";
        dropAlcohol.DataValueField = "Key";
        dropAlcohol.DataBind();

        dropZamieszkanie.DataSource = DatabaseProcedures.getEnumerationByteWithNoData("Wizyta",
"Zamieszkanie", NO_DATA);
        dropZamieszkanie.DataTextField = "Value";
        dropZamieszkanie.DataValueField = "Key";
        dropZamieszkanie.DataBind();

        checkListToxic.DataSource = DatabaseProcedures.getEnumerationByte("Wizyta",
"NarazenieNaToks");
        checkListToxic.DataTextField = "Value";
        checkListToxic.DataValueField = "Key";
        checkListToxic.DataBind();

        dropDominujacyObjawObecnie.DataSource =
DatabaseProcedures.getEnumerationByteWithNoData("Wizyta", "DominujacyObjawObecnie", NO_DATA);
        dropDominujacyObjawObecnie.DataTextField = "Value";
        dropDominujacyObjawObecnie.DataValueField = "Key";
    }
}

```

```

        dropDominujacyObjawObecnie.DataBind();

        checkListObjawyAutonomiczne.DataSource = DatabaseProcedures.getEnumerationByte("Wizyta",
"ObjawyAutonomiczne");
        checkListObjawyAutonomiczne.DataTextField = "Value";
        checkListObjawyAutonomiczne.DataValueField = "Key";
        checkListObjawyAutonomiczne.DataBind();

        dropOtepienie.DataSource = DatabaseProcedures.getEnumerationDecimalWithNoData("Wizyta",
"Otepienie", NO_DATA_DECIMAL);
        dropOtepienie.DataTextField = "Value";
        dropOtepienie.DataValueField = "Key";
        dropOtepienie.DataBind();

        Utils.setSelectedCheckListItems(DatabaseProcedures.getMultiChoice("NarazenieNaToks",
int.Parse(Session["AppointmentId"].ToString()), checkListToxic);
        Utils.setSelectedCheckListItems(DatabaseProcedures.getMultiChoice("ObjawyAutonomiczne",
int.Parse(Session["AppointmentId"].ToString()), checkListObjawyAutonomiczne);
    }

    private void savePartB()
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[update_examination_questionnaire_partB]";
        cmd.Parameters.Add("@IdWizyta", SqlDbType.Int).Value =
int.Parse(Session["AppointmentId"].ToString());
        cmd.Parameters.Add("@PrzebyteLeczenieOperacyjnePD", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropPrzebyteLeczenieOperacyjne.SelectedValue,
NO_DATA.ToString());
        cmd.Parameters.Add("@Papierosy", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropCigarettes.SelectedValue, NO_DATA.ToString());
        cmd.Parameters.Add("@Kawa", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropCoffee.SelectedValue, NO_DATA.ToString());
        cmd.Parameters.Add("@ZielonaHerbata", SqlDbType.TinyInt).Value =
byte.Parse(dropGreenTea.SelectedValue);
        cmd.Parameters.Add("@Alkohol", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropAlcohol.SelectedValue, NO_DATA.ToString());
        cmd.Parameters.Add("@ZabiegowWZnieczOgPrzedRozpoznaniemPD", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNull(textTreatmentNumber.Text);
        cmd.Parameters.Add("@Zamieszkanie", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropZamieszkanie.SelectedValue, NO_DATA.ToString());
        cmd.Parameters.Add("@Uwagi", SqlDbType.VarChar, 50).Value = textNotes.Text;
        cmd.Parameters.Add("@Nadcisnienie", SqlDbType.TinyInt).Value =
byte.Parse(dropNadcisnienie.SelectedValue);
        cmd.Parameters.Add("@BlokeryKanWapn", SqlDbType.TinyInt).Value =
byte.Parse(dropBlokery.SelectedValue);
        cmd.Parameters.Add("@DominujacyObjawObecnie", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropDominujacyObjawObecnie.SelectedValue,
NO_DATA.ToString());
        cmd.Parameters.Add("@DominujacyObjawUwagi", SqlDbType.VarChar, 50).Value =
textDominujacyObjawUwagi.Text.ToString();
        cmd.Parameters.Add("@RLS", SqlDbType.TinyInt).Value = byte.Parse(dropRLS.SelectedValue);
        cmd.Parameters.Add("@ObjawyPsychotyczne", SqlDbType.TinyInt).Value =
byte.Parse(dropPsychotyczne.SelectedValue);
        cmd.Parameters.Add("@Depresja", SqlDbType.TinyInt).Value =
byte.Parse(dropDepresja.SelectedValue);
        SqlParameter otepienieDecimal = new SqlParameter("@Otepienie", SqlDbType.Decimal);
        otepienieDecimal.Precision = 2;
        otepienieDecimal.Scale = 1;
        otepienieDecimal.Value =
DatabaseProcedures.getDecimalOrNullWithNoData(dropOtepienie.SelectedValue,
NO_DATA_DECIMAL.ToString());
        cmd.Parameters.Add(otepienieDecimal);
        cmd.Parameters.Add("@Dyzartria", SqlDbType.TinyInt).Value =
byte.Parse(dropDyzartia.SelectedValue);

```

```

        cmd.Parameters.Add("@DysfagiaObjaw", SqlDbType.TinyInt).Value =
byte.Parse(dropDysfagiaObjaw.SelectedValue);
        cmd.Parameters.Add("@RBD", SqlDbType.TinyInt).Value = byte.Parse(dropRBD.SelectedValue);
        cmd.Parameters.Add("@ZaburzenieRuchomosciGalekOcznych", SqlDbType.TinyInt).Value =
byte.Parse(dropZaburzeniaGalek.SelectedValue);
        cmd.Parameters.Add("@Apraksja", SqlDbType.TinyInt).Value =
byte.Parse(dropApraksja.SelectedValue);
        cmd.Parameters.Add("@TestKlaskania", SqlDbType.TinyInt).Value =
byte.Parse(dropKlaskanie.SelectedValue);
        cmd.Parameters.Add("@ZaburzeniaWechowe", SqlDbType.TinyInt).Value =
byte.Parse(dropZaburzeniaWechowe.SelectedValue);
        cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
        cmd.Parameters.Add("@result", SqlDbType.Int);
        cmd.Parameters["@result"].Direction = ParameterDirection.Output;
        cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
        cmd.Parameters["@message"].Direction = ParameterDirection.Output;
        cmd.Connection = con;

        int success = 0;
        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
            success = (int)cmd.Parameters["@result"].Value;

            if (success == 0)
            {
                Response.Redirect("~/AppointmentForm.aspx");
            }
            else
            {
                labelMessage.Text = (string)cmd.Parameters["@message"].Value;
            }
        }
        catch (SqlException ex)
        {
            labelMessage.Text = ex.Message;
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }
    }

    private void loadPartB()
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "select PrzebyteLeczenieOperacyjnePD, Papierosy, Kawa, ZielonaHerbata,
Alkohol, ZabiegowWZnieczOgPrzedRozpoznaniemPD, " +
            "Zamieszkanie, Uwagi, Nadcisnienie, BlokeryKanWapn, DominujacyObjawObecnie,
DominujacyObjawUwagi, RLS, ObjawyPsychotyczne, Depresja, Otepienie, Dyzartria, DysfagiaObjaw, RBD,
ZaburzenieRuchomosciGalekOcznych, " +
            "Apraksja, TestKlaskania, ZaburzeniaWechowe, MasaCiala, Drzenie, Sztynnosc,
Spowolnienie, " +
            "ObjawyInne, ObjawyInneJakie, CzasOFF, PoprawaPoLDopie from Wizyta where IdWizyta = "
+ Session["AppointmentId"];
        cmd.Connection = con;

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();

```

```

        while (rdr.Read())
        {
            dropPrzebyteLeczenieOperacyjne.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["PrzebyteLeczenieOperacyjnePD"],
NO_DATA.ToString());
            dropCigarettes.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Papierosy"], NO_DATA.ToString());
            dropCoffee.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Kawa"], NO_DATA.ToString());
            dropGreenTea.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["ZielonaHerbata"], NO_DATA.ToString());
            dropAlcohol.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Alkohol"], NO_DATA.ToString());
            textTreatmentNumber.Text =
DatabaseProcedures.getTextByteValue(rdr["ZabiegowWZnieczOgPrzedRozpoznaniemPD"]);
            dropZamieszkanie.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Zamieszkanie"], NO_DATA.ToString());
            textNotes.Text = DatabaseProcedures.getTextStringValue(rdr["Uwagi"]);
            dropNadcisnienie.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["Nadcisnienie"]);
            dropBlokery.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["BlokeryKanWapn"]);
            dropDominujacyObjawObecnie.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["DominujacyObjawObecnie"], NO_DATA.ToString());
            textDominujacyObjawUwagi.Text =
DatabaseProcedures.getTextStringValue(rdr["DominujacyObjawUwagi"]);
            dropRLS.SelectedValue = DatabaseProcedures.getDropYesNoValue(rdr["RLS"]);
            dropPsychotyczne.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["ObjawyPsychotyczne"]);
            dropDepresja.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["Depresja"]);
            dropOtepienie.SelectedValue =
DatabaseProcedures.getDropDecimalValueWithNoData(rdr["Otepienie"], NO_DATA_DECIMAL.ToString());
            dropDyzartia.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["Dyzartia"]);
            dropDysfagiaObjaw.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["DysfagiaObjaw"], NO_DATA.ToString());
            dropRBD.SelectedValue = DatabaseProcedures.getDropYesNoValue(rdr["RBD"]);
            dropZaburzeniaGalek.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["ZaburzenieRuchomosciGalekOcznych"]);
            dropApraksja.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["Apraksja"]);
            dropKlaskanie.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["TestKlaskania"]);
            dropZaburzeniaWechowe.SelectedValue =
DatabaseProcedures.getDropYesNoValue(rdr["ZaburzeniaWechowe"]);
        }
    }
    catch (SqlException ex)
    {
        labelMessage.Text = ex.Message;
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }
}

protected void buttonOK_Click(object sender, EventArgs e)
{
    DatabaseProcedures.saveMultiChoice(Utils.getSelectedCheckListItems(checkListToxic),
"NarazenieNaToks", int.Parse(Session["AppointmentId"].ToString()), User.Identity.Name);

    DatabaseProcedures.saveMultiChoice(Utils.getSelectedCheckListItems(checkListObjawyAutonomiczne),
"ObjawyAutonomiczne", int.Parse(Session["AppointmentId"].ToString()), User.Identity.Name);
}

```

```

        savePartB();
    }
    protected void buttonCancel_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/AppointmentForm.aspx");
    }
}

```

PartCForm.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="PartCForm.aspx.cs" Inherits="PartCForm" %>

```

```

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <h2>Przyjmowane leki</h2>
    <br />
    <asp:Label ID="labelAppointment" runat="server" Text=""></asp:Label>
    <table ID="Table1" runat="server">
        <tr>
            <th>Lek</th>
            <th>Przyjmowanie</th>
            <th>Dawka</th>
        </tr>
        <tr>
            <td>LDopa:</td>
            <td>
                <asp:DropDownList ID="dropLDopa" runat="server">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>
                    <asp:ListItem Value="1">tak</asp:ListItem>
                </asp:DropDownList>
            </td>
            <td>
                <asp:TextBox ID="textLDopa" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>Agonista:</td>
            <td>
                <asp:DropDownList ID="dropAgonista" runat="server">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>
                    <asp:ListItem Value="1">tak</asp:ListItem>
                </asp:DropDownList>
            </td>
            <td>
                <asp:TextBox ID="textAgonista" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>Amantadyna:</td>
            <td>
                <asp:DropDownList ID="dropAmantadyna" runat="server">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>
                    <asp:ListItem Value="1">tak</asp:ListItem>
                </asp:DropDownList>
            </td>
            <td>
                <asp:TextBox ID="textAmantadyna" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>MAOBinh:</td>
            <td>
                <asp:DropDownList ID="dropMAOBinh" runat="server">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>

```



```

        <asp:ListItem Value="1">tak</asp:ListItem>
    </asp:DropDownList>
</td>
<td>
    <asp:TextBox ID="textMAOBinh" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>COMTinh</td>
<td>
    <asp:DropDownList ID="dropCOMTinh" runat="server">
        <asp:ListItem Value="2" Text=""></asp:ListItem>
        <asp:ListItem Value="0">nie</asp:ListItem>
        <asp:ListItem Value="1">tak</asp:ListItem>
    </asp:DropDownList>
</td>
<td>
    <asp:TextBox ID="textCOMTinh" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>Cholinolityk:</td>
<td>
    <asp:DropDownList ID="dropCholinolityk" runat="server">
        <asp:ListItem Value="2" Text=""></asp:ListItem>
        <asp:ListItem Value="0">nie</asp:ListItem>
        <asp:ListItem Value="1">tak</asp:ListItem>
    </asp:DropDownList>
</td>
<td>
    <asp:TextBox ID="textCholinolityk" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>Leki inne:</td>
<td>
    <asp:DropDownList ID="droplekiInne" runat="server">
        <asp:ListItem Value="2" Text=""></asp:ListItem>
        <asp:ListItem Value="0">nie</asp:ListItem>
        <asp:ListItem Value="1">tak</asp:ListItem>
    </asp:DropDownList>
</td>
<td>
    <asp:TextBox ID="textLekiInne" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>L_STIMOpis:</td>
<td></td>
<td>
    <asp:TextBox ID="textL_STIMOpis" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>L_STIMAmplitude:</td>
<td></td>
<td>
    <asp:TextBox ID="textL_STIMAmplitude" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>L_STIMDuration:</td>
<td></td>
<td>
    <asp:TextBox ID="textL_STIMDuration" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>L_STIMFrequency:</td>

```

```

        <td></td>
        <td>
            <asp:TextBox ID="textL_STIMFrequency" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>R_STIMOpis:</td>
        <td></td>
        <td>
            <asp:TextBox ID="textR_STIMOpis" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>R_STIMAmplitude:</td>
        <td></td>
        <td>
            <asp:TextBox ID="textR_STIMAmplitude" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>R_STIMDuration:</td>
        <td></td>
        <td>
            <asp:TextBox ID="textR_STIMDuration" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>R_STIMFrequency:</td>
        <td></td>
        <td>
            <asp:TextBox ID="textR_STIMFrequency" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>LDopa (wypis):</td>
        <td>
            <asp:DropDownList ID="dropWypis_LDopa" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
        <td>
            <asp:TextBox ID="textWypis_LDopa" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>Agonista (wypis):</td>
        <td>
            <asp:DropDownList ID="dropWypis_Agonista" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
        <td>
            <asp:TextBox ID="textWypis_Agonista" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>Amantadyna (wypis):</td>
        <td>
            <asp:DropDownList ID="dropWypis_Amantadyna" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
        <td>

```

```

        <asp:TextBox ID="textWypis_Amantadyna" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>MAOBinh (wypis):</td>
    <td>
        <asp:DropDownList ID="dropWypis_MAOBinh" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
    <td>
        <asp:TextBox ID="textWypis_MAOBinh" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>COMTinh (wypis):</td>
    <td>
        <asp:DropDownList ID="dropWypis_COMTinh" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
    <td>
        <asp:TextBox ID="textWypis_COMTinh" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Cholinolityk (wypis):</td>
    <td>
        <asp:DropDownList ID="dropWypis_Cholinolityk" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
    <td>
        <asp:TextBox ID="textWypis_Cholinolityk" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>LekiInne (wypis):</td>
    <td>
        <asp:DropDownList ID="dropWypis_LekiInne" runat="server">
            <asp:ListItem Value="2" Text=""></asp:ListItem>
            <asp:ListItem Value="0">nie</asp:ListItem>
            <asp:ListItem Value="1">tak</asp:ListItem>
        </asp:DropDownList>
    </td>
    <td>
        <asp:TextBox ID="textWypis_LekiInne" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>L_STIMOpis (wypis):</td>
    <td></td>
    <td>
        <asp:TextBox ID="textWypis_L_STIMOpis" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>L_STIMAmplitude (wypis):</td>
    <td></td>
    <td>
        <asp:TextBox ID="textWypis_L_STIMAmplitude" runat="server"></asp:TextBox>
    </td>
</tr>

```

```

<tr>
    <td>L_STIMDuration (wypis):</td>
    <td></td>
    <td>
        <asp:TextBox ID="textWypis_L_STIMDuration" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>L_STIMFrequency (wypis):</td>
    <td></td>
    <td>
        <asp:TextBox ID="textWypis_L_STIMFrequency" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>R_STIMOpis (wypis):</td>
    <td></td>
    <td>
        <asp:TextBox ID="textWypis_R_STIMOpis" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>R_STIMAmplitude (wypis):</td>
    <td></td>
    <td>
        <asp:TextBox ID="textWypis_R_STIMAmplitude" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>R_STIMDuration (wypis):</td>
    <td></td>
    <td>
        <asp:TextBox ID="textWypis_R_STIMDuration" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>R_STIMFrequency (wypis):</td>
    <td></td>
    <td>
        <asp:TextBox ID="textWypis_R_STIMFrequency" runat="server"></asp:TextBox>
    </td>
</tr>
</table>
<asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
<br />
<asp:Button ID="buttonOK" runat="server" Text="Zatwierdź"
    onclick="buttonOK_Click" />
<asp:Button ID="buttonCancel" runat="server" Text="Anuluj"
    CausesValidation="False" onclick="buttonCancel_Click" />
</asp:Content>

```

PartCForm.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

public partial class PartCForm : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["PatientNumber"] != null && Session["AppointmentId"] != null &&
            Session["AppointmentName"] != null)
        {

```

```

        labelAppointment.Text = "Pacjent: " + Session["PatientNumber"] + "<br />Wizyta: " +
Session["AppointmentName"];
        if (!IsPostBack)
        {
            loadPartC();
        }
    }
    else
    {
        Response.Redirect("~/Main.aspx");
    }
}

private void savePartC()
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "[dbo].[update_examination_questionnaire_partC]";
    cmd.Parameters.Add("@IdWizyta", SqlDbType.Int).Value =
int.Parse(Session["AppointmentId"].ToString());
    cmd.Parameters.Add("@Ldopa", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(dropLdopa.SelectedValues);
    cmd.Parameters.Add("@LDopaObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textLDopa.Text);
    cmd.Parameters.Add("@Agonista", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropAgonista.SelectedValues);
    cmd.Parameters.Add("@AgonistaObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textAgonista.Text);
    cmd.Parameters.Add("@Amantadyna", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropAmantadyna.SelectedValues);
    cmd.Parameters.Add("@AmantadynaObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textAmantadyna.Text);
    cmd.Parameters.Add("@MAOBinh", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropMAOBinh.SelectedValues);
    cmd.Parameters.Add("@MAOBinhObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textMAOBinh.Text);
    cmd.Parameters.Add("@COMTinh", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropCOMTinh.SelectedValues);
    cmd.Parameters.Add("@COMTinhObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textCOMTinh.Text);
    cmd.Parameters.Add("@Cholinolityk", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropCholinolityk.SelectedValues);
    cmd.Parameters.Add("@CholinolitykObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textCholinolityk.Text);
    cmd.Parameters.Add("@LekiInne", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropLekiInne.SelectedValues);
    cmd.Parameters.Add("@LekiInneJakie", SqlDbType.VarChar, 50).Value =
DatabaseProcedures.getStringOrNull(textLekiInne.Text);
    cmd.Parameters.Add("@L_STIMOpis", SqlDbType.VarChar, 30).Value =
DatabaseProcedures.getStringOrNull(textL_STIMOpis.Text);
    cmd.Parameters.Add(saveDecimal("@L_STIMAmplitude", textL_STIMAmplitude));
    cmd.Parameters.Add(saveDecimal("@L_STIMDuration", textL_STIMDuration));
    cmd.Parameters.Add(saveDecimal("@L_STIMFrequency", textL_STIMFrequency));
    cmd.Parameters.Add("@R_STIMOpis", SqlDbType.VarChar, 30).Value =
DatabaseProcedures.getStringOrNull(textR_STIMOpis.Text);
    cmd.Parameters.Add(saveDecimal("@R_STIMAmplitude", textR_STIMAmplitude));
    cmd.Parameters.Add(saveDecimal("@R_STIMDuration", textR_STIMDuration));
    cmd.Parameters.Add(saveDecimal("@R_STIMFrequency", textR_STIMFrequency));
    cmd.Parameters.Add("@Wypis_Ldopa", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(dropWypis_Ldopa.SelectedValues);
    cmd.Parameters.Add("@Wypis_LDopaObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textWypis_LDopa.Text);
    cmd.Parameters.Add("@Wypis_Agonista", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropWypis_Agonista.SelectedValues);
    cmd.Parameters.Add("@Wypis_AgonistaObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textWypis_Agonista.Text);
}

```

```

        cmd.Parameters.Add("@Wypis_Amantadyna", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropWypis_Amantadyna.SelectedValue);
        cmd.Parameters.Add("@Wypis_AmantadynaObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textWypis_Amantadyna.Text);
        cmd.Parameters.Add("@Wypis_MAOBinh", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropWypis_MAOBinh.SelectedValue);
        cmd.Parameters.Add("@Wypis_MAOBinhObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textWypis_MAOBinh.Text);
        cmd.Parameters.Add("@Wypis_COMTinh", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropWypis_COMTinh.SelectedValue);
        cmd.Parameters.Add("@Wypis_COMTinhObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textWypis_COMTinh.Text);
        cmd.Parameters.Add("@Wypis_Cholinolityk", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropWypis_Cholinolityk.SelectedValue);
        cmd.Parameters.Add("@Wypis_CholinolitykObecnie", SqlDbType.SmallInt).Value =
DatabaseProcedures.getShortOrNull(textWypis_Cholinolityk.Text);
        cmd.Parameters.Add("@Wypis_LekiInne", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropWypis_LekiInne.SelectedValue);
        cmd.Parameters.Add("@Wypis_LekiInneJakie", SqlDbType.VarChar, 50).Value =
DatabaseProcedures.getStringOrNull(textWypis_LekiInne.Text);
        cmd.Parameters.Add("@Wypis_L_STIMOpis", SqlDbType.VarChar, 30).Value =
DatabaseProcedures.getStringOrNull(textWypis_L_STIMOpis.Text);
        cmd.Parameters.Add(saveDecimal("@Wypis_L_STIMAmplitude", textWypis_L_STIMAmplitude));
        cmd.Parameters.Add(saveDecimal("@Wypis_L_STIMDuration", textWypis_L_STIMDuration));
        cmd.Parameters.Add(saveDecimal("@Wypis_L_STIMFrequency", textWypis_L_STIMFrequency));
        cmd.Parameters.Add("@Wypis_R_STIMOpis", SqlDbType.VarChar, 30).Value =
DatabaseProcedures.getStringOrNull(textWypis_R_STIMOpis.Text);
        cmd.Parameters.Add(saveDecimal("@Wypis_R_STIMAmplitude", textWypis_R_STIMAmplitude));
        cmd.Parameters.Add(saveDecimal("@Wypis_R_STIMDuration", textWypis_R_STIMDuration));
        cmd.Parameters.Add(saveDecimal("@Wypis_R_STIMFrequency", textWypis_R_STIMFrequency));
        cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
        cmd.Parameters.Add("@result", SqlDbType.Int);
        cmd.Parameters["@result"].Direction = ParameterDirection.Output;
        cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
        cmd.Parameters["@message"].Direction = ParameterDirection.Output;
        cmd.Connection = con;

        int success = 0;
        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
            success = (int)cmd.Parameters["@result"].Value;

            if (success == 0)
            {
                Response.Redirect("~/AppointmentForm.aspx");
            }
            else
            {
                labelMessage.Text = (string)cmd.Parameters["@message"].Value;
            }
        }
        catch (SqlException ex)
        {
            labelMessage.Text = ex.Message;
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }
    }

    private SqlParameter saveDecimal(string parameter, TextBox textBox)
    {

```

```

        SqlParameter decimalParameter = new SqlParameter(parameter, SqlDbType.Decimal);
        decimalParameter.Precision = 5;
        decimalParameter.Scale = 1;
        decimalParameter.Value = DatabaseProcedures.getDecimalOrNull(textBox.Text);

        return decimalParameter;
    }

    private void loadPartC()
    {
        SqlConnection con = new
        SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "select Ldopa, LDopaObecnie, Agonista, AgonistaObecnie, Amantadyna,
        AmantadynaObecnie, MAOBinh, " +
        "MAOBinhObecnie, COMTinh, COMTinhObecnie, Cholinolityk, CholinolitykObecnie, LekiiInne,
        LekiiInneJakie, " +
        "L_STIMOpis, L_STIMAmplitude, L_STIMDuration, L_STIMFrequency, R_STIMOpis,
        R_STIMAmplitude, R_STIMDuration, R_STIMFrequency, " +
        "Wypis_Ldopa, Wypis_LDopaObecnie, Wypis_Agonista, Wypis_AgonistaObecnie,
        Wypis_Amantadyna, Wypis_AmantadynaObecnie, Wypis_MAOBinh, " +
        "Wypis_MAOBinhObecnie, Wypis_COMTinh, Wypis_COMTinhObecnie, Wypis_Cholinolityk,
        Wypis_CholinolitykObecnie, Wypis_LekiiInne, Wypis_LekiiInneJakie, " +
        "Wypis_L_STIMOpis, Wypis_L_STIMAmplitude, Wypis_L_STIMDuration, Wypis_L_STIMFrequency,
        Wypis_R_STIMOpis, Wypis_R_STIMAmplitude, Wypis_R_STIMDuration, Wypis_R_STIMFrequency " +
        "from Wizyta where IdWizyta = " + Session["AppointmentId"]";
        cmd.Connection = con;

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
            while (rdr.Read())
            {
                dropLdopa.SelectedValue = DatabaseProcedures.getDropYesNoValue(rdr["Ldopa"]);
                textLdopa.Text = DatabaseProcedures.getTextShortValue(rdr["LDopaObecnie"]);
                dropAgonista.SelectedValue = DatabaseProcedures.getDropBitValue(rdr["Agonista"]);
                textAgonista.Text = DatabaseProcedures.getTextShortValue(rdr["AgonistaObecnie"]);
                dropAmantadyna.SelectedValue =
                DatabaseProcedures.getDropBitValue(rdr["Amantadyna"]);
                textAmantadyna.Text =
                DatabaseProcedures.getTextShortValue(rdr["AmantadynaObecnie"]);
                dropMAOBinh.SelectedValue = DatabaseProcedures.getDropBitValue(rdr["MAOBinh"]);
                textMAOBinh.Text = DatabaseProcedures.getTextShortValue(rdr["MAOBinhObecnie"]);
                dropCOMTinh.SelectedValue = DatabaseProcedures.getDropBitValue(rdr["COMTinh"]);
                textCOMTinh.Text = DatabaseProcedures.getTextShortValue(rdr["COMTinhObecnie"]);
                dropCholinolityk.SelectedValue =
                DatabaseProcedures.getDropBitValue(rdr["Cholinolityk"]);
                textCholinolityk.Text =
                DatabaseProcedures.getTextShortValue(rdr["CholinolitykObecnie"]);
                dropLekiiInne.SelectedValue = DatabaseProcedures.getDropBitValue(rdr["LekiiInne"]);
                textLekiiInne.Text = DatabaseProcedures.getTextStringValue(rdr["LekiiInneJakie"]);
                textL_STIMOpis.Text = DatabaseProcedures.getTextStringValue(rdr["L_STIMOpis"]);
                textL_STIMAmplitude.Text =
                DatabaseProcedures.getTextDecimalValue(rdr["L_STIMAmplitude"]);
                textL_STIMDuration.Text =
                DatabaseProcedures.getTextDecimalValue(rdr["L_STIMDuration"]);
                textL_STIMFrequency.Text =
                DatabaseProcedures.getTextDecimalValue(rdr["L_STIMFrequency"]);
                textR_STIMOpis.Text = DatabaseProcedures.getTextStringValue(rdr["R_STIMOpis"]);
                textR_STIMAmplitude.Text =
                DatabaseProcedures.getTextDecimalValue(rdr["R_STIMAmplitude"]);
                textR_STIMDuration.Text =
                DatabaseProcedures.getTextDecimalValue(rdr["R_STIMDuration"]);
                textR_STIMFrequency.Text =
                DatabaseProcedures.getTextDecimalValue(rdr["R_STIMFrequency"]);
                dropWypis_Ldopa.SelectedValue =
                DatabaseProcedures.getDropYesNoValue(rdr["Wypis_Ldopa"]);
            }
        }
    }

```

```

        textWypis_LDopa.Text =
DatabaseProcedures.getTextShortValue(rdr["Wypis_LDopaObecnie"]);
        dropWypis_Agonista.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["Wypis_Agonista"]);
        textWypis_Agonista.Text =
DatabaseProcedures.getTextShortValue(rdr["Wypis_AgonistaObecnie"]);
        dropWypis_Amantadyna.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["Wypis_Amantadyna"]);
        textWypis_Amantadyna.Text =
DatabaseProcedures.getTextShortValue(rdr["Wypis_AmantadynaObecnie"]);
        dropWypis_MAOBinh.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["Wypis_MAOBinh"]);
        textWypis_MAOBinh.Text =
DatabaseProcedures.getTextShortValue(rdr["Wypis_MAOBinhObecnie"]);
        dropWypis_COMTinh.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["Wypis_COMTinh"]);
        textWypis_COMTinh.Text =
DatabaseProcedures.getTextShortValue(rdr["Wypis_COMTinhObecnie"]);
        dropWypis_Cholinolityk.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["Wypis_Cholinolityk"]);
        textWypis_Cholinolityk.Text =
DatabaseProcedures.getTextShortValue(rdr["Wypis_CholinolitykObecnie"]);
        dropWypis_LekiInne.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["Wypis_LekiInne"]);
        textWypis_LekiInne.Text =
DatabaseProcedures.getTextStringValue(rdr["Wypis_LekiInneJakie"]);
        textWypis_L_STIMOpis.Text =
DatabaseProcedures.getTextStringValue(rdr["Wypis_L_STIMOpis"]);
        textWypis_L_STIMAmplitude.Text =
DatabaseProcedures.getTextDecimalValue(rdr["Wypis_L_STIMAmplitude"]);
        textWypis_L_STIMDuration.Text =
DatabaseProcedures.getTextDecimalValue(rdr["Wypis_L_STIMDuration"]);
        textWypis_L_STIMFrequency.Text =
DatabaseProcedures.getTextDecimalValue(rdr["Wypis_L_STIMFrequency"]);
        textWypis_R_STIMOpis.Text =
DatabaseProcedures.getTextStringValue(rdr["Wypis_R_STIMOpis"]);
        textWypis_R_STIMAmplitude.Text =
DatabaseProcedures.getTextDecimalValue(rdr["Wypis_R_STIMAmplitude"]);
        textWypis_R_STIMDuration.Text =
DatabaseProcedures.getTextDecimalValue(rdr["Wypis_R_STIMDuration"]);
        textWypis_R_STIMFrequency.Text =
DatabaseProcedures.getTextDecimalValue(rdr["Wypis_R_STIMFrequency"]);
    }
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}
}

protected void buttonOK_Click(object sender, EventArgs e)
{
    savePartC();
}
protected void buttonCancel_Click(object sender, EventArgs e)
{
    Response.Redirect("~/AppointmentForm.aspx");
}
}

```

PartEForm.aspx


```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="PartEForm.aspx.cs" Inherits="PartEForm" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <h2>Kwestionariusz zaburzeń autonomicznych</h2>
    <br />
    <asp:Label ID="labelAppointment" runat="server" Text=""></asp:Label>
    <table ID="Table1" runat="server">
        <tr>
            <td>Wydzielanie śliny:</td>
            <td>
                <asp:DropDownList ID="dropWydzielanieSliny" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Dysfagia:</td>
            <td>
                <asp:DropDownList ID="dropDysfagia" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Dysfagia częstotliwość:</td>
            <td>
                <asp:DropDownList ID="dropDysfagiaCzestotliwosc" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Nudności:</td>
            <td>
                <asp:DropDownList ID="dropNudnosci" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Zaparcia:</td>
            <td>
                <asp:DropDownList ID="dropZaparcia" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Trudności w oddawaniu moczu:</td>
            <td>
                <asp:DropDownList ID="dropTrudnosciOddawaniuMoczu" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Potrzeba nagłego oddania moczu:</td>
            <td>
                <asp:DropDownList ID="dropPotrzebaNaglegoOddaniaMoczu" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Niekompletne opróżnienie pęcherza:</td>
            <td>
                <asp:DropDownList ID="dropNiekompletneOproznieniePecherza" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Słaby strumień moczu:</td>
            <td>
```

```

        <asp:DropDownList ID="dropSlabyStrumienMoczu" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Częstotliwość oddawania moczu:</td>
    <td>
        <asp:DropDownList ID="dropCzestotliwoscOddawaniaMoczu" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Nykturia:</td>
    <td>
        <asp:DropDownList ID="dropNykturia" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Niekontrolowane oddawanie moczu:</td>
    <td>
        <asp:DropDownList ID="dropNiekontrolowaneOddawanieMoczu" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Omdlenia:</td>
    <td>
        <asp:DropDownList ID="dropOmdlenia" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Zaburzenia rytmu serca:</td>
    <td>
        <asp:DropDownList ID="dropZaburzeniaRytmuSerca" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Próba pionizacyjna:</td>
    <td>
        <asp:DropDownList ID="dropProbaPionizacyjna" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Wzrost potliwości twarzy i karku:</td>
    <td>
        <asp:DropDownList ID="dropWzrostPodtliwosciTwarzKark" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Wzrost potliwości ramion i dłoni:</td>
    <td>
        <asp:DropDownList ID="dropWzrostPotliwosciRamionaDlonie" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Wzrost potliwości brzucha i pleców:</td>
    <td>
        <asp:DropDownList ID="dropWzrostPotliwosciBrzuchPlecy" runat="server">
        </asp:DropDownList>
    </td>
</tr>
<tr>
    <td>Wzrost potliwości kończyn dolnych i stóp:</td>

```

```

        <td>
            <asp:DropDownList ID="dropWzrostPotliwosciKonczynyDolneStopy" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Spadek potliwości twarzy i karku:</td>
        <td>
            <asp:DropDownList ID="dropSpadekPodtliwosciTwarzKark" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Spadek potliwości ramion i dłoni:</td>
        <td>
            <asp:DropDownList ID="dropSpadekPotliwosciRamionaDlonie" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Spadek potliwości brzucha i pleców:</td>
        <td>
            <asp:DropDownList ID="dropSpadekPotliwosciBrzuchPlecy" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Spadek potliwości kończyn dolnych i stóp:</td>
        <td>
            <asp:DropDownList ID="dropSpadekPotliwosciKonczynyDolneStopy" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Nietolerancja wysokich temperatur:</td>
        <td>
            <asp:DropDownList ID="dropNietolerancjaWysokichTemp" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Nietolerancja niskich temperatur:</td>
        <td>
            <asp:DropDownList ID="dropNietolerancjaNiskichTemp" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Łojotok:</td>
        <td>
            <asp:DropDownList ID="dropLojotok" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Spadek libido:</td>
        <td>
            <asp:DropDownList ID="dropSpadekLibido" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Kłopoty osiągnięcia erekcji:</td>
        <td>
            <asp:DropDownList ID="dropKłopotyOsiagnieciaErekcji" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>

```

```

        <td>Kłopoty utrzymania erekcji:</td>
        <td>
            <asp:DropDownList ID="dropKłopotyUtrzymaniaErekcji" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
</table>
<asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
<br />
<asp:Button ID="buttonOK" runat="server" Text="Zatwierdź"
    onclick="buttonOK_Click" />
<asp:Button ID="buttonCancel" runat="server" Text="Anuluj"
    CausesValidation="False" onclick="buttonCancel_Click" />
</asp:Content>

```

PartEForm.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

public partial class PartEForm : System.Web.UI.Page
{
    private static byte NO_DATA = 100;
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["PatientNumber"] != null && Session["AppointmentId"] != null &&
            Session["AppointmentName"] != null)
        {
            labelAppointment.Text = "Pacjent: " + Session["PatientNumber"] + "<br />Wizyta: " +
                Session["AppointmentName"];
            if (!IsPostBack)
            {
                loadPartE();
            }
        }
        else
        {
            Response.Redirect("~/Main.aspx");
        }

        if (!IsPostBack)
        {
            dropAddItems(Page, DatabaseProcedures.getEnumerationBytewithNoData("Wizyta",
                "_ObjawAutonomiczny", NO_DATA));
        }
    }

    private void dropAddItems(Control parent, Dictionary<byte, string> items)
    {
        foreach (Control control in parent.Controls)
        {
            DropDownList drop = control as DropDownList;
            if (drop != null)
            {
                drop.DataSource = items;
                drop.DataTextField = "Value";
                drop.DataValueField = "Key";
                drop.DataBind();
            }

            if (control.HasControls())
            {

```

```

        dropAddItems(control, items);
    }
}

private void savePartE()
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "[dbo].[update_examination_questionnaire_partE_x01]";
    cmd.Parameters.Add("@IdWizyta", SqlDbType.Int).Value =
int.Parse(Session["AppointmentId"].ToString());
    cmd.Parameters.Add("@WydzialanieSliny", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropWydzialanieSliny.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@Dysfagia", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropDysfagia.SelectedValue, NO_DATA.ToString());
    cmd.Parameters.Add("@DysfagiaCzestotliwosc", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropDysfagiaCzestotliwosc.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@Nudnosci", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropNudnosci.SelectedValue, NO_DATA.ToString());
    cmd.Parameters.Add("@Zaparcia", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropZaparcia.SelectedValue, NO_DATA.ToString());
    cmd.Parameters.Add("@TrudnosciWOddawaniuMoczu", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropTrudnosciWOddawaniuMoczu.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@PotrzebaNagleOddaniaMoczu", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropPotrzebaNagleOddaniaMoczu.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@NiekompletneOproznieniePecherza", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropNiekompletneOproznieniePecherza.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@SlabyStrumienMoczu", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropSlabyStrumienMoczu.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@CzestotliwoscOddawaniaMoczu", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropCzestotliwoscOddawaniaMoczu.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@Nykturia", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropNykturia.SelectedValue, NO_DATA.ToString());
    cmd.Parameters.Add("@NiekontrolowaneOddawanieMoczu", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropNiekontrolowaneOddawanieMoczu.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@Omdlenia", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropOmdlenia.SelectedValue, NO_DATA.ToString());
    cmd.Parameters.Add("@ZaburzeniaRytmuSerca", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropZaburzeniaRytmuSerca.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@ProbaPionizacyjna", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropProbaPionizacyjna.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@WzrostPodtliwosciTwarzKark", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropWzrostPodtliwosciTwarzKark.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@WzrostPotliwosciRamionaDlonie", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropWzrostPotliwosciRamionaDlonie.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@WzrostPotliwosciBrzuchPlecy", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropWzrostPotliwosciBrzuchPlecy.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@WzrostPotliwosciKonczynyDolneStopy", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropWzrostPotliwosciKonczynyDolneStopy.SelectedValue,
NO_DATA.ToString());
    cmd.Parameters.Add("@SpadekPodtliwosciTwarzKark", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropSpadekPodtliwosciTwarzKark.SelectedValue,
NO_DATA.ToString());
}

```

```

        cmd.Parameters.Add("@SpadekPotliwosciRamionaDlonie", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropSpadekPotliwosciRamionaDlonie.SelectedValue,
NO_DATA.ToString());
        cmd.Parameters.Add("@SpadekPotliwosciBrzuchPlecy", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropSpadekPotliwosciBrzuchPlecy.SelectedValue,
NO_DATA.ToString());
        cmd.Parameters.Add("@SpadekPotliwosciKonczynyDolneStopy", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropSpadekPotliwosciKonczynyDolneStopy.SelectedValue,
NO_DATA.ToString());
        cmd.Parameters.Add("@NietolerancjaWysokichTemp", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropNietolerancjaWysokichTemp.SelectedValue,
NO_DATA.ToString());
        cmd.Parameters.Add("@NietolerancjaNiskichTemp", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropNietolerancjaNiskichTemp.SelectedValue,
NO_DATA.ToString());
        cmd.Parameters.Add("@Lojotok", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropLojotok.SelectedValue, NO_DATA.ToString());
        cmd.Parameters.Add("@SpadekLibido", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropSpadekLibido.SelectedValue, NO_DATA.ToString());
        cmd.Parameters.Add("@KlopotyOsiagnieciaErekcji", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropKlopotyOsiagnieciaErekcji.SelectedValue,
NO_DATA.ToString());
        cmd.Parameters.Add("@KlopotyUtrzymanieErekcji", SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(dropKlopotyUtrzymanieErekcji.SelectedValue,
NO_DATA.ToString());
        cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
        cmd.Parameters.Add("@result", SqlDbType.Int);
        cmd.Parameters["@result"].Direction = ParameterDirection.Output;
        cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
        cmd.Parameters["@message"].Direction = ParameterDirection.Output;
        cmd.Connection = con;

        int success = 0;
        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
            success = (int)cmd.Parameters["@result"].Value;

            if (success == 0)
            {
                Response.Redirect("~/AppointmentForm.aspx");
            }
            else
            {
                labelMessage.Text = (string)cmd.Parameters["@message"].Value;
            }
        }
        catch (SqlException ex)
        {
            labelMessage.Text = ex.Message;
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }
    }

    private void loadPartE()
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.Text;

```

```

cmd.CommandText = "select WydzielanieSliny, Dysfagia, DysfagiaCzestotliwosc, Nudnosci,
Zaparcia, TrudnosciWoddawaniuMoczu, " +
    "PotrzebaNaglegoOddaniaMoczu, NiekompletneOproznieniePecherza, SlabyStrumienMoczu,
CzestotliwoscOddawaniaMoczu, Nykturia, NiekontrolowaneOddawanieMoczu, Omdlenia, " +
    "ZaburzeniaRytmuSerca, ProbaPionizacyjna, WzrostPodtlivostiTwarzKark,
WzrostPotliwosciRamionaDlonie, WzrostPotliwosciBrzuchPlecy, WzrostPotliwosciKonczynyDolneStopy, "
+
    "SpadekPotliwosciBrzuchPlecy, SpadekPodtlivostiTwarzKark,
SpadekPotliwosciRamionaDlonie, " +
    "SpadekPotliwosciKonczynyDolneStopy, NietolerancjaWysokichTemp,
NietolerancjaNiskichTemp, Lojotok, SpadekLibido, KłopotyOsiagnieciaErekcji,
KłopotyUtrzymaniaErekcji " +
    "from Wizyta where IdWizyta = " + Session["AppointmentId"];
cmd.Connection = con;

try
{
    con.Open();
    SqlDataReader rdr = cmd.ExecuteReader();
    while (rdr.Read())
    {
        dropWydzielanieSliny.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["WydzielanieSliny"], NO_DATA.ToString());
        dropDysfagia.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Dysfagia"], NO_DATA.ToString());
        dropDysfagiaCzestotliwosc.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["DysfagiaCzestotliwosc"], NO_DATA.ToString());
        dropNudnosci.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Nudnosci"], NO_DATA.ToString());
        dropZaparcia.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Zaparcia"], NO_DATA.ToString());
        dropTrudnosciWoddawaniuMoczu.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["TrudnosciWoddawaniuMoczu"],
NO_DATA.ToString());
        dropPotrzebaNaglegoOddaniaMoczu.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["PotrzebaNaglegoOddaniaMoczu"],
NO_DATA.ToString());
        dropNiekompletneOproznieniePecherza.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["NiekompletneOproznieniePecherza"],
NO_DATA.ToString());
        dropSlabyStrumienMoczu.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["SlabyStrumienMoczu"], NO_DATA.ToString());
        dropCzestotliwoscOddawaniaMoczu.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["CzestotliwoscOddawaniaMoczu"],
NO_DATA.ToString());
        dropNiekontrolowaneOddawanieMoczu.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["NiekontrolowaneOddawanieMoczu"],
NO_DATA.ToString());
        dropNykturia.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Nykturia"], NO_DATA.ToString());
        dropOmdlenia.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Omdlenia"], NO_DATA.ToString());
        dropZaburzeniaRytmuSerca.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["ZaburzeniaRytmuSerca"], NO_DATA.ToString());
        dropProbaPionizacyjna.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["ProbaPionizacyjna"], NO_DATA.ToString());
        dropWzrostPodtlivostiTwarzKark.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["WzrostPodtlivostiTwarzKark"],
NO_DATA.ToString());
        dropWzrostPotliwosciRamionaDlonie.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["WzrostPotliwosciRamionaDlonie"],
NO_DATA.ToString());
        dropWzrostPotliwosciBrzuchPlecy.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["WzrostPotliwosciBrzuchPlecy"],
NO_DATA.ToString());
        dropWzrostPotliwosciKonczynyDolneStopy.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["WzrostPotliwosciKonczynyDolneStopy"],
NO_DATA.ToString());
    }
}

```

```

                dropSpadekPodtliwosciTwarzKark.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["SpadekPodtliwosciTwarzKark"],
NO_DATA.ToString());
                dropSpadekPotliwosciRamionaDlonie.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["SpadekPotliwosciRamionaDlonie"],
NO_DATA.ToString());
                dropSpadekPotliwosciBrzuchPlecy.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["SpadekPotliwosciBrzuchPlecy"],
NO_DATA.ToString());
                dropSpadekPotliwosciKonczynyDolneStopy.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["SpadekPotliwosciKonczynyDolneStopy"],
NO_DATA.ToString());
                dropNietolerancjaWysokichTemp.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["NietolerancjaWysokichTemp"],
NO_DATA.ToString());
                dropNietolerancjaNiskichTemp.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["NietolerancjaNiskichTemp"],
NO_DATA.ToString());
                dropLojotok.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["Lojotok"], NO_DATA.ToString());
                dropSpadekLibido.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["SpadekLibido"], NO_DATA.ToString());
                dropKlopotyOsiagnieciaErekcji.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["KlopotyOsiagnieciaErekcji"],
NO_DATA.ToString());
                dropKlopotyUtrzymaniaErekcji.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["KlopotyUtrzymaniaErekcji"],
NO_DATA.ToString());
            }
        }
        catch (SqlException ex)
        {
            labelMessage.Text = ex.Message;
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }
    }

    protected void buttonOK_Click(object sender, EventArgs e)
    {
        savePartE();
    }
    protected void buttonCancel_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/AppointmentForm.aspx");
    }
}

```

PartFForm.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="PartFForm.aspx.cs" Inherits="PartFForm" %>

```

```

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <script type="text/javascript">
        window.onload = function () {
            var scrollY = parseInt('<%=Request.Form["scrollY"] %>');
            if (!isNaN(scrollY)) {
                window.scrollTo(0, scrollY);
            }
        };
        window.onscroll = function () {

```



```

var scrollY = document.body.scrollTop;
if (scrollY == 0) {
    if (window.pageYOffset) {
        scrollY = window.pageYOffset;
    }
    else {
        scrollY = (document.body.parentElement) ?
document.body.parentElement.scrollTop : 0;
    }
}
if (scrollY > 0) {
    var input = document.getElementById("scrollY");
    if (input == null) {
        input = document.createElement("input");
        input.setAttribute("type", "hidden");
        input.setAttribute("id", "scrollY");
        input.setAttribute("name", "scrollY");
        document.forms[0].appendChild(input);
    }
    input.value = scrollY;
}
};
</script>
<h2>UPDRS i skale oceny klinicznej</h2>
<br />
<asp:Label ID="labelAppointment" runat="server" Text=""></asp:Label>
<asp:Table ID="tableUPDRS" runat="server">
    <asp:TableHeaderRow>
        <asp:TableHeaderCell>BMT</asp:TableHeaderCell>
        <asp:TableHeaderCell ColumnSpan="2">ON</asp:TableHeaderCell>
        <asp:TableHeaderCell ColumnSpan="2">OFF</asp:TableHeaderCell>
    </asp:TableHeaderRow>
    <asp:TableHeaderRow>
        <asp:TableHeaderCell>DBS</asp:TableHeaderCell>
        <asp:TableHeaderCell>ON-LP</asp:TableHeaderCell>
        <asp:TableHeaderCell>OFF</asp:TableHeaderCell>
        <asp:TableHeaderCell>ON-LP</asp:TableHeaderCell>
        <asp:TableHeaderCell>OFF</asp:TableHeaderCell>
    </asp:TableHeaderRow>
</asp:Table>

<asp:Button ID="buttonCalculate" runat="server" Text="Podlicz"
    onclick="buttonCalculate_Click" />
<br />
<br />
<asp:Table ID="tableUPDRSExtra" runat="server">
</asp:Table>

<asp:Button ID="buttonSavePartA" runat="server" Text="Zatwierdź"
    onclick="buttonSavePartA_Click" />
<br />
<br />
<asp:Table ID="tablePart2" runat="server">
</asp:Table>

<asp:Button ID="buttonSavePartB" runat="server" Text="Zatwierdź"
    onclick="buttonSavePartB_Click" />
<br />
<br />
<asp:Table ID="tablePart3" runat="server">
</asp:Table>

<asp:Button ID="buttonSavePartC" runat="server" Text="Zatwierdź"
    onclick="buttonSavePartC_Click" />
<br />
<br />
<table ID="Table1" runat="server">
    <tr>
        <td>PDQ39:</td>

```

```

        <td>
            <asp:TextBox ID="textPDQ39" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>AIMS:</td>
        <td>
            <asp:TextBox ID="textAIMS" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>Epworth:</td>
        <td>
            <asp:TextBox ID="textEpworth" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>CGI:</td>
        <td>
            <asp:TextBox ID="textCGI" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>FSS:</td>
        <td>
            <asp:TextBox ID="textFSS" runat="server"></asp:TextBox>
        </td>
    </tr>
</table>
<asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
<br />
<asp:Button ID="buttonOK" runat="server" Text="Zatwierdź"
    onclick="buttonOK_Click" />
<asp:Button ID="buttonCancel" runat="server" Text="Powrót"
    CausesValidation="False" onclick="buttonCancel_Click" />
</asp:Content>

```

PartFForm.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

public partial class PartFForm : System.Web.UI.Page
{
    private static int VARIANTS = 4;
    private List<Tuple<DropDownList[], string>> UPDRSList1 = new List<Tuple<DropDownList[],
string>>();
    private List<Tuple<DropDownList[], string>> UPDRSList2 = new List<Tuple<DropDownList[],
string>>();
    private Tuple<TextBox[], string> UPDRSListCalculated1;
    private Tuple<TextBox[], string> UPDRSListCalculated2;
    private Tuple<DropDownList[], string> variantsHYscale;
    private Tuple<DropDownList[], string> variantsSchwabEnglandScale;
    private Tuple<DropDownList[], string> variantsJazzNovo;
    private Tuple<DropDownList[], string> variantsWideookulograf;
    private Tuple<DropDownList[], string> variantsLatencymeter;
    private List<Tuple<TextBox[], string>> variantsPartAList = new List<Tuple<TextBox[],
string>>();
    private List<Tuple<DropDownList[], string>> variantsPartBList = new List<Tuple<DropDownList[],
string>>();
}

```

```

        private List<Tuple<TextBox[], string>> variantsPartBList2 = new List<Tuple<TextBox[],
string>>();
        private Tuple<DropDownList[], string> variantsTremorometria;
        private List<Tuple<TextBox[], string>> variantsPartBList3 = new List<Tuple<TextBox[],
string>>();
        private Tuple<DropDownList[], string> variantsTestSchodkowy;
        private Tuple<TextBox[], string> variantsTestSchodkowyCzas1;
        private Tuple<TextBox[], string> variantsTestSchodkowyCzas2;
        private Tuple<DropDownList[], string> variantsTestMarszu;
        private Tuple<TextBox[], string> variantsTestMarszuCzas1;
        private Tuple<TextBox[], string> variantsTestMarszuCzas2;
        private Tuple<DropDownList[], string> variantsPosturografia;
        private Tuple<DropDownList[], string> variantsMotionAnalysis;
        private List<Tuple<TextBox[], string>> variantsPartCList = new List<Tuple<TextBox[],
string>>();
        private Tuple<DropDownList[], string> variantsTandemPivot;
        private static byte NO_DATA = 101; //SchwabEnglandScale 100%
        private static decimal NO_DATA_DECIMAL = 100;

        protected void Page_Load(object sender, EventArgs e)
        {
            if (Session["PatientNumber"] != null && Session["AppointmentId"] != null &&
Session["AppointmentName"] != null)
            {
                initControls();
                labelAppointment.Text = "Pacjent: " + Session["PatientNumber"] + "<br />Wizyta: " +
Session["AppointmentName"];
                if (!IsPostBack)
                {
                    loadVariantIds();
                    int[] variantIds = (int[])ViewState["VariantIds"];
                    if (variantIds != null)
                    {
                        for (int i = 0; i < variantIds.Length; i++)
                        {
                            loadPartA(variantIds[i], i);
                        }
                        for (int i = 0; i < variantIds.Length; i++)
                        {
                            loadPartB(variantIds[i], i);
                        }
                        for (int i = 0; i < variantIds.Length; i++)
                        {
                            loadPartC(variantIds[i], i);
                        }
                    }
                    else
                    {
                        initParts();
                    }
                    loadPartF();
                }
            }
            else
            {
                Response.Redirect("~/Main.aspx");
            }
        }

        private void initControls()
        {
            UPDRSList2.Add(addVariantDropDowns("UPDRS_I", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_I", NO_DATA)));
            UPDRSList2.Add(addVariantDropDowns("UPDRS_II", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_II", NO_DATA)));
            UPDRSList1.Add(addVariantDropDowns("UPDRS_18", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_18", NO_DATA)));
            UPDRSList1.Add(addVariantDropDowns("UPDRS_19", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_19", NO_DATA)));

```

```

        UPDRSList1.Add(addVariantDropDowns("UPDRS_20_FaceLipsChin", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_20_FaceLipsChin", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_20_RHand", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_20_RHand", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_20_LHand", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_20_LHand", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_20_RFoot", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_20_RFoot", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_20_LFoot", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_20_LFoot", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_21_RHand", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_21_RHand", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_21_LHand", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_21_LHand", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_22_Neck", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_22_Neck", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_22_RHand", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_22_RHand", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_22_LHand", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_22_LHand", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_22_RFoot", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_22_RFoot", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_22_LFoot", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_22_LFoot", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_23_R", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_23_R", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_23_L", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_23_L", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_24_R", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_24_R", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_24_L", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_24_L", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_25_R", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_25_R", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_25_L", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_25_L", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_26_R", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_26_R", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_26_L", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_26_L", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_27", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_27", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_28", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_28", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_29", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_29", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_30", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_30", NO_DATA)));
        UPDRSList1.Add(addVariantDropDowns("UPDRS_31", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_31", NO_DATA)));
        UPDRSListCalculated1 = addVariantTextBoxes("UPDRS_III", tableUPDRS, false);
        UPDRSList2.Add(addVariantDropDowns("UPDRS_IV", tableUPDRS,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "UPDRS_IV", NO_DATA)));
        UPDRSListCalculated2 = addVariantTextBoxes("UPDRS_TOTAL", tableUPDRS, false);

        variantsHYscale = addVariantDropDowns("HYscale", tableUPDRSExtra,
DatabaseProcedures.getEnumerationDecimalWithNoData("Badanie", "HYscale", NO_DATA_DECIMAL));
        variantsSchwabEnglandScale = addVariantDropDowns("SchwabEnglandScale", tableUPDRSExtra,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "SchwabEnglandScale", NO_DATA));
        Dictionary<byte, string> dictionaryYesNo = new Dictionary<byte, string>();
        dictionaryYesNo.Add(2, "");
        dictionaryYesNo.Add(0, "nie");
        dictionaryYesNo.Add(1, "tak");
        variantsJazzNovo = addVariantDropDowns("JazzNovo", tableUPDRSExtra, dictionaryYesNo);
        variantsWideookulograf = addVariantDropDowns("Wideookulograf", tableUPDRSExtra,
dictionaryYesNo);
        variantsLatencymeter = addVariantDropDowns("Latencymeter", tableUPDRSExtra,
dictionaryYesNo);

```

```

        variantsPartAList.Add(addVariantTextBoxes("LatencymeterDurationLEFT", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterLatencyLEFT", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterAmplitudeLEFT", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterPeakVelocityLEFT", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterDurationRIGHT", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterLatencyRIGHT", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterAmplitudeRIGHT", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterPeakVelocityRIGHT",
tableUPDRSEExtra, true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterDurationALL", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterLatencyALL", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterAmplitudeALL", tableUPDRSEExtra,
true));
        variantsPartAList.Add(addVariantTextBoxes("LatencymeterPeakVelocityALL", tableUPDRSEExtra,
true));

        variantsTremorometria = addVariantDropDowns("Tremorometria", tablePart2, dictionaryYesNo);
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_0_1", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_1_2", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_2_3", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_3_4", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_4_5", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_5_6", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_6_7", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_7_8", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_8_9", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_9_10", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaLEFT_23_24", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_0_1", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_1_2", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_2_3", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_3_4", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_4_5", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_5_6", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_6_7", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_7_8", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_8_9", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_9_10", tablePart2, true));
        variantsPartBList3.Add(addVariantTextBoxes("TremorometriaRIGHT_23_24", tablePart2, true));

        variantsTestSchodkowy = addVariantDropDowns("TestSchodkowy", tablePart2, dictionaryYesNo);
        variantsTestSchodkowyCzas1 = addVariantTextBoxes("TestSchodkowyCzas1", tablePart2, true);
        variantsTestSchodkowyCzas2 = addVariantTextBoxes("TestSchodkowyCzas2", tablePart2, true);
        variantsTestMarszu = addVariantDropDowns("TestMarszu", tablePart2, dictionaryYesNo);
        variantsTestMarszuCzas1 = addVariantTextBoxes("TestMarszuCzas1", tablePart2, true);
        variantsTestMarszuCzas2 = addVariantTextBoxes("TestMarszuCzas2", tablePart2, true);
        variantsPosturografia = addVariantDropDowns("Posturografia", tablePart2, dictionaryYesNo);
        variantsMotionAnalysis = addVariantDropDowns("MotionAnalysis", tablePart2,
dictionaryYesNo);
        variantsPartBList.Add(variantsTremorometria);
        variantsPartBList.Add(variantsTestSchodkowy);
        variantsPartBList.Add(variantsTestMarszu);
        variantsPartBList.Add(variantsPosturografia);
        variantsPartBList.Add(variantsMotionAnalysis);
        variantsPartBList2.Add(variantsTestSchodkowyCzas1);
        variantsPartBList2.Add(variantsTestSchodkowyCzas2);
        variantsPartBList2.Add(variantsTestMarszuCzas1);
        variantsPartBList2.Add(variantsTestMarszuCzas2);

        variantsPartCList.Add(addVariantTextBoxes("UpAndGo", tablePart3, true));

```

```

        variantsPartCList.Add(addVariantTextBoxes("UpAndGoLiczby", tablePart3, true));
        variantsPartCList.Add(addVariantTextBoxes("UpAndGoKubekPrawa", tablePart3, true));
        variantsPartCList.Add(addVariantTextBoxes("UpAndGoKubekLewa", tablePart3, true));
        variantsPartCList.Add(addVariantTextBoxes("TST", tablePart3, true));
        variantsTandemPivot = addVariantDropDowns("TandemPivot", tablePart3,
DatabaseProcedures.getEnumerationByteWithNoData("Badanie", "TandemPivot", NO_DATA));
        variantsPartCList.Add(addVariantTextBoxes("WTT", tablePart3, true));

        disablePhases();

    }

    private TableRow addVariantRow(String label, Table table)
    {
        TableRow row = new TableRow();
        table.CellPadding = 3;
        table.CellSpacing = 0;
        if (table.Rows.Count % 2 == 0)
        {
            row.BackColor = System.Drawing.Color.FromArgb(255, 255, 240, 178);
        }
        table.Rows.Add(row);
        Label labelVariant = new Label();
        labelVariant.Text = label;
        labelVariant.Width = new Unit(200);
        TableCell cell = new TableCell();
        row.Controls.Add(cell);
        cell.Controls.Add(labelVariant);

        return row;
    }

    private Tuple<DropDownList[], string> addVariantDropDowns(String label, Table table, object
dictionary)
    {
        TableRow row = addVariantRow(label, table);
        DropDownList[] columns = new DropDownList[VARIANTS];
        for (int i = 0; i < VARIANTS; i++)
        {
            TableCell cell = new TableCell();
            row.Controls.Add(cell);
            DropDownList drop = new DropDownList();
            drop.Width = new Unit(104);
            drop.DataSource = dictionary;
            drop.DataTextField = "Value";
            drop.DataValueField = "Key";
            drop.DataBind();
            //drop.AutoPostBack = true;
            //drop.SelectedIndexChanged += new
System.EventHandler(dropUPDRS_SelectedIndexChanged);
            cell.Controls.Add(drop);
            columns[i] = drop;
        }

        Tuple<DropDownList[], string> tuple = new Tuple<DropDownList[], string>(columns, label);

        return tuple;
    }

    //protected void dropUPDRS_SelectedIndexChanged(object sender, EventArgs e)
    //{
    //    calculateUPDRS();
    //}

    private Tuple<TextBox[], string> addVariantTextBoxes(String label, Table table, bool enabled)
    {
        TableRow row = addVariantRow(label, table);
        TextBox[] columns = new TextBox[VARIANTS];

```

```

        for (int i = 0; i < VARIANTS; i++)
        {
            TableCell cell = new TableCell();
            row.Controls.Add(cell);
            TextBox text = new TextBox();
            text.Enabled = enabled;
            text.Width = new Unit(100);
            cell.Controls.Add(text);
            columns[i] = text;
        }

        Tuple<TextBox[], string> tuple = new Tuple<TextBox[], string>(columns, label);

        return tuple;
    }

    protected void buttonCalculate_Click(object sender, EventArgs e)
    {
        calculateUPDRS();
    }

    private void calculateUPDRS()
    {
        for (int i = 0; i < VARIANTS; i++)
        {
            UPDRSListCalculated1.Item1[i].Text = calculateSum(UPDRSList1, i).ToString();
            UPDRSListCalculated2.Item1[i].Text = calculateSum(UPDRSList2, i).ToString();
        }
    }

    private int calculateSum(List<Tuple<DropDownList[], string>> UPDRSList, int variant)
    {
        int sum = 0;
        foreach (Tuple<DropDownList[], string> dropVariants in UPDRSList)
        {
            int value = int.Parse(dropVariants.Item1[variant].SelectedValue);
            if (value != NO_DATA)
            {
                sum += value;
            }
        }

        return sum;
    }

    private int saveVariantPartA(byte DBS, bool BMT, int variant)
    {
        SqlConnection con = new
        SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[update_variant_examination_data_partA]";
        cmd.Parameters.Add("@IdWizyta", SqlDbType.Int).Value =
        int.Parse(Session["AppointmentId"].ToString());
        cmd.Parameters.Add("@DBS", SqlDbType.TinyInt).Value = DBS;
        cmd.Parameters.Add("@BMT", SqlDbType.Bit).Value = BMT;
        for (int i = 0; i < UPDRSList2.Count; i++)
        {
            cmd.Parameters.Add("@ " + UPDRSList2[i].Item2, SqlDbType.TinyInt).Value =
            DatabaseProcedures.getBytesOrNullWithNoData(UPDRSList2[i].Item1[variant].SelectedValue,
            NO_DATA.ToString());
        }
        for (int i = 0; i < UPDRSList1.Count; i++)
        {
            cmd.Parameters.Add("@ " + UPDRSList1[i].Item2, SqlDbType.TinyInt).Value =
            DatabaseProcedures.getBytesOrNullWithNoData(UPDRSList1[i].Item1[variant].SelectedValue,
            NO_DATA.ToString());
        }
    }

```



```

        cmd.Parameters.Add("@ " + UPDRSListCalculated1.Item2, SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNull(UPDRSListCalculated1.Item1[variant].Text);
        cmd.Parameters.Add("@ " + UPDRSListCalculated2.Item2, SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNull(UPDRSListCalculated2.Item1[variant].Text);
        SqlParameter HYscaleDecimal = new SqlParameter("@ " + variantsHYscale.Item2,
SqlDbType.Decimal);
        HYscaleDecimal.Precision = 2;
        HYscaleDecimal.Scale = 1;
        HYscaleDecimal.Value =
DatabaseProcedures.getDecimalOrNullWithNoData(variantsHYscale.Item1[variant].SelectedValue,
NO_DATA_DECIMAL.ToString());
        cmd.Parameters.Add(HYscaleDecimal);
        cmd.Parameters.Add("@ " + variantsSchwabEnglandScale.Item2, SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(variantsSchwabEnglandScale.Item1[variant].SelectedValue
, NO_DATA.ToString());
        cmd.Parameters.Add("@ " + variantsJazzNovo.Item2, SqlDbType.TinyInt).Value =
DatabaseProcedures.getBitOrNull(variantsJazzNovo.Item1[variant].SelectedValue);
        cmd.Parameters.Add("@ " + variantsWideokulograf.Item2, SqlDbType.TinyInt).Value =
DatabaseProcedures.getBitOrNull(variantsWideokulograf.Item1[variant].SelectedValue);
        cmd.Parameters.Add("@ " + variantsLatencymeter.Item2, SqlDbType.TinyInt).Value =
DatabaseProcedures.getBitOrNull(variantsLatencymeter.Item1[variant].SelectedValue);
        for (int i = 0; i < variantsPartAList.Count; i++)
        {
            SqlParameter latencymeterDecimal = new SqlParameter("@ " + variantsPartAList[i].Item2,
SqlDbType.Decimal);
            latencymeterDecimal.Precision = 6;
            latencymeterDecimal.Scale = 2;
            latencymeterDecimal.Value =
DatabaseProcedures.getDecimalOrNull(variantsPartAList[i].Item1[variant].Text);
            cmd.Parameters.Add(latencymeterDecimal);
        }
        bool update = false;
        if (ViewState["VariantIds"] != null)
        {
            update = true;
        }
        cmd.Parameters.Add("@allow_update_existing", SqlDbType.Bit).Value = update;
        cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
        cmd.Parameters.Add("@result", SqlDbType.Int);
        cmd.Parameters["@result"].Direction = ParameterDirection.Output;
        cmd.Parameters.Add("@variant_id", SqlDbType.Int);
        cmd.Parameters["@variant_id"].Direction = ParameterDirection.Output;
        cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
        cmd.Parameters["@message"].Direction = ParameterDirection.Output;
        cmd.Connection = con;

        int success = 0;
        int variantId = 0;
        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
            success = (int)cmd.Parameters["@result"].Value;
            variantId = (int)cmd.Parameters["@variant_id"].Value;

            if (success == 0)
            {
            }
            else
            {
                labelMessage.Text = (string)cmd.Parameters["@message"].Value;
            }
        }
        catch (SqlException ex)
        {
            labelMessage.Text = ex.Message;
        }
        finally

```



```

        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }

        return variantId;
    }

    private void saveVariantPartB(int examinationId, int variant)
    {
        SqlConnection con = new
        SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[update_variant_examination_data_partB]";
        cmd.Parameters.Add("@IdBadanie", SqlDbType.Int).Value = examinationId;
        for (int i = 0; i < variantsPartBList.Count; i++)
        {
            cmd.Parameters.Add("@ " + variantsPartBList[i].Item2, SqlDbType.Bit).Value =
            DatabaseProcedures.getBitOrNull(variantsPartBList[i].Item1[variant].SelectedValue);
        }

        for (int i = 0; i < variantsPartBList2.Count; i++)
        {
            SqlParameter testDecimal = new SqlParameter("@ " + variantsPartBList2[i].Item2,
            SqlDbType.Decimal);
            testDecimal.Precision = 4;
            testDecimal.Scale = 2;
            testDecimal.Value =
            DatabaseProcedures.getDecimalOrNull(variantsPartBList2[i].Item1[variant].Text);
            cmd.Parameters.Add(testDecimal);
        }

        for (int i = 0; i < variantsPartBList3.Count; i++)
        {
            SqlParameter tremorometriaDecimal = new SqlParameter("@ " +
            variantsPartBList3[i].Item2, SqlDbType.Decimal);
            tremorometriaDecimal.Precision = 7;
            tremorometriaDecimal.Scale = 2;
            tremorometriaDecimal.Value =
            DatabaseProcedures.getDecimalOrNull(variantsPartBList3[i].Item1[variant].Text);
            cmd.Parameters.Add(tremorometriaDecimal);
        }

        cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
        cmd.Parameters.Add("@result", SqlDbType.Int);
        cmd.Parameters["@result"].Direction = ParameterDirection.Output;
        cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
        cmd.Parameters["@message"].Direction = ParameterDirection.Output;
        cmd.Connection = con;

        int success = 0;
        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
            success = (int)cmd.Parameters["@result"].Value;

            if (success == 0)
            {
            }
            else
            {
                labelMessage.Text = (string)cmd.Parameters["@message"].Value;
            }
        }
    }

```

```

    }
    catch (SqlException ex)
    {
        labelMessage.Text = ex.Message;
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }
}

private void saveVariantPartC(int examinationId, int variant)
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "[dbo].[update_variant_examination_data_partC]";
    cmd.Parameters.Add("@IdBadanie", SqlDbType.Int).Value = examinationId;
    for (int i = 0; i < variantsPartCList.Count; i++)
    {
        SqlParameter UpAndGoDecimal = new SqlParameter("@ " + variantsPartCList[i].Item2,
SqlDbType.Decimal);
        UpAndGoDecimal.Precision = 3;
        UpAndGoDecimal.Scale = 1;
        UpAndGoDecimal.Value =
DatabaseProcedures.getDecimalOrNull(variantsPartCList[i].Item1[variant].Text);
        cmd.Parameters.Add(UpAndGoDecimal);
    }
    cmd.Parameters.Add("@ " + variantsTandemPivot.Item2, SqlDbType.TinyInt).Value =
DatabaseProcedures.getBytesOrNullWithNoData(variantsTandemPivot.Item1[variant].Text,
NO_DATA.ToString());
    cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
    cmd.Parameters.Add("@result", SqlDbType.Int);
    cmd.Parameters["@result"].Direction = ParameterDirection.Output;
    cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
    cmd.Parameters["@message"].Direction = ParameterDirection.Output;
    cmd.Connection = con;

    int success = 0;
    try
    {
        con.Open();
        cmd.ExecuteNonQuery();
        success = (int)cmd.Parameters["@result"].Value;

        if (success == 0)
        {
        }
        else
        {
            labelMessage.Text = (string)cmd.Parameters["@message"].Value;
        }
    }
    catch (SqlException ex)
    {
        labelMessage.Text = ex.Message;
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }
}

```

```

    }
}

private void savePartF()
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.CommandText = "[dbo].[update_examination_questionnaire_partF]";
    cmd.Parameters.Add("@IdWizyta", SqlDbType.Int).Value =
int.Parse(Session["AppointmentId"].ToString());
    cmd.Parameters.Add("@PDQ39", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textPDQ39.Text);
    cmd.Parameters.Add("@AIMS", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textAIMS.Text);
    cmd.Parameters.Add("@Epworth", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textEpworth.Text);
    cmd.Parameters.Add("@CGI", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textCGI.Text);
    SqlParameter FSSDecimal = new SqlParameter("@FSS", SqlDbType.Decimal);
    FSSDecimal.Precision = 3;
    FSSDecimal.Scale = 1;
    FSSDecimal.Value = DatabaseProcedures.getDecimalOrNull(textFSS.Text);
    cmd.Parameters.Add(FSSDecimal);
    cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
    cmd.Parameters.Add("@result", SqlDbType.Int);
    cmd.Parameters["@result"].Direction = ParameterDirection.Output;
    cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
    cmd.Parameters["@message"].Direction = ParameterDirection.Output;
    cmd.Connection = con;

    int success = 0;
    try
    {
        con.Open();
        cmd.ExecuteNonQuery();
        success = (int)cmd.Parameters["@result"].Value;

        if (success == 0)
        {
            Response.Redirect("~/AppointmentForm.aspx");
        }
        else
        {
            labelMessage.Text = (string)cmd.Parameters["@message"].Value;
        }
    }
    catch (SqlException ex)
    {
        labelMessage.Text = ex.Message;
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }
}

private void loadVariantIds()
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.Text;

```

```

cmd.CommandText = "select IdBadanie, DBS, BMT " +
    "from Badanie where IdWizyta = " + Session["AppointmentId"];
cmd.Connection = con;

try
{
    con.Open();
    SqlDataReader rdr = cmd.ExecuteReader();

    int[] variantIds = new int[VARIANTS];
    int i = 0;
    while (rdr.Read() && i < VARIANTS)
    {
        byte DBS = (byte)rdr["DBS"];
        bool BMT = (bool)rdr["BMT"];
        int variantId = (int)rdr["IdBadanie"];
        variantIds[getVariantColumn(DBS, BMT)] = variantId;
        i++;
    }
    if (i > 0)
    {
        ViewState["VariantIds"] = variantIds;
    }
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}

private int getVariantColumn(byte DBS, bool BMT)
{
    if (BMT)
    {
        if (DBS == 1)
        {
            return 0;
        }
        else if (DBS == 0)
        {
            return 1;
        }
    }
    else
    {
        if (DBS == 1)
        {
            return 2;
        }
        else if (DBS == 0)
        {
            return 3;
        }
    }

    return 0;
}

private void loadPartA(int variantId, int variant)
{

```

```

SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
SqlCommand cmd = new SqlCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "select DBS, " +
"BMT, " +
"UPDRS_I, " +
"UPDRS_II, " +
"UPDRS_18, " +
"UPDRS_19, " +
"UPDRS_20_FaceLipsChin, " +
"UPDRS_20_RHand, " +
"UPDRS_20_LHand, " +
"UPDRS_20_RFoot, " +
"UPDRS_20_LFoot, " +
"UPDRS_21_RHand, " +
"UPDRS_21_LHand, " +
"UPDRS_22_Neck, " +
"UPDRS_22_RHand, " +
"UPDRS_22_LHand, " +
"UPDRS_22_RFoot, " +
"UPDRS_22_LFoot, " +
"UPDRS_23_R, " +
"UPDRS_23_L, " +
"UPDRS_24_R, " +
"UPDRS_24_L, " +
"UPDRS_25_R, " +
"UPDRS_25_L, " +
"UPDRS_26_R, " +
"UPDRS_26_L, " +
"UPDRS_27, " +
"UPDRS_28, " +
"UPDRS_29, " +
"UPDRS_30, " +
"UPDRS_31, " +
"UPDRS_III, " +
"UPDRS_IV, " +
"UPDRS_TOTAL, " +
"HYscale, " +
"SchwabEnglandScale, " +
"JazzNovo, " +
"Wideookulograf, " +
"Latencymeter, " +
"LatencymeterDurationLEFT, " +
"LatencymeterLatencyLEFT, " +
"LatencymeterAmplitudeLEFT, " +
"LatencymeterPeakVelocityLEFT, " +
"LatencymeterDurationRIGHT, " +
"LatencymeterLatencyRIGHT, " +
"LatencymeterAmplitudeRIGHT, " +
"LatencymeterPeakVelocityRIGHT, " +
"LatencymeterDurationALL, " +
"LatencymeterLatencyALL, " +
"LatencymeterAmplitudeALL, " +
"LatencymeterPeakVelocityALL " +
"from Badanie where IdBadanie = " + variantId;
cmd.Connection = con;

try
{
    con.Open();
    SqlDataReader rdr = cmd.ExecuteReader();
    while (rdr.Read())
    {
        for (int i = 0; i < UPDRSList2.Count; i++)
        {
            UPDRSList2[i].Item1[variant].SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr[UPDRSList2[i].Item2], NO_DATA.ToString());
        }
    }
}

```

```

        for (int i = 0; i < UPDRSList1.Count; i++)
        {
            UPDRSList1[i].Item1[variant].SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr[UPDRSList1[i].Item2], NO_DATA.ToString());
        }
        UPDRSListCalculated1.Item1[variant].Text =
DatabaseProcedures.getTextByteValue(rdr[UPDRSListCalculated1.Item2]);
        UPDRSListCalculated2.Item1[variant].Text =
DatabaseProcedures.getTextByteValue(rdr[UPDRSListCalculated2.Item2]);
        variantsHYScale.Item1[variant].SelectedValue =
DatabaseProcedures.getDropDecimalValueWithNoData(rdr[variantsHYScale.Item2],
NO_DATA_DECIMAL.ToString());
        variantsSchwabEnglandScale.Item1[variant].SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr[variantsSchwabEnglandScale.Item2],
NO_DATA.ToString());
        variantsJazzNovo.Item1[variant].SelectedValue =
DatabaseProcedures.getDropBitValue(rdr[variantsJazzNovo.Item2]);
        variantsWideokulograf.Item1[variant].SelectedValue =
DatabaseProcedures.getDropBitValue(rdr[variantsWideokulograf.Item2]);
        variantsLatencymeter.Item1[variant].SelectedValue =
DatabaseProcedures.getDropBitValue(rdr[variantsLatencymeter.Item2]);
        for (int i = 0; i < variantsPartAList.Count; i++)
        {
            variantsPartAList[i].Item1[variant].Text =
DatabaseProcedures.getTextDecimalValue(rdr[variantsPartAList[i].Item2]);
        }
    }
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}

private void loadPartB(int variantId, int variant)
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select Tremorometria, " +
        "TremorometriaLEFT_0_1, " +
        "TremorometriaLEFT_1_2, " +
        "TremorometriaLEFT_2_3, " +
        "TremorometriaLEFT_3_4, " +
        "TremorometriaLEFT_4_5, " +
        "TremorometriaLEFT_5_6, " +
        "TremorometriaLEFT_6_7, " +
        "TremorometriaLEFT_7_8, " +
        "TremorometriaLEFT_8_9, " +
        "TremorometriaLEFT_9_10, " +
        "TremorometriaLEFT_23_24, " +
        "TremorometriaRIGHT_0_1, " +
        "TremorometriaRIGHT_1_2, " +
        "TremorometriaRIGHT_2_3, " +
        "TremorometriaRIGHT_3_4, " +
        "TremorometriaRIGHT_4_5, " +
        "TremorometriaRIGHT_5_6, " +
        "TremorometriaRIGHT_6_7, " +
        "TremorometriaRIGHT_7_8, " +
        "TremorometriaRIGHT_8_9, " +

```

```

        "TremorometriaRIGHT_9_10, " +
        "TremorometriaRIGHT_23_24, " +
        "TestSchodkowy, " +
        "TestSchodkowyCzas1, " +
        "TestSchodkowyCzas2, " +
        "TestMarszu, " +
        "TestMarszuCzas1, " +
        "TestMarszuCzas2, " +
        "Posturografia, " +
        "MotionAnalysis " +
        "from Badanie where IdBadanie = " + variantId;
cmd.Connection = con;

try
{
    con.Open();
    SqlDataReader rdr = cmd.ExecuteReader();
    while (rdr.Read())
    {
        for (int i = 0; i < variantsPartBList.Count; i++)
        {
            variantsPartBList[i].Item1[variant].SelectedValue =
DatabaseProcedures.getDropBitValue(rdr[variantsPartBList[i].Item2]);
        }
        for (int i = 0; i < variantsPartBList2.Count; i++)
        {
            variantsPartBList2[i].Item1[variant].Text =
DatabaseProcedures.getTextDecimalValue(rdr[variantsPartBList2[i].Item2]);
        }
        for (int i = 0; i < variantsPartBList3.Count; i++)
        {
            variantsPartBList3[i].Item1[variant].Text =
DatabaseProcedures.getTextDecimalValue(rdr[variantsPartBList3[i].Item2]);
        }
    }
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}
}

private void loadPartC(int variantId, int variant)
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
SqlCommand cmd = new SqlCommand();
cmd.CommandType = CommandType.Text;
cmd.CommandText = "select UpAndGo, " +
        "UpAndGoLiczby, " +
        "UpAndGoKubekPrawa, " +
        "UpAndGoKubekLewa, " +
        "TST, " +
        "TandemPivot, " +
        "WTT " +
        "from Badanie where IdBadanie = " + variantId;
cmd.Connection = con;

try
{
    con.Open();

```

```

        SqlDataReader rdr = cmd.ExecuteReader();
        while (rdr.Read())
        {
            for (int i = 0; i < variantsPartCList.Count; i++)
            {
                variantsPartCList[i].Item1[variant].Text =
                DatabaseProcedures.getTextDecimalValue(rdr[variantsPartCList[i].Item2]);
            }
            variantsTandemPivot.Item1[variant].SelectedValue =
            DatabaseProcedures.getDropMultiValueWithNoData(rdr[variantsTandemPivot.Item2],
            NO_DATA.ToString());
        }
    }
    catch (SqlException ex)
    {
        labelMessage.Text = ex.Message;
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }
}

private void loadPartF()
{
    SqlConnection con = new
    SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select PDQ39, AIMS, Epworth, CGI, FSS " +
        "from Wizyta where IdWizyta = " + Session["AppointmentId"];
    cmd.Connection = con;

    try
    {
        con.Open();
        SqlDataReader rdr = cmd.ExecuteReader();
        while (rdr.Read())
        {
            textPDQ39.Text = DatabaseProcedures.getTextByteValue(rdr["PDQ39"]);
            textAIMS.Text = DatabaseProcedures.getTextByteValue(rdr["AIMS"]);
            textEpworth.Text = DatabaseProcedures.getTextByteValue(rdr["Epworth"]);
            textCGI.Text = DatabaseProcedures.getTextByteValue(rdr["CGI"]);
            textFSS.Text = DatabaseProcedures.getTextDecimalValue(rdr["FSS"]);
        }
    }
    catch (SqlException ex)
    {
        labelMessage.Text = ex.Message;
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }
}

private void initParts()
{
    buttonSavePartB.Enabled = false;
    buttonSavePartC.Enabled = false;
}

```



```

for (int variant = 0; variant < VARIANTS; variant++)
{
    for (int i = 0; i < UPDRSList2.Count; i++)
    {
        UPDRSList2[i].Item1[variant].SelectedValue = NO_DATA.ToString();
    }
    for (int i = 0; i < UPDRSList1.Count; i++)
    {
        UPDRSList1[i].Item1[variant].SelectedValue = NO_DATA.ToString();
    }
    variantsHYscale.Item1[variant].SelectedValue = NO_DATA_DECIMAL.ToString();
    variantsSchwabEnglandScale.Item1[variant].SelectedValue = NO_DATA.ToString();
    variantsJazzNovo.Item1[variant].SelectedValue = NO_DATA.ToString();
    variantsWideokulograf.Item1[variant].SelectedValue = NO_DATA.ToString();
    variantsLatencymeter.Item1[variant].SelectedValue = NO_DATA.ToString();
    variantsTandemPivot.Item1[variant].SelectedValue = NO_DATA.ToString();
}
}

private void disablePhases()
{
    List<int> variantList = new List<int>();

    if (Session["AppointmentName"].ToString() == "przedoperacyjna")
    {
        variantList.Add(getVariantColumn(1, true));
        variantList.Add(getVariantColumn(1, false));
        foreach (Tuple<DropDownList[], string> tuple in UPDRSList1)
        {
            disableWebControlVariants(tuple.Item1, variantList);
        }
        foreach (Tuple<DropDownList[], string> tuple in UPDRSList2)
        {
            disableWebControlVariants(tuple.Item1, variantList);
        }
        foreach (Tuple<TextBox[], string> tuple in variantsPartAList)
        {
            disableWebControlVariants(tuple.Item1, variantList);
        }
        foreach (Tuple<DropDownList[], string> tuple in variantsPartBList)
        {
            disableWebControlVariants(tuple.Item1, variantList);
        }
        foreach (Tuple<TextBox[], string> tuple in variantsPartBList2)
        {
            disableWebControlVariants(tuple.Item1, variantList);
        }
        foreach (Tuple<TextBox[], string> tuple in variantsPartBList3)
        {
            disableWebControlVariants(tuple.Item1, variantList);
        }
        disableWebControlVariants(variantsHYscale.Item1, variantList);
        disableWebControlVariants(variantsSchwabEnglandScale.Item1, variantList);
        disableWebControlVariants(variantsJazzNovo.Item1, variantList);
        disableWebControlVariants(variantsWideokulograf.Item1, variantList);
        disableWebControlVariants(variantsLatencymeter.Item1, variantList);

        disableWebControlVariants(variantsTestSchodkowy.Item1, variantList);
        disableWebControlVariants(variantsTestSchodkowyCzas1.Item1, variantList);
        disableWebControlVariants(variantsTestSchodkowyCzas2.Item1, variantList);
        disableWebControlVariants(variantsTestMarszu.Item1, variantList);
        disableWebControlVariants(variantsTestMarszuCzas1.Item1, variantList);
        disableWebControlVariants(variantsTestMarszuCzas2.Item1, variantList);
        disableWebControlVariants(variantsPosturografia.Item1, variantList);
        disableWebControlVariants(variantsMotionAnalysis.Item1, variantList);
        foreach (Tuple<TextBox[], string> tuple in variantsPartC in variantsPartCList)
        {
            disableWebControlVariants(variantsPartC.Item1, variantList);
        }
    }
}

```

```

        disableWebControlVariants(variantsTandemPivot.Item1, variantList);
    }
}

private void disableWebControlVariants(WebControl[] webControlVariants, List<int> variantList)
{
    foreach (int i in variantList)
    {
        webControlVariants[i].Enabled = false;
    }
}

protected void buttonOK_Click(object sender, EventArgs e)
{
    savePartF();
}

protected void buttonCancel_Click(object sender, EventArgs e)
{
    Response.Redirect("~/AppointmentForm.aspx");
}

protected void buttonSavePartA_Click(object sender, EventArgs e)
{
    calculateUPDRS();
    int[] variantIds = new int[VARIANTS];
    for (byte DBS = 0, i = 0; i < VARIANTS; DBS++, i++)
    {
        bool BMT = true;
        int variant = getVariantColumn(DBS, BMT);
        variantIds[variant] = saveVariantPartA(DBS, BMT, variant);
        i++;
        BMT = false;
        variant = getVariantColumn(DBS, BMT);
        variantIds[variant] = saveVariantPartA(DBS, BMT, variant);
    }
    if (variantIds.Length > 0)
    {
        ViewState["VariantIds"] = variantIds;
        buttonSavePartB.Enabled = true;
        buttonSavePartC.Enabled = true;
    }
}

protected void buttonSavePartB_Click(object sender, EventArgs e)
{
    int[] variantIds = (int[])ViewState["VariantIds"];
    if (variantIds != null)
    {
        for (int i = 0; i < variantIds.Length; i++)
        {
            saveVariantPartB(variantIds[i], i);
            i++;
            saveVariantPartB(variantIds[i], i);
        }
    }
}

protected void buttonSavePartC_Click(object sender, EventArgs e)
{
    int[] variantIds = (int[])ViewState["VariantIds"];
    if (variantIds != null)
    {
        for (int i = 0; i < variantIds.Length; i++)
        {
            saveVariantPartC(variantIds[i], i);
            i++;
            saveVariantPartC(variantIds[i], i);
        }
    }
}
}
}

```

PartGForm.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="PartGForm.aspx.cs" Inherits="PartGForm" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <h2>Badania i testy psychologiczne</h2>
    <br />
    <asp:Label ID="labelAppointment" runat="server" Text=""></asp:Label>
    <table ID="Table1" runat="server">
        <tr>
            <td>Test zegara:</td>
            <td>
                <asp:DropDownList ID="dropTestZegara" runat="server">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>
                    <asp:ListItem Value="1">tak</asp:ListItem>
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>MMSE:</td>
            <td>
                <asp:TextBox ID="textMMSE" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>WAIS-R - wiadomości:</td>
            <td>
                <asp:TextBox ID="textWAISR_Wiadomosci" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>WAIS-R - powtarzanie cyfr:</td>
            <td>
                <asp:TextBox ID="textWAISR_PowtarzanieCyfr" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>Skala depresji Becka:</td>
            <td>
                <asp:TextBox ID="textSkalaDepresjiBecka" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>Test fluencji zwierzęta:</td>
            <td>
                <asp:TextBox ID="textTestFluencjiZwierzeta" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>Test fluencji ostre:</td>
            <td>
                <asp:TextBox ID="textTestFluencjiOstre" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>Test fluencji K:</td>
            <td>
                <asp:TextBox ID="textTestFluencjiK" runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>Test łączenia punktów A:</td>
            <td>
                <asp:TextBox ID="textTestLaczeniaPunktowA" runat="server"></asp:TextBox>
            </td>
        </tr>
    </table>
</asp:Content>
```

```

<tr>
    <td>Test łączenia punktów B:</td>
    <td>
        <asp:TextBox ID="textTestLaczeniaPunktowB" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Test uczenia słowno-słuchowego:</td>
    <td>
        <asp:TextBox ID="textTestUczeniaSłownoSłuchowego" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Test Stroopa:</td>
    <td>
        <asp:TextBox ID="textTestStroopa" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Test Minnesota:</td>
    <td>
        <asp:TextBox ID="textTestMinnesota" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Inne psychologiczne:</td>
    <td>
        <asp:TextBox ID="textInnePsychologiczne" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Opis badania:</td>
    <td>
        <asp:TextBox ID="textOpisBadania" runat="server" Rows="5"
        TextMode="MultiLine"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Wnioski:</td>
    <td>
        <asp:TextBox ID="textWnioski" runat="server" Rows="5"
        TextMode="MultiLine"></asp:TextBox>
    </td>
</tr>
</table>
<asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
<br />
<asp:Button ID="buttonOK" runat="server" Text="Zatwierdź"
    onclick="buttonOK_Click" />
<asp:Button ID="buttonCancel" runat="server" Text="Anuluj"
    CausesValidation="False" onclick="buttonCancel_Click" />
</asp:Content>

```

PartGForm.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

```

```

public partial class PartGForm : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

```

```

        if (Session["PatientNumber"] != null && Session["AppointmentId"] != null &&
Session["AppointmentName"] != null)
        {
            labelAppointment.Text = "Pacjent: " + Session["PatientNumber"] + "<br />Wizyta: " +
Session["AppointmentName"];
            if (!IsPostBack)
            {
                loadPartG();
            }
        }
        else
        {
            Response.Redirect("~/Main.aspx");
        }
    }

    private void savePartG()
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[update_examination_questionnaire_partG]";
        cmd.Parameters.Add("@IdWizyta", SqlDbType.Int).Value =
int.Parse(Session["AppointmentId"].ToString());
        cmd.Parameters.Add("@TestZegara", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropTestZegara.SelectedValues);
        cmd.Parameters.Add("@MMSE", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textMMSE.Text);
        cmd.Parameters.Add("@WAIS_R_Wiadomosci", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textWAISR_Wiadomosci.Text);
        cmd.Parameters.Add("@WAIS_R_PowtarzanieCyfr", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textWAISR_PowtarzanieCyfr.Text);
        cmd.Parameters.Add("@SkalaDepresjiBecka", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textSkalaDepresjiBecka.Text);
        cmd.Parameters.Add("@TestFluencjiZwierzeta", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textTestFluencjiZwierzeta.Text);
        cmd.Parameters.Add("@TestFluencjiOstre", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textTestFluencjiOstre.Text);
        cmd.Parameters.Add("@TestFluencjik", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textTestFluencjik.Text);
        cmd.Parameters.Add("@TestLaczeniaPunktowa", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textTestLaczeniaPunktowaA.Text);
        cmd.Parameters.Add("@TestLaczeniaPunktowB", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textTestLaczeniaPunktowB.Text);
        cmd.Parameters.Add("@TestUczeniaSlownoSluchowego", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textTestUczeniaSlownoSluchowego.Text);
        cmd.Parameters.Add("@TestStroopa", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textTestStroopa.Text);
        cmd.Parameters.Add("@TestMinnesota", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNull(textTestMinnesota.Text);
        cmd.Parameters.Add("@InnePsychologiczne", SqlDbType.VarChar, 150).Value =
DatabaseProcedures.getStringOrNull(textInnePsychologiczne.Text);
        cmd.Parameters.Add("@OpisBadania", SqlDbType.VarChar, 2000).Value =
DatabaseProcedures.getStringOrNull(textOpisBadania.Text);
        cmd.Parameters.Add("@Wnioski", SqlDbType.VarChar, 2000).Value =
DatabaseProcedures.getStringOrNull(textWnioski.Text);
        cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
        cmd.Parameters.Add("@result", SqlDbType.Int);
        cmd.Parameters["@result"].Direction = ParameterDirection.Output;
        cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
        cmd.Parameters["@message"].Direction = ParameterDirection.Output;
        cmd.Connection = con;

        int success = 0;
        try
        {
            con.Open();
            cmd.ExecuteNonQuery();

```

```

        success = (int)cmd.Parameters["@result"].Value;

        if (success == 0)
        {
            Response.Redirect("~/AppointmentForm.aspx");
        }
        else
        {
            labelMessage.Text = (string)cmd.Parameters["@message"].Value;
        }
    }
    catch (SqlException ex)
    {
        labelMessage.Text = ex.Message;
    }
    finally
    {
        cmd.Dispose();
        if (con != null)
        {
            con.Close();
        }
    }
}

private void loadPartG()
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select TestZegara, MMSE, WAIS_R_Wiadomosci, WAIS_R_PowtarzanieCyfr,
SkalaDepresjiBecka, TestFluencjiZwierzeta, TestFluencjiOstre, " +
        "TestFluencjiK, TestLaczeniaPunktowA, TestLaczeniaPunktowB,
TestUczeniaSlownoSluchowego, TestStroopa, TestMinnesota, InnePsychologiczne, " +
        "OpisBadania, Wnioski " +
        "from Wizyta where IdWizyta = " + Session["AppointmentId"];
    cmd.Connection = con;

    try
    {
        con.Open();
        SqlDataReader rdr = cmd.ExecuteReader();
        while (rdr.Read())
        {
            dropTestZegara.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["TestZegara"]);
            textMMSE.Text = DatabaseProcedures.getTextByteValue(rdr["MMSE"]);
            textWAISR_Wiadomosci.Text =
DatabaseProcedures.getTextByteValue(rdr["WAIS_R_Wiadomosci"]);
            textWAISR_PowtarzanieCyfr.Text =
DatabaseProcedures.getTextByteValue(rdr["WAIS_R_PowtarzanieCyfr"]);
            textSkalaDepresjiBecka.Text =
DatabaseProcedures.getTextByteValue(rdr["SkalaDepresjiBecka"]);
            textTestFluencjiZwierzeta.Text =
DatabaseProcedures.getTextByteValue(rdr["TestFluencjiZwierzeta"]);
            textTestFluencjiOstre.Text =
DatabaseProcedures.getTextByteValue(rdr["TestFluencjiOstre"]);
            textTestFluencjiK.Text =
DatabaseProcedures.getTextByteValue(rdr["TestFluencjiK"]);
            textTestLaczeniaPunktowA.Text =
DatabaseProcedures.getTextByteValue(rdr["TestLaczeniaPunktowA"]);
            textTestLaczeniaPunktowB.Text =
DatabaseProcedures.getTextByteValue(rdr["TestLaczeniaPunktowB"]);
            textTestUczeniaSlownoSluchowego.Text =
DatabaseProcedures.getTextByteValue(rdr["TestUczeniaSlownoSluchowego"]);
            textTestStroopa.Text = DatabaseProcedures.getTextByteValue(rdr["TestStroopa"]);
            textTestMinnesota.Text =
DatabaseProcedures.getTextByteValue(rdr["TestMinnesota"]);

```

```

        textInnePsychologiczne.Text =
DatabaseProcedures.getTextStringValue(rdr["InnePsychologiczne"]);
        textOpisBadania.Text = DatabaseProcedures.getTextStringValue(rdr["OpisBadania"]);
        textWnioski.Text = DatabaseProcedures.getTextStringValue(rdr["Wnioski"]);
    }
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}
}

protected void buttonOK_Click(object sender, EventArgs e)
{
    savePartG();
}
protected void buttonCancel_Click(object sender, EventArgs e)
{
    Response.Redirect("~/AppointmentForm.aspx");
}
}

```

PartHForm.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="PartHForm.aspx.cs" Inherits="PartHForm" %>

```

```

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <h2>Badania dodatkowe</h2>
    <br />
    <asp:Label ID="labelAppointment" runat="server" Text=""></asp:Label>
    <table ID="Table1" runat="server">
        <tr>
            <td>Holter:</td>
            <td>
                <asp:DropDownList ID="dropHolter" runat="server">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>
                    <asp:ListItem Value="1">tak</asp:ListItem>
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Badanie węchu:</td>
            <td>
                <asp:DropDownList ID="dropBadanieWechu" runat="server">
                    <asp:ListItem Value="2" Text=""></asp:ListItem>
                    <asp:ListItem Value="0">nie</asp:ListItem>
                    <asp:ListItem Value="1">tak</asp:ListItem>
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Wynik węchu:</td>
            <td>
                <asp:DropDownList ID="dropWynikWechu" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
    </table>

```

```

        <td>Limit dysfagii:</td>
        <td>
            <asp:DropDownList ID="dropLimitDysfagii" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>pH-metria przełyku:</td>
        <td>
            <asp:DropDownList ID="dropPHmetriaPrzełyku" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>SPECT:</td>
        <td>
            <asp:DropDownList ID="dropSPECT" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>SPECT wynik:</td>
        <td>
            <asp:CheckBoxList ID="checkListSPECTWynik" runat="server">
            </asp:CheckBoxList>
        </td>
    </tr>
    <tr>
        <td>MRI:</td>
        <td>
            <asp:DropDownList ID="dropMRI" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>MRI wynik:</td>
        <td>
            <asp:TextBox ID="textMRIwynik" runat="server" Rows="5"
TextMode="Multiline"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>USG śródmózgowia:</td>
        <td>
            <asp:DropDownList ID="dropUSGsrodmozgowia" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>USG wynik:</td>
        <td>
            <asp:DropDownList ID="dropUSGWynik" runat="server">
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Genetyka:</td>

```



```

        <td>
            <asp:DropDownList ID="dropGenetyka" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Genetyka wynik:</td>
        <td>
            <asp:TextBox ID="textGenetykaWynik" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>Surowica:</td>
        <td>
            <asp:DropDownList ID="dropSurowica" runat="server">
                <asp:ListItem Value="2" Text=""></asp:ListItem>
                <asp:ListItem Value="0">nie</asp:ListItem>
                <asp:ListItem Value="1">tak</asp:ListItem>
            </asp:DropDownList>
        </td>
    </tr>
    <tr>
        <td>Surowica pozostało:</td>
        <td>
            <asp:TextBox ID="textSurowicaPozostało" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>Ferrytyna:</td>
        <td>
            <asp:TextBox ID="textFerrytyna" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>CRP:</td>
        <td>
            <asp:TextBox ID="textCRP" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>NTproCNP:</td>
        <td>
            <asp:TextBox ID="textNTproCNP" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>URCA:</td>
        <td>
            <asp:TextBox ID="textURCA" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>WitD:</td>
        <td>
            <asp:TextBox ID="textWitD" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>CHOL:</td>
        <td>
            <asp:TextBox ID="textCHOL" runat="server"></asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>TGI:</td>
        <td>

```

```

        <asp:TextBox ID="textTGI" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>HDL:</td>
    <td>
        <asp:TextBox ID="textHDL" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>LDL:</td>
    <td>
        <asp:TextBox ID="textLDL" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>olLDL:</td>
    <td>
        <asp:TextBox ID="textolLDL" runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>Laboratoryjne inne:</td>
    <td>
        <asp:TextBox ID="textLaboratoryjneInne" runat="server" Rows="5"
TextMode="MultiLine"></asp:TextBox>
    </td>
</tr>
</table>
<asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
<br />
<asp:Button ID="buttonOK" runat="server" Text="Zatwierdź"
onclick="buttonOK_Click" />
<asp:Button ID="buttonCancel" runat="server" Text="Anuluj"
CausesValidation="False" onclick="buttonCancel_Click" />
</asp:Content>

```

PartHForm.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

public partial class PartHForm : System.Web.UI.Page
{
    private static byte NO_DATA = 100;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["PatientNumber"] != null && Session["AppointmentId"] != null &&
Session["AppointmentName"] != null)
        {
            labelAppointment.Text = "Pacjent: " + Session["PatientNumber"] + "<br />Wizyta: " +
Session["AppointmentName"];
            if (!IsPostBack)
            {
                initControls();
                loadPartH();
            }
        }
        else
        {
            Response.Redirect("~/Main.aspx");
        }
    }
}

```

```

    }

    private void initControls()
    {
        dropWynikWechu.DataSource = DatabaseProcedures.getEnumerationByteWithNoData("Wizyta",
"WynikWechu", NO_DATA);
        dropWynikWechu.DataTextField = "Value";
        dropWynikWechu.DataValueField = "Key";
        dropWynikWechu.DataBind();

        dropLimitDysfagii.DataSource = DatabaseProcedures.getEnumerationByteWithNoData("Wizyta",
"LimitDysfagii", NO_DATA);
        dropLimitDysfagii.DataTextField = "Value";
        dropLimitDysfagii.DataValueField = "Key";
        dropLimitDysfagii.DataBind();

        checkListSPECTWynik.DataSource = DatabaseProcedures.getEnumerationByte("Wizyta",
"SPECTWynik");
        checkListSPECTWynik.DataTextField = "Value";
        checkListSPECTWynik.DataValueField = "Key";
        checkListSPECTWynik.DataBind();

        dropUSGWynik.DataSource = DatabaseProcedures.getEnumerationByteWithNoData("Wizyta",
"USGWynik", NO_DATA);
        dropUSGWynik.DataTextField = "Value";
        dropUSGWynik.DataValueField = "Key";
        dropUSGWynik.DataBind();

        Utils.setSelectedCheckListItems(DatabaseProcedures.getMultiChoice("SPECTWynik",
int.Parse(Session["AppointmentId"].ToString()))), checkListSPECTWynik);
    }

    private void saveParthH()
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[update_examination_questionnaire_parthH]";
        cmd.Parameters.Add("@IdWizyta", SqlDbType.Int).Value =
int.Parse(Session["AppointmentId"].ToString());
        cmd.Parameters.Add("@Holter", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropHolter.SelectedValue);
        cmd.Parameters.Add("@BadanieWechu", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropBadanieWechu.SelectedValue);
        cmd.Parameters.Add("@WynikWechu", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNullWithNoData(dropWynikWechu.SelectedValue, NO_DATA.ToString());
        cmd.Parameters.Add("@LimitDysfagii", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNullWithNoData(dropLimitDysfagii.SelectedValue, NO_DATA.ToString());
        cmd.Parameters.Add("@pH_metriaPrzeżyku", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropHmetriaPrzeżyku.SelectedValue);
        cmd.Parameters.Add("@SPECT", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropSPECT.SelectedValue);
        cmd.Parameters.Add("@MRI", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropMRI.SelectedValue);
        cmd.Parameters.Add("@MRIwynik", SqlDbType.VarChar, 2000).Value =
DatabaseProcedures.getStringOrNull(textMRIwynik.Text);
        cmd.Parameters.Add("@USGsrodmozgowia", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropUSGsrodmozgowia.SelectedValue);
        cmd.Parameters.Add("@USGWynik", SqlDbType.TinyInt).Value =
DatabaseProcedures.getByteOrNullWithNoData(dropUSGWynik.SelectedValue, NO_DATA.ToString());
        cmd.Parameters.Add("@Genetyka", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropGenetyka.SelectedValue);
        cmd.Parameters.Add("@GenetykaWynik", SqlDbType.VarChar, 50).Value =
DatabaseProcedures.getStringOrNull(textGenetykaWynik.Text);
        cmd.Parameters.Add("@Surowica", SqlDbType.Bit).Value =
DatabaseProcedures.getBitOrNull(dropSurowica.SelectedValue);
        cmd.Parameters.Add("@SurowicaPozostało", SqlDbType.VarChar, 50).Value =
DatabaseProcedures.getStringOrNull(textSurowicaPozostało.Text);
    }

```

```

cmd.Parameters.Add(saveDecimal("@Ferrytyna", textFerrytyna));
cmd.Parameters.Add(saveDecimal("@CRP", textCRP));
cmd.Parameters.Add(saveDecimal("@NTproCNP", textNTproCNP));
cmd.Parameters.Add(saveDecimal("@URCA", textURCA));
cmd.Parameters.Add(saveDecimal("@WitD", textWitD));
cmd.Parameters.Add(saveDecimal("@CHOL", textCHOL));
cmd.Parameters.Add(saveDecimal("@TGI", textTGI));
cmd.Parameters.Add(saveDecimal("@HDL", textFerrytyna));
cmd.Parameters.Add(saveDecimal("@LDL", textLDL));
cmd.Parameters.Add(saveDecimal("@oLDL", textoLDL));
cmd.Parameters.Add("@LaboratoryjneInne", SqlDbType.VarChar, 1000).Value =
DatabaseProcedures.getStringOrNull(textLaboratoryjneInne.Text);
cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
cmd.Parameters.Add("@result", SqlDbType.Int);
cmd.Parameters["@result"].Direction = ParameterDirection.Output;
cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
cmd.Parameters["@message"].Direction = ParameterDirection.Output;
cmd.Connection = con;

int success = 0;
try
{
    con.Open();
    cmd.ExecuteNonQuery();
    success = (int)cmd.Parameters["@result"].Value;

    if (success == 0)
    {
        Response.Redirect("~/AppointmentForm.aspx");
    }
    else
    {
        labelMessage.Text = (string)cmd.Parameters["@message"].Value;
    }
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}
}

private SqlParameter saveDecimal(string parameter, TextBox textBox)
{
    SqlParameter decimalParameter = new SqlParameter(parameter, SqlDbType.Decimal);
    decimalParameter.Precision = 7;
    decimalParameter.Scale = 3;
    decimalParameter.Value = DatabaseProcedures.getDecimalOrNull(textBox.Text);

    return decimalParameter;
}

private void loadPartH()
{
    SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
    SqlCommand cmd = new SqlCommand();
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select Holter, BadanieWechu, WynikWechu, LimitDysfagii,
pH_metriaPrzełyku, SPECT, " +
        "MRI, MRIwynik, USGsrodmozgowia, USGWynik, Genetyka, GenetykaWynik, Surowica,
SurowicaPozostało, " +

```

```

        "Ferrytyna, CRP, NTproCNP, URCA, WitD, CHOL, TGI, HDL, LDL, oLDL, LaboratoryjneInne "
+
        "from Wizyta where IdWizyta = " + Session["AppointmentId"];
cmd.Connection = con;

try
{
    con.Open();
    SqlDataReader rdr = cmd.ExecuteReader();
    while (rdr.Read())
    {
        dropHolter.SelectedValue = DatabaseProcedures.getDropBitValue(rdr["Holter"]);
        dropBadanieWechu.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["BadanieWechu"]);
        dropWynikWechu.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["WynikWechu"], NO_DATA.ToString());
        dropLimitDysfagii.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["LimitDysfagii"], NO_DATA.ToString());
        dropHmetriaPrzełyku.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["pH_metriaPrzełyku"]);
        dropSPECT.SelectedValue = DatabaseProcedures.getDropBitValue(rdr["SPECT"]);
        dropMRI.SelectedValue = DatabaseProcedures.getDropBitValue(rdr["MRI"]);
        textMRIwynik.Text = DatabaseProcedures.getTextStringValue(rdr["MRIwynik"]);
        dropUSGsrodmozgowia.SelectedValue =
DatabaseProcedures.getDropBitValue(rdr["USGsrodmozgowia"]);
        dropUSGWynik.SelectedValue =
DatabaseProcedures.getDropMultiValueWithNoData(rdr["USGWynik"], NO_DATA.ToString());
        dropGenetyka.SelectedValue = DatabaseProcedures.getDropBitValue(rdr["Genetyka"]);
        textGenetykawynik.Text =
DatabaseProcedures.getTextStringValue(rdr["Genetykawynik"]);
        dropSurowica.SelectedValue = DatabaseProcedures.getDropBitValue(rdr["Surowica"]);
        textSurowicaPozostało.Text =
DatabaseProcedures.getTextStringValue(rdr["SurowicaPozostało"]);
        textFerrytyna.Text = DatabaseProcedures.getTextDecimalValue(rdr["Ferrytyna"]);
        textCRP.Text = DatabaseProcedures.getTextDecimalValue(rdr["CRP"]);
        textNTproCNP.Text = DatabaseProcedures.getTextDecimalValue(rdr["NTproCNP"]);
        textURCA.Text = DatabaseProcedures.getTextDecimalValue(rdr["URCA"]);
        textWitD.Text = DatabaseProcedures.getTextDecimalValue(rdr["WitD"]);
        textCHOL.Text = DatabaseProcedures.getTextDecimalValue(rdr["CHOL"]);
        textTGI.Text = DatabaseProcedures.getTextDecimalValue(rdr["TGI"]);
        textHDL.Text = DatabaseProcedures.getTextDecimalValue(rdr["HDL"]);
        textLDL.Text = DatabaseProcedures.getTextDecimalValue(rdr["LDL"]);
        textoLDL.Text = DatabaseProcedures.getTextDecimalValue(rdr["oLDL"]);
        textLaboratoryjneInne.Text =
DatabaseProcedures.getTextStringValue(rdr["LaboratoryjneInne"]);
    }
}
catch (SqlException ex)
{
    labelMessage.Text = ex.Message;
}
finally
{
    cmd.Dispose();
    if (con != null)
    {
        con.Close();
    }
}

protected void buttonOK_Click(object sender, EventArgs e)
{
    DatabaseProcedures.saveMultiChoice(Utils.getSelectedCheckListItems(checkListSPECTWynik),
"SPECTWynik", int.Parse(Session["AppointmentId"].ToString()), User.Identity.Name);
    saveParth();
}
protected void buttonCancel_Click(object sender, EventArgs e)
{

```

```

    }
    Response.Redirect("~/AppointmentForm.aspx");
}

```

PatientForm.aspx

```

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="PatientForm.aspx.cs" Inherits="PatientForm" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" Runat="Server">
    <table ID="Table1" runat="server">
        <tr>
            <td>Grupa:</td>
            <td>
                <asp:DropDownList ID="dropGroup" runat="server" AutoPostBack="True"
                    onselectedindexchanged="dropGroup_SelectedIndexChanged">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Numer pacjenta:</td>
            <td>
                <asp:TextBox ID="textPatientNumber" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
                    ErrorMessage="Proszę podać numer."
                    ControlToValidate="textPatientNumber"></asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td>Rok urodzenia:</td>
            <td>
                <asp:DropDownList ID="dropYear" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Miesiąc urodzenia:</td>
            <td>
                <asp:DropDownList ID="dropMonth" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Płeć:</td>
            <td>
                <asp:RadioButton ID="radioWoman" runat="server" Text="Kobieta"
                    GroupName="Sex" Checked="True" />
                <asp:RadioButton ID="radioMan" runat="server" Text="Mężczyzna"
                    GroupName="Sex" />
            </td>
        </tr>
        <tr>
            <td>Lokalizacja:</td>
            <td>
                <asp:DropDownList ID="dropLocation" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
        <tr>
            <td>Liczba elektrod:</td>
            <td>
                <asp:DropDownList ID="dropElectrodes" runat="server">
                </asp:DropDownList>
            </td>
        </tr>
    </table>
    <asp:Label ID="labelMessage" runat="server" Text=""></asp:Label>
<br />

```

```

        <asp:Button ID="buttonOK" runat="server" Text="Zatwierdź"
            onclick="buttonOK_Click" />
        <asp:Button ID="buttonCancel" runat="server" Text="Anuluj"
            CausesValidation="False" onclick="buttonCancel_Click" />
    </asp:Content>

```

PatientForm.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Web.Configuration;
using System.Data;

public partial class PatientForm : System.Web.UI.Page
{
    private bool update = false;

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            dropGroup.DataSource = DatabaseProcedures.getEnumerationString("Pacjent",
                "NazwaGrupy");
            dropGroup.DataTextField = "Value";
            dropGroup.DataValueField = "Key";
            dropGroup.DataBind();

            int minYear = 1920;
            int currentYear = DateTime.Now.Year;
            for (int i = minYear; i <= currentYear; i++)
            {
                dropYear.Items.Add(new ListItem("" + i, "" + i));
            }
            dropYear.SelectedIndex = currentYear - minYear - 50;

            for (int i = 1; i <= 12; i++)
            {
                dropMonth.Items.Add(new ListItem("" + i, "" + i));
            }

            dropLocation.DataSource = DatabaseProcedures.getEnumerationString("Pacjent",
                "Lokalizacja");
            dropLocation.DataTextField = "Value";
            dropLocation.DataValueField = "Key";
            dropLocation.DataBind();

            for (int i = 0; i <= 10; i++)
            {
                dropElectrodes.Items.Add(new ListItem("" + i, "" + i));
            }
            dropElectrodes.SelectedIndex = 2;
        }

        if (Session["PatientNumber"] != null)
        {
            String patientNumber = Session["PatientNumber"].ToString();
            update = true;
            if (!IsPostBack)
            {
                textPatientNumber.Text = patientNumber;
                textPatientNumber.Enabled = false;
                dropGroup.Enabled = false;
                loadPatient(patientNumber);
            }
        }
    }
}

```

```

        else
        {
            if (!IsPostBack)
            {
                textPatientNumber.Text = suggestNewPatientNumber(dropGroup.SelectedValue);
            }
        }
    }

    protected void buttonCancel_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/Main.aspx");
    }

    protected void buttonOK_Click(object sender, EventArgs e)
    {
        int sex = 0;
        if (radioMan.Checked)
        {
            sex = 1;
        }

        savePatient(textPatientNumber.Text, dropGroup.SelectedValue,
int.Parse(dropYear.SelectedValue), int.Parse(dropMonth.SelectedValue), sex,
        dropLocation.SelectedValue, int.Parse(dropElectrodes.SelectedValue), update);
    }

    private void savePatient(string number, string group, int birthYear, int birthMonth, int sex,
        string location, int electrodes, bool update)
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[update_patient_1]";
        cmd.Parameters.Add("@NumerPacjenta", SqlDbType.VarChar, 20).Value = number;
        cmd.Parameters.Add("@NazwaGrupy", SqlDbType.VarChar, 3).Value = group;
        cmd.Parameters.Add("@RokUrodzenia", SqlDbType.SmallInt).Value = (short)birthYear;
        cmd.Parameters.Add("@MiesiacUrodzenia", SqlDbType.TinyInt).Value = (byte)birthMonth;
        cmd.Parameters.Add("@Plec", SqlDbType.TinyInt).Value = sex;
        cmd.Parameters.Add("@Lokalizacja", SqlDbType.VarChar, 10).Value = location;
        cmd.Parameters.Add("@LiczbaElektrod", SqlDbType.TinyInt).Value = (byte)electrodes;
        cmd.Parameters.Add("@allow_update_existing", SqlDbType.Bit).Value = update;
        cmd.Parameters.Add("@actor_login", SqlDbType.VarChar, 50).Value = User.Identity.Name;
        cmd.Parameters.Add("@result", SqlDbType.Int);
        cmd.Parameters["@result"].Direction = ParameterDirection.Output;
        cmd.Parameters.Add("@message", SqlDbType.VarChar, 200);
        cmd.Parameters["@message"].Direction = ParameterDirection.Output;
        cmd.Connection = con;

        int success = 0;
        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
            success = (int) cmd.Parameters["@result"].Value;
            if (success == 2)
            {
                labelMessage.Text = (string) cmd.Parameters["@message"].Value;
            }
            else
            {
                Response.Redirect("~/Main.aspx");
            }
        }
        catch (SqlException ex)
        {
            labelMessage.Text = ex.Message;
        }
    }

```



```

        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }
    }

    private void loadPatient(string patientNumber)
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "select NazwaGrupy, RokUrodzenia, MiesiacUrodzenia, Plec, Lokalizacja,
LiczbaElektrod from Pacjent where NumerPacjenta = '" + patientNumber + "'";
        cmd.Connection = con;

        try
        {
            con.Open();
            SqlDataReader rdr = cmd.ExecuteReader();
            while (rdr.Read())
            {
                dropGroup.SelectedValue = rdr["NazwaGrupy"].ToString();
                dropYear.SelectedValue = ((short)rdr["RokUrodzenia"]).ToString();
                dropMonth.SelectedValue = ((byte)rdr["MiesiacUrodzenia"]).ToString();
                int sex = ((byte)rdr["Plec"]); ;
                if (sex == 0)
                {
                    radiowoman.Checked = true;
                }
                else
                {
                    radioMan.Checked = true;
                }
                dropLocation.SelectedValue = (string)rdr["Lokalizacja"];
                dropElectrodes.SelectedValue = ((byte)rdr["LiczbaElektrod"]).ToString();
            }
        }
        catch (SqlException ex)
        {
            labelMessage.Text = ex.Message;
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }
    }

    private string suggestNewPatientNumber(string group)
    {
        SqlConnection con = new
SqlConnection(WebConfigurationManager.ConnectionStrings[DatabaseProcedures.SERVER].ToString());
        SqlCommand cmd = new SqlCommand();
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.CommandText = "[dbo].[suggest_new_patient_number_with_group]";
        cmd.Parameters.Add("@admission_date", SqlDbType.Date).Value = DateTime.Now.ToString("yyyy-
MM-dd");
        cmd.Parameters.Add("@patient_group", SqlDbType.VarChar, 3).Value = group;
        cmd.Parameters.Add("@patient_number", SqlDbType.VarChar, 20);
        cmd.Parameters["@patient_number"].Direction = ParameterDirection.Output;
        cmd.Connection = con;
    }

```

```

        string patientNumber = "";
        try
        {
            con.Open();
            cmd.ExecuteNonQuery();
            patientNumber =
DatabaseProcedures.getTextStringValue(cmd.Parameters["@patient_number"].Value);
        }
        catch (SqlException ex)
        {
            labelMessage.Text = ex.Message;
        }
        finally
        {
            cmd.Dispose();
            if (con != null)
            {
                con.Close();
            }
        }

        return patientNumber;
    }

    protected void dropGroup_SelectedIndexChanged(object sender, EventArgs e)
    {
        textPatientNumber.Text = suggestNewPatientNumber(dropGroup.SelectedValue);
    }
}

Site.master
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Site.master.cs" Inherits="SiteMaster" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head runat="server">
    <title></title>
    <link href="~/Styles/Site.css" rel="stylesheet" type="text/css" />
    <asp:ContentPlaceHolder ID="HeadContent" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form runat="server">
        <div class="page">
            <div class="header">
                <div class="title">
                    <h1>
                        TPP
                    </h1>
                </div>
                <div class="loginDisplay">
                    <asp:LoginView ID="HeadLoginView" runat="server" EnableViewState="false">
                        <AnonymousTemplate>
                            [ <a href="~/Login.aspx" ID="HeadLoginStatus" runat="server">Zaloguj
się</a> ]
                        </AnonymousTemplate>
                        <LoggedInTemplate>
                            Zalogowany jako: <span class="bold"><asp:LoginName ID="HeadLoginName"
runat="server" /></span>
                            [ <asp:LoginStatus ID="HeadLoginStatus" runat="server"
LogoutAction="Redirect" LogoutText="Wyloguj" LogoutPageUrl="~/"/> ]
                        </LoggedInTemplate>
                    </asp:LoginView>
                </div>
                <div class="clear hideSkiplink">
                    <asp:Menu ID="NavigationMenu" runat="server" CssClass="menu"
EnableViewState="false" IncludeStyleBlock="false" Orientation="Horizontal">

```

```

        <Items>
            <asp:MenuItem NavigateUrl="~/Default.aspx" Text="Strona powitalna"/>
            <asp:MenuItem NavigateUrl="~/About.aspx" Text="0 stronie"/>
            <%--
            <asp:MenuItem NavigateUrl="~/Login.aspx" Text="Zaloguj"/>
            --%>
        </Items>
    </asp:Menu>
</div>
</div>
<div class="main">
    <asp:ContentPlaceHolder ID="MainContent" runat="server"/>
</div>
<div class="clear">
</div>
</div>
<div class="footer">

</div>
</form>
</body>
</html>

Site.master.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class SiteMaster : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}

Web.config
<?xml version="1.0"?>
<!--
    For more information on how to configure your ASP.NET application, please visit
    http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
    <connectionStrings>
        <add name="ApplicationServices" connectionString="data
source=.\SQLEXPRESS;Integrated Security=SSPI;AttachDBFilename=|DataDirectory|\aspnetdb.mdf;User
Instance=true" providerName="System.Data.SqlClient"/>
        <add name="TPPServer" connectionString="data source=localhost;Database=TPP;Integrated
Security=SSPI" providerName="System.Data.SqlClient"/>
    </connectionStrings>
    <system.web>
        <sessionState timeout="240"></sessionState>
    </system.web>
    <location path="Styles">
        <system.web>
            <authorization>
                <allow users="*" />
            </authorization>
        </system.web>
    </location>
    <system.web>
        <compilation debug="true" targetFramework="4.0"/>
        <authentication mode="Forms">
            <forms loginUrl="~/Login.aspx" protection="All" timeout="2880"/>
        </authentication>
        <authorization>

```

```

        <deny users ="?" />
        <allow users = "*" />
    </authorization>
    <membership>
        <providers>
            <clear/>
            <add name="AspNetSqlMembershipProvider"
type="System.Web.Security.SqlMembershipProvider" connectionStringName="ApplicationServices"
enablePasswordRetrieval="false" enablePasswordReset="true" requiresQuestionAndAnswer="false"
requiresUniqueEmail="false" maxInvalidPasswordAttempts="5" minRequiredPasswordLength="6"
minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10" applicationName="/" />
        </providers>
    </membership>
    <profile>
        <providers>
            <clear/>
            <add name="AspNetSqlProfileProvider"
type="System.Web.Profile.SqlProfileProvider" connectionStringName="ApplicationServices"
applicationName="/" />
        </providers>
    </profile>
    <roleManager enabled="false">
        <providers>
            <clear/>
            <add name="AspNetSqlRoleProvider"
type="System.Web.Security.SqlRoleProvider" connectionStringName="ApplicationServices"
applicationName="/" />
            <add name="AspNetWindowsTokenRoleProvider"
type="System.Web.Security.WindowsTokenRoleProvider" applicationName="/" />
        </providers>
    </roleManager>
</system.web>
<system.webServer>
    <modules runAllManagedModulesForAllRequests="true" />
</system.webServer>
</configuration>

```