```
using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using System.Windows.Threading;
using System.Threading;
using System.Text.RegularExpressions;
using System.Data.SqlClient;
using System.Diagnostics;
using System.Data;
//*
//COPYRIGHT Weingartner 2017
//*
namespace FEC
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        int te = 20;
        private static string getconnected()
        {
            //create connection string
            SqlConnectionStringBuilder b = new SqlConnectionStringBuilder();
            b.DataSource = "mssql3.gear.host";
            b.InitialCatalog = "phoenix";
            b.IntegratedSecurity = false;
            b.UserID = "phoenix";
            b.Password = "@CorpNetWeRSecure!&";
            b.PersistSecurityInfo = true;
            b.TrustServerCertificate = true;
            b.ConnectTimeout = 200;
            string connectionString = b.ConnectionString;
            return connectionString;
        }
        private void MainWindow_MouseDown(object sender, MouseButtonEventArgs e)
        {
            if (e.LeftButton == MouseButtonState.Pressed)
            {
                DragMove();
            }
        }
        readonly Regex alpha = new Regex(@"^[a-zA-Z]");
        readonly Regex beta = new Regex(@"^[0-9]");
        private  void reset()
        {
            //reset everything to default
            blurFront.Opacity = 0;
            blurFront.Visibility = Visibility.Collapsed;
            New_Grid.Visibility = Visibility.Collapsed;
            New_Grid.Margin = new Thickness(0, 0, 0, 0);
            New_Grid.Opacity = 0;

            Edit_Grid.Visibility = Visibility.Collapsed;
            Edit_Grid.Margin = new Thickness(0, 0, 0, 0);
            Edit_Grid.Opacity = 0;

            View_Grid.Visibility = Visibility.Collapsed;
            View_Grid.Margin = new Thickness(0, 0, 0, 0);
            View_Grid.Opacity = 0;

            Help_Grid.Visibility = Visibility.Collapsed;
            Help_Grid.Margin = new Thickness(0, -40, 0, 0);
            Help_Grid.Opacity = 0;



        }
        private void reset2()
        {

            New_Customer_Grid.Visibility = Visibility.Collapsed;
            New_Customer_Grid.Margin = new Thickness(0, 0, 0, 0);
            New_Customer_Grid.Opacity = 0;

            New_Employee_Grid.Visibility = Visibility.Collapsed;
            New_Employee_Grid.Margin = new Thickness(0, 0, 0, 0);
            New_Employee_Grid.Opacity = 0;

            Edit_Customer_Grid.Visibility = Visibility.Collapsed;
            Edit_Customer_Grid.Margin = new Thickness(0, 0, 0, 0);
            Edit_Customer_Grid.Opacity = 0;

            Edit_Employee_Grid.Visibility = Visibility.Collapsed;
            Edit_Employee_Grid.Margin = new Thickness(0, 0, 0, 0);
            Edit_Employee_Grid.Opacity = 0;



        }
        private void loaded(object sender, RoutedEventArgs e)
        {


            reset();
            reset2();
        }
        private void quit_MouseClick(object sender, RoutedEventArgs e)
        {
            this.Close();
        }

        private void minimize_MouseClick(object sender, RoutedEventArgs e)
        {
            this.WindowState = WindowState.Minimized;
        }


        private void new_back(object sender, RoutedEventArgs e)
        {


            //hide new pane
            DispatcherTimer hide_new = new DispatcherTimer();
            hide_new.Tick += hide_new_Tick;
            hide_new.Interval = new TimeSpan(0, 0, 0, 0, 15);
            hide_new.Start();
            //fadeout homeblur
            DispatcherTimer fadeout_home = new DispatcherTimer();
            fadeout_home.Tick += fadeout_home_Tick;
            fadeout_home.Interval = new TimeSpan(0, 0, 0, 0, 15);
            fadeout_home.Start();

        }
        double blurOpa = 0;
        private void fadein_home_Tick(object sender, EventArgs e)
        {
            if (blurOpa < 1)
            {
                blurOpa += .04;
                blurFront.Opacity = blurOpa;
            }
            else
            {
                blurFront.Opacity = 1;
                blurOpa = 1;
                (sender as DispatcherTimer).Stop();
            }
        }

        private void fadeout_home_Tick(object sender, EventArgs e)
        {
            if (blurOpa > 0)
            {
                blurOpa -= .04;
                blurFront.Opacity = blurOpa;
            }
            else
            {
                //blurOpa = 0;
                blurFront.Opacity = 0;
                blurFront.Visibility = Visibility.Collapsed;

                (sender as DispatcherTimer).Stop();
            }
        }

        {
            blurFront.Visibility = Visibility.Visible;
            //init
            New_Grid.Margin = new Thickness(0, te, 0, -te);
            New_Grid.Opacity = 0;
            New_Grid.Visibility = Visibility.Visible;
            //fadein homeblur
            DispatcherTimer fadein_home = new DispatcherTimer();
            fadein_home.Tick += fadein_home_Tick;
            fadein_home.Interval = new TimeSpan(0, 0, 0, 0, 15);
            fadein_home.Start();

            DispatcherTimer show_new = new DispatcherTimer();
            show_new.Tick += show_new_Tick;
            show_new.Interval = new TimeSpan(0, 0, 0, 0, 15);
            show_new.Start();
        }
        int new_h = 20;
        double new_o = 0;
        private void show_new_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (new_o < .9)
            {
                new_o += .05;
                New_Grid.Opacity = new_o;
            }
            else
            {
                New_Grid.Opacity = 1;
                (sender as DispatcherTimer).Stop();
            }
            //change position
            if(new_h > 0)
            {
                new_h--;
                New_Grid.Margin = new Thickness(0, new_h, 0, 0);
            }
            else
            {

            }

        }

        private void hide_new_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (new_o > 0)
            {
                new_o -= .05;
                New_Grid.Opacity = new_o;
            }
            //change position
            if (new_h < te)
```

```
                new_h++;
                New_Grid.Margin = new Thickness(0, new_h, 0, 0);
            }
            else
            {
                //New_Grid.Visibility = Visibility.Collapsed;
                reset();
                (sender as DispatcherTimer).Stop();
            }
        }


        private void edit_back(object sender, RoutedEventArgs e)
        {


            //hide edit pane
            DispatcherTimer hide_edit = new DispatcherTimer();
            hide_edit.Tick += hide_edit_Tick;
            hide_edit.Interval = new TimeSpan(0, 0, 0, 0, 15);
            hide_edit.Start();
            //fadeout homeblur
            DispatcherTimer fadeout_home = new DispatcherTimer();
            fadeout_home.Tick += fadeout_home_Tick;
            fadeout_home.Interval = new TimeSpan(0, 0, 0, 0, 15);
            fadeout_home.Start();

        }


        private void edit_MouseClick(object sender, RoutedEventArgs e)
        {
            blurFront.Visibility = Visibility.Visible;
            //init
            Edit_Grid.Margin = new Thickness(0, te, 0, -te);
            Edit_Grid.Opacity = 0;
            Edit_Grid.Visibility = Visibility.Visible;
            //fadein homeblur
            DispatcherTimer fadein_home = new DispatcherTimer();
            fadein_home.Tick += fadein_home_Tick;
            fadein_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
            fadein_home.Start();

            DispatcherTimer show_edit = new DispatcherTimer();
            show_edit.Tick += show_edit_Tick;
            show_edit.Interval = new TimeSpan(0, 0, 0, 0, 10);
            show_edit.Start();
        }
        int edit_h = 20;
        double edit_o = 0;
        private void show_edit_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (edit_o < .9)
            {
                edit_o += .05;
                Edit_Grid.Opacity = edit_o;
            }
            else
            {

                Edit_Grid.Opacity = 1;
                (sender as DispatcherTimer).Stop();
            }
            //change position
            if (edit_h > 0)
            {
                edit_h--;
                Edit_Grid.Margin = new Thickness(0, edit_h, 0, 0);
            }
            else
            {

            }

        }

        private void hide_edit_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (edit_o > 0)
            {
                edit_o -= .05;
                Edit_Grid.Opacity = edit_o;
            }
            //change position
            if (edit_h < te)
            {
                edit_h++;
                Edit_Grid.Margin = new Thickness(0, edit_h, 0, 0);
            }
            else
            {
                //edit_Grid.Visibility = Visibility.Collapsed;
                reset();
                (sender as DispatcherTimer).Stop();
            }
        }

        private void view_back(object sender, RoutedEventArgs e)
        {


            //hide view pane
            DispatcherTimer hide_view = new DispatcherTimer();
            hide_view.Tick += hide_view_Tick;
            hide_view.Interval = new TimeSpan(0, 0, 0, 0, 10);
            hide_view.Start();
            //fadeout homeblur
            DispatcherTimer fadeout_home = new DispatcherTimer();
            fadeout_home.Tick += fadeout_home_Tick;
            fadeout_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
            fadeout_home.Start();

        }


        private void view_MouseClick(object sender, RoutedEventArgs e)
        {
            blurFront.Visibility = Visibility.Visible;
            //init
            View_Grid.Margin = new Thickness(0, te, 0, -te);
            View_Grid.Opacity = 0;
            View_Grid.Visibility = Visibility.Visible;
            //fadein homeblur
            DispatcherTimer fadein_home = new DispatcherTimer();
            fadein_home.Tick += fadein_home_Tick;
            fadein_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
            fadein_home.Start();

            DispatcherTimer show_view = new DispatcherTimer();
            show_view.Tick += show_view_Tick;
            show_view.Interval = new TimeSpan(0, 0, 0, 0, 10);
            show_view.Start();
        }
        int view_h = 20;
        double view_o = 0;
        private void show_view_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (view_o < .9)
            {
                view_o += .05;
                View_Grid.Opacity = view_o;
            }
            else
            {
                View_Grid.Opacity = 1;
                (sender as DispatcherTimer).Stop();
            }
            //change position
            if (view_h > 0)
            {
                view_h--;
                View_Grid.Margin = new Thickness(0, view_h, 0, 0);
            }
            else
            {

            }

        }

        private void hide_view_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (view_o > 0)
            {
                view_o -= .05;
                View_Grid.Opacity = view_o;
            }

            //change position
            if (view_h < te)
            {
                view_h++;
                View_Grid.Margin = new Thickness(0, view_h, 0, 0);
            }
            else
            {
                //view_Grid.Visibility = Visibility.Collapsed;
                reset();
                (sender as DispatcherTimer).Stop();
            }
        }

        private void help_back(object sender, RoutedEventArgs e)
        {


            //hide help pane
            DispatcherTimer hide_help = new DispatcherTimer();
            hide_help.Tick += hide_help_Tick;
            hide_help.Interval = new TimeSpan(0, 0, 0, 0, 10);
            hide_help.Start();
            //fadeout homeblur
            DispatcherTimer fadeout_home = new DispatcherTimer();
            fadeout_home.Tick += fadeout_home_Tick;
            fadeout_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
            fadeout_home.Start();

        }


        private void help_MouseClick(object sender, RoutedEventArgs e)
        {
            blurFront.Visibility = Visibility.Visible;
            //init
            Help_Grid.Margin = new Thickness(0, te, 0, -te);
            Help_Grid.Opacity = 0;
            Help_Grid.Visibility = Visibility.Visible;
            //fadein homeblur
            DispatcherTimer fadein_home = new DispatcherTimer();
            fadein_home.Tick += fadein_home_Tick;
            fadein_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
            fadein_home.Start();

            DispatcherTimer show_help = new DispatcherTimer();
            show_help.Tick += show_help_Tick;
            show_help.Interval = new TimeSpan(0, 0, 0, 0, 10);
            show_help.Start();
        }
        int help_h = 20;
        double help_o = 0;
        private void show_help_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (help_o < .9)
            {
                help_o += .05;
                Help_Grid.Opacity = help_o;
            }
            else
            {
                Help_Grid.Opacity = 1;
                (sender as DispatcherTimer).Stop();
            }
            //change position
            if (help_h > 0)
            {
                help_h--;
                Help_Grid.Margin = new Thickness(0, help_h, 0, 0);
            }
            else
            {

            }

        }

        private void hide_help_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (help_o > 0)
            {
                help_o -= .05;
                Help_Grid.Opacity = help_o;
            }

            //change position
            if (help_h < te)
            {
                help_h++;
                Help_Grid.Margin = new Thickness(0, help_h, 0, 0);
            }
            else
            {
                //help_Grid.Visibility = Visibility.Collapsed;
                reset();
                (sender as DispatcherTimer).Stop();
            }
        }
        //new customer


        private void new_customer_save_click(object sender, RoutedEventArgs e)
        {
            //validate info
            if (alpha.IsMatch(new_customer_tb_1.Text) == false)
            {
                nc_error.Content = "*First Name has invalid characters";
                nc_error.Visibility = Visibility.Visible;
            }
            else if (alpha.IsMatch(new_customer_tb_2.Text) == false)
            {
                nc_error.Content = "*Last Name has invalid characters";
                nc_error.Visibility = Visibility.Visible;
            }
            else if (beta.IsMatch(new_customer_tb_3.Text) == false)
            {
                nc_error.Content = "*Age has invalid characters";
                nc_error.Visibility = Visibility.Visible;
            }
            else if (nc_rb_1.IsChecked == false && nc_rb_2.IsChecked == false)
            {
                nc_error.Content = "*Please select a gender";
                nc_error.Visibility = Visibility.Visible;
            }
            else
            {
                //save customer to server
                SqlConnection connect = new SqlConnection(getconnected());
                SqlCommand command = new SqlCommand("INSERT into fecCustomer (First, Last, Age, Gender, Attendance) VALUES (@First, @Last, @Age, @Gender, @Attendance)");
                try
                {
                    command.Connection = connect;
                    command.CommandType = CommandType.Text;
                    connect.Open();
                    command.Parameters.Add("@First", SqlDbType.VarChar);
                    command.Parameters["@First"].Value = new_customer_tb_1.Text;
                    command.Parameters.Add("@Last", SqlDbType.VarChar);
                    command.Parameters["@Last"].Value = new_customer_tb_2.Text;
                    command.Parameters.Add("@Age", SqlDbType.VarChar);
                    command.Parameters["@Age"].Value = new_customer_tb_3.Text;
                    command.Parameters.Add("@Gender", SqlDbType.VarChar);
                    if (nc_rb_1.IsChecked == true)
                    {
                        command.Parameters["@Gender"].Value = "male";
                    }
                    else
                    {
                        command.Parameters["@Gender"].Value = "female";
                    }
                    command.Parameters.Add("@Attendance", SqlDbType.VarChar);
                    command.Parameters["@Attendance"].Value = "0";
                    command.ExecuteNonQuery();
                    connect.Close();
                    command.Dispose();
                    connect.Dispose();

                    //reset fields
                    new_customer_tb_1.Clear();
                    new_customer_tb_2.Clear();
                    new_customer_tb_3.Clear();
                    nc_rb_1.IsChecked = false;
                    nc_rb_2.IsChecked = false;

                    //go back to home
                    //./HIDE New_Customer_Grid
                    new_customer_back(null, null);


                }
                catch (Exception el)
                {
                    connect.Close();
                    command.Dispose();
                    connect.Dispose();
                    //connection failed
                   nc_error.Content = "Could not connect to server";
                    nc_error.Visibility = Visibility.Visible;
                    Console.WriteLine(el.StackTrace);
                }
            }
        }

        private void new_customer_back(object sender, RoutedEventArgs e)
        {


            //hide new_customer pane
            DispatcherTimer hide_new_customer = new DispatcherTimer();
            hide_new_customer.Tick += hide_new_customer_Tick;
            hide_new_customer.Interval = new TimeSpan(0, 0, 0, 0, 10);
            hide_new_customer.Start();
            //no need for fadout blur
            main_new_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
        }


        private void new_customer_MouseClick(object sender, RoutedEventArgs e)
        {
            //fade out New grid

            new_back_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
            Thread.Sleep(200);
            blurFront.Visibility = Visibility.Visible;

            //init
            New_Customer_Grid.Margin = new Thickness(0, te, 0, -te);
            New_Customer_Grid.Opacity = 0;
            New_Customer_Grid.Visibility = Visibility.Visible;
            blurFront.Opacity = 1;
            blurFront.Visibility = Visibility.Visible;


            show_new_customer.Tick += show_new_customer_Tick;
            show_new_customer.Interval = new TimeSpan(0, 0, 0, 0, 10);
            show_new_customer.Start();
        }
        int new_customer_h = 20;
        double new_customer_o = 0;
        private void show_new_customer_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (new_customer_o < .9)
            {
                new_customer_o += .05;
                New_Customer_Grid.Opacity = new_customer_o;
            }
            else
            {
                New_Customer_Grid.Opacity = 1;
                (sender as DispatcherTimer).Stop();
            }
            //change position
            if (new_customer_h > 0)
            {
                new_customer_h--;
                New_Customer_Grid.Margin = new Thickness(0, new_customer_h, 0, 0);
            }
            else
            {

            }

        }

        private void hide_new_customer_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (new_customer_o > 0)
            {
                new_customer_o -= .05;
                New_Customer_Grid.Opacity = new_customer_o;
            }

            //change position
            if (new_customer_h < te)
            {
                new_customer_h++;
                New_Customer_Grid.Margin = new Thickness(0, new_customer_h, 0, 0);
            }
            else
            {
                //new_customer_Grid.Visibility = Visibility.Collapsed;
                reset2();
                (sender as DispatcherTimer).Stop();
            }
        }

        //**(*(*(*(*(*
        private void new_employee_save_click(object sender, EventArgs e)
        {
            //validate info
            ne_error.Visibility = Visibility.Collapsed;
            if (alpha.IsMatch(new_employee_tb_1.Text) == false)
            {
                ne_error.Content = "*First Name has invalid characters";
                ne_error.Visibility = Visibility.Visible;
            }
            else if (alpha.IsMatch(new_employee_tb_2.Text) == false)
            {
                ne_error.Content = "*Last Name has invalid characters";
                ne_error.Visibility = Visibility.Visible;
            }
            else if (beta.IsMatch(new_employee_tb_3.Text) == false)
            {
                ne_error.Content = "*Age has invalid characters";
                ne_error.Visibility = Visibility.Visible;
            }
            else
            {
                //save employee to server
                SqlConnection connect = new SqlConnection(getconnected());
                SqlCommand command = new SqlCommand("INSERT into fecEmployee (First, Last, Age, Shifts) VALUES (@First, @Last, @Age, @Shifts)");
                try
                {
                    command.Connection = connect;
                    command.CommandType = CommandType.Text;
                    connect.Open();
                    command.Parameters.Add("@First", SqlDbType.VarChar);
                    command.Parameters["@First"].Value = new_employee_tb_1.Text;
                    command.Parameters.Add("@Last", SqlDbType.VarChar);
                    command.Parameters["@Last"].Value = new_employee_tb_2.Text;
                    command.Parameters.Add("@Age", SqlDbType.VarChar);
                    command.Parameters["@Age"].Value = new_employee_tb_3.Text;
                    command.Parameters.Add("@Shifts", SqlDbType.VarChar);
                    string shifty = "";
                    if (ne_rb_1.IsChecked == true)
                        shifty += "-Monday-";
                    else
                        shifty += "-.-";
                    if (ne_rb_2.IsChecked == true)
                        shifty += "-Tuesday-";
                    else
                        shifty += "-.-";
                    if (ne_rb_3.IsChecked == true)
                        shifty += "-Wednesday-";
                    else
                        shifty += "-.-";
                    if (ne_rb_4.IsChecked == true)
```

```
                        shifty += "-Thursday-";
                    else
                        shifty += "-.-";
                    if (ne_rb_5.IsChecked == true)
                        shifty +=  "-Friday-";
                    else
                        shifty += "-.-";
                    if (ne_rb_6.IsChecked == true)
                        shifty += "-Saturday-";
                    else
                        shifty += "-.-";
                    command.Parameters["@Shifts"].Value = shifty;

                    command.ExecuteNonQuery();
                    connect.Close();
                    command.Dispose();
                    connect.Dispose();

                    //reset fields
                    new_employee_tb_1.Clear();
                    new_employee_tb_2.Clear();
                    new_employee_tb_3.Clear();
                    ne_rb_1.IsChecked = false;
                    ne_rb_2.IsChecked = false;
                    ne_rb_3.IsChecked = false;
                    ne_rb_4.IsChecked = false;
                    ne_rb_5.IsChecked = false;
                    ne_rb_6.IsChecked = false;


                    //go back to home
                    //./HIDE New_Employee_Grid
                    new_employee_back(null, null);


                }
                catch (Exception el)
                {
                    connect.Close();
                    command.Dispose();
                    connect.Dispose();
                    //connection failed
                    ne_error.Content = "Could not connect to server";
                    ne_error.Visibility = Visibility.Visible;
                    Console.WriteLine(el.StackTrace);
                }
            }
        }
        private void new_employee_back(object sender, RoutedEventArgs e)
        {


            //hide new_employee pane
            DispatcherTimer hide_new_employee = new DispatcherTimer();
            hide_new_employee.Tick += hide_new_employee_Tick;
            hide_new_employee.Interval = new TimeSpan(0, 0, 0, 0, 10);
            hide_new_employee.Start();
            //no need for fadout blur
            main_new_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
        }


        private void new_employee_MouseClick(object sender, RoutedEventArgs e)
        {
            //fade out New grid

            new_back_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
            Thread.Sleep(200);
            blurFront.Visibility = Visibility.Visible;

            //init
            New_Employee_Grid.Margin = new Thickness(0, te, 0, -te);
            New_Employee_Grid.Opacity = 0;
            New_Employee_Grid.Visibility = Visibility.Visible;
            blurFront.Opacity = 1;
            blurFront.Visibility = Visibility.Visible;


            DispatcherTimer show_new_employee = new DispatcherTimer();
            show_new_employee.Tick += show_new_employee_Tick;
            show_new_employee.Interval = new TimeSpan(0, 0, 0, 0, 10);
            show_new_employee.Start();
        }
        int new_employee_h = 20;
        double new_employee_o = 0;
        private void show_new_employee_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (new_employee_o < .9)
            {
                new_employee_o += .05;
                New_Employee_Grid.Opacity = new_employee_o;
            }
            else
            {
                New_Employee_Grid.Opacity = 1;
                (sender as DispatcherTimer).Stop();
            }
            //change position
            if (new_employee_h > 0)
            {
                new_employee_h--;
                New_Employee_Grid.Margin = new Thickness(0, new_employee_h, 0, 0);
            }
            else
            {

            }

        }

        private void hide_new_employee_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (new_employee_o > 0)
            {
                new_employee_o -= .05;
                New_Employee_Grid.Opacity = new_employee_o;
            }

            //change position
            if (new_employee_h < te)
            {
                new_employee_h++;
                New_Employee_Grid.Margin = new Thickness(0, new_employee_h, 0, 0);
            }
            else
            {
                //new_employee_Grid.Visibility = Visibility.Collapsed;
                reset2();
                (sender as DispatcherTimer).Stop();
            }
        }
        //()()()())()()(()()())()()()((
        private void edit_customer_plus_click(object sender, EventArgs e)
        {
            int num = Convert.ToInt32(edit_customer_attCount.Content);
            Console.WriteLine(num.ToString() + "!");
            num++;
            edit_customer_attCount.Content = Convert.ToString(num);
        }
        private void edit_customer_minus_click(object sender, EventArgs e)
        {
            int num = Convert.ToInt32(edit_customer_attCount.Content);
            Console.WriteLine(num.ToString() + "!");
            num--;
            if (num >= 0)
            {
                edit_customer_attCount.Content = Convert.ToString(num);
            }
            else
            {
                edit_customer_attCount.Content = "0";
            }

        }
        private void edit_customer_find_click(object sender, EventArgs e)
        {
           if (ec_cover.Visibility != Visibility.Visible)
            {
                //validate info
                ne_error.Visibility = Visibility.Collapsed;
                if (alpha.IsMatch(edit_customer_tb_1.Text) == false)
                {
                    ec_error.Content = "*First Name has invalid characters";
                    ec_error.Visibility = Visibility.Visible;
                }
                else if (alpha.IsMatch(edit_customer_tb_2.Text) == false)
                {
                    ec_error.Content = "*Last Name has invalid characters";
                    ec_error.Visibility = Visibility.Visible;
                }
                else
                {
                    //check if customer exists
                    SqlConnection connect = new SqlConnection(getconnected());
                    SqlCommand command = new SqlCommand("SELECT First, Last, Age, Attendance FROM fecCustomer WHERE First = @First AND Last = @Last");
                    try
                    {
                        command.Connection = connect;
                        command.CommandType = CommandType.Text;
                        connect.Open();
                        command.Parameters.Add("@First", SqlDbType.VarChar);
                        command.Parameters["@First"].Value = edit_customer_tb_1.Text;
                        command.Parameters.Add("@Last", SqlDbType.VarChar);
                        command.Parameters["@Last"].Value = edit_customer_tb_2.Text;

                        SqlDataReader sqlReader = command.ExecuteReader();
                        if (sqlReader.HasRows == true)
                        {
                            //we got a result
                            ec_error.Content = "Customer Found";
                            ec_error.Visibility = Visibility.Visible;
                            while(sqlReader.Read())
                            {
                                edit_customer_tb_3.Text = sqlReader.GetString(2);
                                if (Convert.ToInt32(sqlReader.GetString(3)) > 0)
                                {
                                    edit_customer_attCount.Content = sqlReader.GetString(3);
                                }
                                else
                                {
                                    edit_customer_attCount.Content = "0";
                                }
                                break;
                            }
                            ec_cover.Visibility = Visibility.Visible;
                            edit_customer_save_btn.Content = "Save";



                        }
                        else
                        {
                            //no result
                            ec_error.Content = "*Customer not found";
                            ec_error.Visibility = Visibility.Visible;
                        }

                        connect.Close();
                        command.Dispose();
                        connect.Dispose();



                        //go back to home
                        //./HIDE Edit_Customer_Grid
                       // edit_customer_back(null, null);


                    }
                    catch (Exception el)
                    {
                        connect.Close();
                        command.Dispose();
                        connect.Dispose();
                        //connection failed
                        ec_error.Content = "Could not connect to server";
                        ec_error.Visibility = Visibility.Visible;
                        Console.WriteLine(el.StackTrace);
                    }
                }
            }
           else
            {
                //editing time!
                //validate info
                ec_error.Visibility = Visibility.Collapsed;
                if (alpha.IsMatch(edit_customer_tb_1.Text) == false)
                {
                    nc_error.Content = "*First Name has invalid characters";
                    nc_error.Visibility = Visibility.Visible;
                }
                else if (alpha.IsMatch(edit_customer_tb_2.Text) == false)
                {
                    nc_error.Content = "*Last Name has invalid characters";
                    nc_error.Visibility = Visibility.Visible;
                }
                else if (beta.IsMatch(edit_customer_tb_3.Text) == false)
                {
                    nc_error.Content = "*Age has invalid characters";
                    nc_error.Visibility = Visibility.Visible;
                }
                else
                {
                    //save customer to server
                    SqlConnection connect = new SqlConnection(getconnected());
                    SqlCommand command = new SqlCommand("UPDATE fecCustomer SET
                    First=@First, Last=@Last, Age=@Age, Attendance=@Attendance
```

```
                    WHERE First = @First AND Last = @Last");
                    try
                    {
                        command.Connection = connect;
                        command.CommandType = CommandType.Text;
                        connect.Open();
                        command.Parameters.Add("@First", SqlDbType.VarChar);
                        command.Parameters["@First"].Value = edit_customer_tb_1.Text;
                        command.Parameters.Add("@Last", SqlDbType.VarChar);
                        command.Parameters["@Last"].Value = edit_customer_tb_2.Text;
                        command.Parameters.Add("@Age", SqlDbType.VarChar);
                        command.Parameters["@Age"].Value = edit_customer_tb_3.Text;

                        command.Parameters.Add("@Attendance", SqlDbType.VarChar);
                        command.Parameters["@Attendance"].Value = Convert.ToString(edit_customer_attCount.Content);
                        command.ExecuteNonQuery();
                        connect.Close();
                        command.Dispose();
                        connect.Dispose();

                        //reset fields
                        edit_customer_tb_1.Clear();
                        edit_customer_tb_2.Clear();
                        edit_customer_tb_3.Clear();
                        edit_customer_save_btn.Content = "Find";

                        //go back to home
                        //./HIDE edit_Customer_Grid
                        edit_customer_back(null, null);


                    }
                    catch (Exception el)
                    {
                        connect.Close();
                        command.Dispose();
                        connect.Dispose();
                        //connection failed
                        ec_error.Content = "Could not connect to server";
                        ec_error.Visibility = Visibility.Visible;
                        Console.WriteLine(el.StackTrace);
                    }
                }

            }
        }
        private void edit_customer_back(object sender, RoutedEventArgs e)
        {
            ec_error.Visibility = Visibility.Collapsed;
            if (ec_cover.Visibility != Visibility.Visible)
            {

                //hide edit_customer pane
                DispatcherTimer hide_edit_customer = new DispatcherTimer();
                hide_edit_customer.Tick += hide_edit_customer_Tick;
                hide_edit_customer.Interval = new TimeSpan(0, 0, 0, 0, 10);
                hide_edit_customer.Start();
                //no need for fadout blur
                main_edit_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
            }
            else
            {
                //dont exit pane, but reset to std edit view
                edit_customer_save_btn.Content = "Find";
                ec_cover.Visibility = Visibility.Collapsed;
                edit_customer_tb_1.Clear();
                edit_customer_tb_2.Clear();
            }
        }


        private void edit_customer_MouseClick(object sender, RoutedEventArgs e)
        {
            //fade out New grid
            ec_cover.Visibility = Visibility.Collapsed;
            edit_customer_tb_1.Clear();
            edit_customer_tb_2.Clear();
            edit_customer_tb_3.Clear();
            edit_customer_save_btn.Content = "Find";

            edit_back_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));

            blurFront.Visibility = Visibility.Visible;

            //init
            Edit_Customer_Grid.Margin = new Thickness(0, te, 0, -te);
            Edit_Customer_Grid.Opacity = 0;
            Edit_Customer_Grid.Visibility = Visibility.Visible;
            blurFront.Opacity = 1;
            blurFront.Visibility = Visibility.Visible;


            DispatcherTimer show_edit_customer = new DispatcherTimer();
            show_edit_customer.Tick += show_edit_customer_Tick;
            show_edit_customer.Interval = new TimeSpan(0, 0, 0, 0, 10);
            show_edit_customer.Start();
        }
        int edit_customer_h = 20;
        double edit_customer_o = 0;
        private void show_edit_customer_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (edit_customer_o < .9)
            {
                edit_customer_o += .05;
                Edit_Customer_Grid.Opacity = edit_customer_o;
            }
            else
            {
                Edit_Customer_Grid.Opacity = 1;
                (sender as DispatcherTimer).Stop();
            }
            //change position
            if (edit_customer_h > 0)
            {
                edit_customer_h--;
                Edit_Customer_Grid.Margin = new Thickness(0, edit_customer_h, 0, 0);
            }
            else
            {

            }

        }

        private void hide_edit_customer_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (edit_customer_o > 0)
            {
                edit_customer_o -= .05;
                Edit_Customer_Grid.Opacity = edit_customer_o;
            }

            //change position
            if (edit_customer_h < te)
            {
                edit_customer_h++;
                Edit_Customer_Grid.Margin = new Thickness(0, edit_customer_h, 0, 0);
            }
            else
            {
                //edit_customer_Grid.Visibility = Visibility.Collapsed;
                reset2();
                (sender as DispatcherTimer).Stop();
            }
        }

        //*
        //EDIT EMPLOYEE
        //*

        private void edit_employee_find_click(object sender, EventArgs e)
        {
            if (ee_cover.Visibility != Visibility.Visible)
            {
                //validate info
                ee_error.Visibility = Visibility.Collapsed;
                if (alpha.IsMatch(edit_employee_tb_1.Text) == false)
                {
                    ee_error.Content = "*First Name has invalid characters";
                    ee_error.Visibility = Visibility.Visible;
                }
                else if (alpha.IsMatch(edit_employee_tb_2.Text) == false)
                {
                    ee_error.Content = "*Last Name has invalid characters";
                    ee_error.Visibility = Visibility.Visible;
                }
                else
                {
                    //check if customer exists
                    SqlConnection connect = new SqlConnection(getconnected());
                    SqlCommand command = new SqlCommand("SELECT First, Last, Age, Shifts FROM fecEmployee WHERE First = @First AND Last = @Last");
                    try
                    {
                        command.Connection = connect;
                        command.CommandType = CommandType.Text;
                        connect.Open();
                        command.Parameters.Add("@First", SqlDbType.VarChar);
                        command.Parameters["@First"].Value = edit_employee_tb_1.Text;
                        command.Parameters.Add("@Last", SqlDbType.VarChar);
                        command.Parameters["@Last"].Value = edit_employee_tb_2.Text;

                        SqlDataReader sqlReader = command.ExecuteReader();
                        if (sqlReader.HasRows == true)
                        {
                            //we got a result
                            ee_error.Content = "Employee Found";
                            ee_error.Visibility = Visibility.Visible;
                            while (sqlReader.Read())
                            {
                                edit_employee_tb_3.Text = Convert.ToString(sqlReader.GetValue(2));
                                //take shifts and parse
                                String[] sifts = sqlReader.GetString(3).Split('-');
                                sifts = sifts.Where(x => !string.IsNullOrEmpty(x)).ToArray();
                                if (sifts[0] != ".")
                                    ee_rb_1.IsChecked = true;
                                if (sifts[1] != ".")
                                    ee_rb_2.IsChecked = true;
                                if (sifts[2] != ".")
                                    ee_rb_3.IsChecked = true;
                                if (sifts[3] != ".")
                                    ee_rb_4.IsChecked = true;
                                if (sifts[4] != ".")
                                    ee_rb_5.IsChecked = true;
                                if (sifts[5] != ".")
                                    ee_rb_6.IsChecked = true;

                                break;

                            }
                            ee_cover.Visibility = Visibility.Visible;
                            edit_employee_save_btn.Content = "Save";



                        }
                        else
                        {
                            //no result
                            ee_error.Content = "*Employee not found";
                            ee_error.Visibility = Visibility.Visible;
                        }

                        connect.Close();
                        command.Dispose();
                        connect.Dispose();



                        //go back to home
                        //./HIDE Edit_Employee_Grid
                        // edit_employee_back(null, null);


                    }
                    catch (Exception el)
                    {
                        connect.Close();
                        command.Dispose();
                        connect.Dispose();
                        //connection failed
                        ee_error.Content = "Could not connect to server";
                        ee_error.Visibility = Visibility.Visible;
                        Console.WriteLine(el.StackTrace);
                    }
                }
            }
            else
            {
                //editing time!
                //validate info
                ee_error.Visibility = Visibility.Collapsed;
                if (alpha.IsMatch(edit_employee_tb_1.Text) == false)
                {
                    ee_error.Content = "*First Name has invalid characters";
                    ee_error.Visibility = Visibility.Visible;
                }
                else if (alpha.IsMatch(edit_employee_tb_2.Text) == false)
                {
                    ee_error.Content = "*Last Name has invalid characters";
                    ee_error.Visibility = Visibility.Visible;
                }
                else if (beta.IsMatch(edit_employee_tb_3.Text) == false)
                {
                    ee_error.Content = "*Age has invalid characters";
                    ee_error.Visibility = Visibility.Visible;
                }
                else
                {
                    //save customer to server
                    SqlConnection connect = new SqlConnection(getconnected());
                    SqlCommand command = new SqlCommand("UPDATE fecEmployee SET First=@First, Last=@Last, Age=@Age, Shifts=@Shifts WHERE First = @First AND Last = @Last");
                    try
                    {
                        command.Connection = connect;
                        command.CommandType = CommandType.Text;
                        connect.Open();
                        command.Parameters.Add("@First", SqlDbType.VarChar);
                        command.Parameters["@First"].Value = edit_employee_tb_1.Text;
                        command.Parameters.Add("@Last", SqlDbType.VarChar);
                        command.Parameters["@Last"].Value = edit_employee_tb_2.Text;
                        command.Parameters.Add("@Age", SqlDbType.VarChar);
                        command.Parameters["@Age"].Value = edit_employee_tb_3.Text;
                        string shifty = "";
                        if (ee_rb_1.IsChecked == true)
                            shifty += "-Monday-";
                        else
                            shifty += "-.-";
                        if (ee_rb_2.IsChecked == true)
                            shifty += "-Tuesday-";
                        else
                            shifty += "-.-";
                        if (ee_rb_3.IsChecked == true)
                            shifty += "-Wedeesday-";
                        else
                            shifty += "-.-";
                        if (ee_rb_4.IsChecked == true)
                            shifty += "-Thursday-";
                        else
                            shifty += "-.-";
                        if (ee_rb_5.IsChecked == true)
                            shifty += "-Friday-";
                        else
                            shifty += "-.-";
                        if (ee_rb_6.IsChecked == true)
                            shifty += "-Saturday-";
                        else
                            shifty += "-.-";
                        command.Parameters.Add("@Shifts", SqlDbType.VarChar);
                        command.Parameters["@Shifts"].Value = shifty;
                        command.ExecuteNonQuery();
                        connect.Close();
                        command.Dispose();
                        connect.Dispose();

                        //reset fields
                        edit_employee_tb_1.Clear();
                        edit_employee_tb_2.Clear();
                        edit_employee_tb_3.Clear();
                        ee_rb_1.IsChecked = false;
                        ee_rb_2.IsChecked = false;
                        ee_rb_3.IsChecked = false;
                        ee_rb_4.IsChecked = false;
                        ee_rb_5.IsChecked = false;
                        ee_rb_6.IsChecked = false;
                        ec_cover.Visibility = Visibility.Collapsed;
                        edit_employee_save_btn.Content = "Find";

                        //go back to home
                        //./HIDE edit_Customer_Grid
                        edit_employee_back(null, null);


                    }
                    catch (Exception el)
                    {
                        connect.Close();
                        command.Dispose();
                        connect.Dispose();
                        //connection failed
                        ee_error.Content = "Could not connect to server";
                        ee_error.Visibility = Visibility.Visible;
                        Console.WriteLine(el.StackTrace);
                    }
                }

            }
        }
        private void edit_employee_back(object sender, RoutedEventArgs e)
        {
            ee_error.Visibility = Visibility.Collapsed;
            if (ee_cover.Visibility != Visibility.Visible)
            {

                //hide edit_employee pane
                DispatcherTimer hide_edit_employee = new DispatcherTimer();
                hide_edit_employee.Tick += hide_edit_employee_Tick;
                hide_edit_employee.Interval = new TimeSpan(0, 0, 0, 0, 10);
                hide_edit_employee.Start();
                //no need for fadout blur
                main_edit_btn.RaiseEvent(new RoutedEventArgs
                  (Button.ClickEvent));
            else
            {
                //dont exit pane, but reset to std edit view
                edit_employee_save_btn.Content = "Find";
                ee_cover.Visibility = Visibility.Collapsed;
                edit_employee_tb_1.Clear();
                edit_employee_tb_2.Clear();
            }
        }


        private void edit_employee_MouseClick(object sender, RoutedEventArgs e)
        {
            //fade out New grid
            ee_cover.Visibility = Visibility.Collapsed;
            edit_employee_tb_1.Clear();
            edit_employee_tb_2.Clear();
            edit_employee_tb_3.Clear();
            edit_employee_save_btn.Content = "Find";

            edit_back_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));

            blurFront.Visibility = Visibility.Visible;

            //init
            Edit_Employee_Grid.Margin = new Thickness(0, te, 0, -te);
            Edit_Employee_Grid.Opacity = 0;
            Edit_Employee_Grid.Visibility = Visibility.Visible;
            blurFront.Opacity = 1;
            blurFront.Visibility = Visibility.Visible;


            DispatcherTimer show_edit_employee = new DispatcherTimer();
            show_edit_employee.Tick += show_edit_employee_Tick;
            show_edit_employee.Interval = new TimeSpan(0, 0, 0, 0, 10);
            show_edit_employee.Start();
        }
        int edit_employee_h = 20;
        double edit_employee_o = 0;
        private void show_edit_employee_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (edit_employee_o < .9)
            {
                edit_employee_o += .05;
                Edit_Employee_Grid.Opacity = edit_employee_o;
            }
            else
            {
                Edit_Employee_Grid.Opacity = 1;
                (sender as DispatcherTimer).Stop();
            }
            //change position
            if (edit_employee_h > 0)
            {
                edit_employee_h--;
                Edit_Employee_Grid.Margin = new Thickness(0, edit_employee_h, 0, 0);
            }
            else
            {

            }

        }

        private void hide_edit_employee_Tick(object sender, EventArgs e)
        {
            //change alhpa
            if (edit_employee_o > 0)
            {
                edit_employee_o -= .05;
                Edit_Employee_Grid.Opacity = edit_employee_o;
            }

            //change position
            if (edit_employee_h < te)
            {
                edit_employee_h++;
                Edit_Employee_Grid.Margin = new Thickness(0, edit_employee_h, 0, 0);
            }
            else
            {
                //edit_employee_Grid.Visibility = Visibility.Collapsed;
                reset2();
                (sender as DispatcherTimer).Stop();
            }
        }

        //VIEW REPORT
        public Dictionary<string, Process> OpenedProcesses = new Dictionary<string, Process>(StringComparer.OrdinalIgnoreCase);
        private void view_customers_Report(object sender, RoutedEventArgs e)
        {

            SqlConnection connect = new SqlConnection(getconnected());
            SqlCommand command = new SqlCommand("SELECT First, Last, Gender, Age, Attendance FROM fecCustomer");
            try
            {
                string filePath = System.Environment.CurrentDirectory + "\\customer_report.txt";
                command.Connection = connect;
                command.CommandType = CommandType.Text;
                connect.Open();
                SqlDataReader sqlReader = command.ExecuteReader();
                if (sqlReader.HasRows == true)
                {
                    //we got results
                    File.Delete(filePath);
                    StreamWriter file = new StreamWriter(filePath, true);
                    int id = 1;
                    while (sqlReader.Read())
                    {
                        //take shifts and parse
                        /*
                        String[] sifts = sqlReader.GetString(3).Split('-');
                        sifts = sifts.Where(x => !string.IsNullOrEmpty(x)).ToArray();
                        sifts = sifts.Where(x => (x != ".")).ToArray();
                        */
                        //write to text file line
                        file.WriteLine(id.ToString() + ": " + (sqlReader.GetValue(0)).ToString() + " " + (sqlReader.GetValue(1)).ToString() + " Gender-" + (sqlReader.GetValue(2)).ToString() + " Age-" + (sqlReader.GetValue(3)).ToString() + " Attendance-" + (sqlReader.GetValue(4)).ToString());
                        id++;
                    }
                    //show text file
                    file.Close();
                    Process proc;
                    if (this.OpenedProcesses.TryGetValue(filePath, out proc))
                    {
                        v_error.Content = "The file is already open!";
                        v_error.Visibility = Visibility.Visible;
                        // it could be possible to activate the window of the open process but that is another question on its own.

                    }
                    else
                    {
                        proc = System.Diagnostics.Process.Start(filePath);
                        this.OpenedProcesses[filePath] = proc;
                    }


                }
                else
                {
                    //no result
                    v_error.Content = "*No customers exist";
                    v_error.Visibility = Visibility.Visible;
                }

                connect.Close();
                command.Dispose();
                connect.Dispose();



                //go back to home
                //./HIDE Edit_Employee_Grid
                // edit_employee_back(null, null);
```

```
            }
            catch (Exception el)
            {
                connect.Close();
                command.Dispose();
                connect.Dispose();
                //connection failed
                v_error.Content = "Could not connect to server";
                v_error.Visibility = Visibility.Visible;
                Console.WriteLine(el.StackTrace);
            }
        }
        private void view_employees_Report(object sender, RoutedEventArgs e)
        {
            SqlConnection connect = new SqlConnection(getconnected());
            SqlCommand command = new SqlCommand("SELECT First, Last, Age, Shifts FROM fecEmployee");
            try
            {
                string filePath = System.Environment.CurrentDirectory + "\\employee_report.txt";

                command.Connection = connect;
                command.CommandType = CommandType.Text;
                connect.Open();
                SqlDataReader sqlReader = command.ExecuteReader();
                if (sqlReader.HasRows == true)
                {
                    //we got results
                    File.Delete(filePath);
                    StreamWriter file = new StreamWriter(filePath, true);
                    int id = 1;
                    while (sqlReader.Read())
                    {
                        //take shifts and parse

                        String[] sifts = sqlReader.GetString(3).Split('-');
                        sifts = sifts.Where(x => !string.IsNullOrEmpty(x)).ToArray();
                        sifts = sifts.Where(x => (x != ".")).ToArray();

                        //write to text file line
                        file.WriteLine(id.ToString() + ": " + (sqlReader.GetValue(0)).ToString() + " " + (sqlReader.GetValue(1)).ToString() + " Age-" + (sqlReader.GetValue(2)).ToString() + " Shifts-" + String.Join(", ", sifts));
                        id++;
                    }
                    //show text file
                    file.Close();
                    Process proc;
                    if (this.OpenedProcesses.TryGetValue(filePath, out proc) && !proc.HasExited)
                    {
                        v_error.Content = "The file is already open!";
                        v_error.Visibility = Visibility.Visible;
                        // it could be possible to activate the window of the open process but that is another question on its own.

                    }
                    else
                    {
                        proc = System.Diagnostics.Process.Start(filePath);
                        this.OpenedProcesses[filePath] = proc;
                    }
                }
                else
                {
                    //no result
                    ee_error.Content = "*No Employees exist";
                    ee_error.Visibility = Visibility.Visible;
                }

                connect.Close();
                command.Dispose();
                connect.Dispose();



                //go back to home
                //./HIDE Edit_Employee_Grid
                // edit_employee_back(null, null);


            }
            catch (Exception el)
            {
                connect.Close();
                command.Dispose();
                connect.Dispose();
                //connection failed
                ee_error.Content = "Could not connect to server";
                ee_error.Visibility = Visibility.Visible;
                Console.WriteLine(el.StackTrace);
            }
        }

    }
}

```