

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.IO;
5 using System.Windows;
6 using System.Windows.Controls;
7 using System.Windows.Input;
8 using System.Windows.Threading;
9 using System.Threading;
10 using System.Text.RegularExpressions;
11 using System.Data.SqlClient;
12 using System.Diagnostics;
13 using System.Data;
14 /**
15 //COPYRIGHT Weingartner 2017
16 /**
17 namespace FEC
18 {
19     /// <summary>
20     /// Interaction logic for MainWindow.xaml
21     /// </summary>
22     public partial class MainWindow : Window
23     {
24         public MainWindow()
25         {
26             InitializeComponent();
27         }
28
29         int te = 20;
30         private static string getconnected()
31         {
32             //create connection string
33             SqlConnectionStringBuilder b = new SqlConnectionStringBuilder();
34             b.DataSource = "mssql3.gear.host";
35             b.InitialCatalog = "phoenix";
36             b.IntegratedSecurity = false;
37             b.UserID = "phoenix";
38             b.Password = "@CorpNetWeRSecure!&";
39             b.PersistSecurityInfo = true;
40             b.TrustServerCertificate = true;
41             b.ConnectTimeout = 200;
42             string connectionString = b.ConnectionString;
43             return connectionString;
44         }
45         private void MainWindow_MouseDown(object sender, MouseButtonEventArgs e)
46         {
47             if (e.LeftButton == MouseButtonState.Pressed)
48             {
49                 DragMove();
50             }
51         }
52         readonly Regex alpha = new Regex(@"^[a-zA-Z]");
53         readonly Regex beta = new Regex(@"^[0-9]");
54         private void reset()
```

```
55     {
56         //reset everything to default
57         blurFront.Opacity = 0;
58         blurFront.Visibility = Visibility.Collapsed;
59         New_Grid.Visibility = Visibility.Collapsed;
60         New_Grid.Margin = new Thickness(0, 0, 0, 0);
61         New_Grid.Opacity = 0;
62
63         Edit_Grid.Visibility = Visibility.Collapsed;
64         Edit_Grid.Margin = new Thickness(0, 0, 0, 0);
65         Edit_Grid.Opacity = 0;
66
67         View_Grid.Visibility = Visibility.Collapsed;
68         View_Grid.Margin = new Thickness(0, 0, 0, 0);
69         View_Grid.Opacity = 0;
70
71         Help_Grid.Visibility = Visibility.Collapsed;
72         Help_Grid.Margin = new Thickness(0, -40, 0, 0);
73         Help_Grid.Opacity = 0;
74
75
76     }
77
78     private void reset2()
79     {
80
81         New_Customer_Grid.Visibility = Visibility.Collapsed;
82         New_Customer_Grid.Margin = new Thickness(0, 0, 0, 0);
83         New_Customer_Grid.Opacity = 0;
84
85         New_Employee_Grid.Visibility = Visibility.Collapsed;
86         New_Employee_Grid.Margin = new Thickness(0, 0, 0, 0);
87         New_Employee_Grid.Opacity = 0;
88
89         Edit_Customer_Grid.Visibility = Visibility.Collapsed;
90         Edit_Customer_Grid.Margin = new Thickness(0, 0, 0, 0);
91         Edit_Customer_Grid.Opacity = 0;
92
93         Edit_Employee_Grid.Visibility = Visibility.Collapsed;
94         Edit_Employee_Grid.Margin = new Thickness(0, 0, 0, 0);
95         Edit_Employee_Grid.Opacity = 0;
96
97
98     }
99
100    private void loaded(object sender, RoutedEventArgs e)
101    {
102
103
104        reset();
105        reset2();
106    }
107    private void quit_MouseClick(object sender, RoutedEventArgs e)
108    {
109        this.Close();
110    }
```

```
111
112     private void minimize_MouseClick(object sender, RoutedEventArgs e)
113     {
114         this.WindowState = WindowState.Minimized;
115     }
116
117
118     private void new_back(object sender, RoutedEventArgs e)
119     {
120
121
122         //hide new pane
123         DispatcherTimer hide_new = new DispatcherTimer();
124         hide_new.Tick += hide_new_Tick;
125         hide_new.Interval = new TimeSpan(0, 0, 0, 0, 15);
126         hide_new.Start();
127         //fadeout homeblur
128         DispatcherTimer fadeout_home = new DispatcherTimer();
129         fadeout_home.Tick += fadeout_home_Tick;
130         fadeout_home.Interval = new TimeSpan(0, 0, 0, 0, 15);
131         fadeout_home.Start();
132     }
133
134     double blurOpa = 0;
135     private void fadein_home_Tick(object sender, EventArgs e)
136     {
137         if (blurOpa < 1)
138         {
139             blurOpa += .04;
140             blurFront.Opacity = blurOpa;
141         }
142         else
143         {
144             blurFront.Opacity = 1;
145             blurOpa = 1;
146             (sender as DispatcherTimer).Stop();
147         }
148     }
149
150     private void fadeout_home_Tick(object sender, EventArgs e)
151     {
152         if (blurOpa > 0)
153         {
154             blurOpa -= .04;
155             blurFront.Opacity = blurOpa;
156         }
157         else
158         {
159             //blurOpa = 0;
160             blurFront.Opacity = 0;
161             blurFront.Visibility = Visibility.Collapsed;
162
163             (sender as DispatcherTimer).Stop();
164         }
165     }
166
167
```

```
168     {
169         blurFront.Visibility = Visibility.Visible;
170         //init
171         New_Grid.Margin = new Thickness(0, te, 0, -te);
172         New_Grid.Opacity = 0;
173         New_Grid.Visibility = Visibility.Visible;
174         //fadein homeblur
175         DispatcherTimer fadein_home = new DispatcherTimer();
176         fadein_home.Tick += fadein_home_Tick;
177         fadein_home.Interval = new TimeSpan(0, 0, 0, 0, 15);
178         fadein_home.Start();
179
180         DispatcherTimer show_new = new DispatcherTimer();
181         show_new.Tick += show_new_Tick;
182         show_new.Interval = new TimeSpan(0, 0, 0, 0, 15);
183         show_new.Start();
184     }
185     int new_h = 20;
186     double new_o = 0;
187     private void show_new_Tick(object sender, EventArgs e)
188     {
189         //change alhpa
190         if (new_o < .9)
191         {
192             new_o += .05;
193             New_Grid.Opacity = new_o;
194         }
195         else
196         {
197             New_Grid.Opacity = 1;
198             (sender as DispatcherTimer).Stop();
199         }
200         //change position
201         if(new_h > 0)
202         {
203             new_h--;
204             New_Grid.Margin = new Thickness(0, new_h, 0, 0);
205         }
206         else
207         {
208         }
209     }
210 }
211
212
213 private void hide_new_Tick(object sender, EventArgs e)
214 {
215     //change alhpa
216     if (new_o > 0)
217     {
218         new_o -= .05;
219         New_Grid.Opacity = new_o;
220     }
221     //change position
222     if (new_h < te)
223     {
```

```
224         new_h++;
225         New_Grid.Margin = new Thickness(0, new_h, 0, 0);
226     }
227     else
228     {
229         //New_Grid.Visibility = Visibility.Collapsed;
230         reset();
231         (sender as DispatcherTimer).Stop();
232     }
233 }
234
235
236 private void edit_back(object sender, RoutedEventArgs e)
237 {
238
239
240     //hide edit pane
241     DispatcherTimer hide_edit = new DispatcherTimer();
242     hide_edit.Tick += hide_edit_Tick;
243     hide_edit.Interval = new TimeSpan(0, 0, 0, 0, 15);
244     hide_edit.Start();
245     //fadeout homeblur
246     DispatcherTimer fadeout_home = new DispatcherTimer();
247     fadeout_home.Tick += fadeout_home_Tick;
248     fadeout_home.Interval = new TimeSpan(0, 0, 0, 0, 15);
249     fadeout_home.Start();
250 }
251
252
253
254 private void edit_MouseClick(object sender, RoutedEventArgs e)
255 {
256     blurFront.Visibility = Visibility.Visible;
257     //init
258     Edit_Grid.Margin = new Thickness(0, te, 0, -te);
259     Edit_Grid.Opacity = 0;
260     Edit_Grid.Visibility = Visibility.Visible;
261     //fadein homeblur
262     DispatcherTimer fadein_home = new DispatcherTimer();
263     fadein_home.Tick += fadein_home_Tick;
264     fadein_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
265     fadein_home.Start();
266
267     DispatcherTimer show_edit = new DispatcherTimer();
268     show_edit.Tick += show_edit_Tick;
269     show_edit.Interval = new TimeSpan(0, 0, 0, 0, 10);
270     show_edit.Start();
271 }
272 int edit_h = 20;
273 double edit_o = 0;
274 private void show_edit_Tick(object sender, EventArgs e)
275 {
276     //change alhpa
277     if (edit_o < .9)
278     {
```

```
279         edit_o += .05;
280         Edit_Grid.Opacity = edit_o;
281     }
282     else
283     {
284
285         Edit_Grid.Opacity = 1;
286         (sender as DispatcherTimer).Stop();
287     }
288     //change position
289     if (edit_h > 0)
290     {
291         edit_h--;
292         Edit_Grid.Margin = new Thickness(0, edit_h, 0, 0);
293     }
294     else
295     {
296
297     }
298 }
299
300
301 private void hide_edit_Tick(object sender, EventArgs e)
302 {
303     //change alhpa
304     if (edit_o > 0)
305     {
306         edit_o -= .05;
307         Edit_Grid.Opacity = edit_o;
308     }
309     //change position
310     if (edit_h < te)
311     {
312         edit_h++;
313         Edit_Grid.Margin = new Thickness(0, edit_h, 0, 0);
314     }
315     else
316     {
317         //edit_Grid.Visibility = Visibility.Collapsed;
318         reset();
319         (sender as DispatcherTimer).Stop();
320     }
321 }
322
323 private void view_back(object sender, RoutedEventArgs e)
324 {
325
326
327     //hide view pane
328     DispatcherTimer hide_view = new DispatcherTimer();
329     hide_view.Tick += hide_view_Tick;
330     hide_view.Interval = new TimeSpan(0, 0, 0, 0, 10);
331     hide_view.Start();
332     //fadeout homeblur
333     DispatcherTimer fadeout_home = new DispatcherTimer();
334     fadeout_home.Tick += fadeout_home_Tick;
```

```
335         fadeout_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
336         fadeout_home.Start();
337     }
338 }
339
340
341 private void view_MouseClick(object sender, RoutedEventArgs e)
342 {
343     blurFront.Visibility = Visibility.Visible;
344     //init
345     View_Grid.Margin = new Thickness(0, te, 0, -te);
346     View_Grid.Opacity = 0;
347     View_Grid.Visibility = Visibility.Visible;
348     //fadein homeblur
349     DispatcherTimer fadein_home = new DispatcherTimer();
350     fadein_home.Tick += fadein_home_Tick;
351     fadein_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
352     fadein_home.Start();
353
354     DispatcherTimer show_view = new DispatcherTimer();
355     show_view.Tick += show_view_Tick;
356     show_view.Interval = new TimeSpan(0, 0, 0, 0, 10);
357     show_view.Start();
358 }
359 int view_h = 20;
360 double view_o = 0;
361 private void show_view_Tick(object sender, EventArgs e)
362 {
363     //change alhpa
364     if (view_o < .9)
365     {
366         view_o += .05;
367         View_Grid.Opacity = view_o;
368     }
369     else
370     {
371         View_Grid.Opacity = 1;
372         (sender as DispatcherTimer).Stop();
373     }
374     //change position
375     if (view_h > 0)
376     {
377         view_h--;
378         View_Grid.Margin = new Thickness(0, view_h, 0, 0);
379     }
380     else
381     {
382     }
383 }
384 }
385
386
387 private void hide_view_Tick(object sender, EventArgs e)
388 {
389     //change alhpa
390     if (view_o < 0)
```

```
391     {
392         view_o -= .05;
393         View_Grid.Opacity = view_o;
394     }
395
396     //change position
397     if (view_h < te)
398     {
399         view_h++;
400         View_Grid.Margin = new Thickness(0, view_h, 0, 0);
401     }
402     else
403     {
404         //view_Grid.Visibility = Visibility.Collapsed;
405         reset();
406         (sender as DispatcherTimer).Stop();
407     }
408 }
409
410 private void help_back(object sender, RoutedEventArgs e)
411 {
412
413
414     //hide help pane
415     DispatcherTimer hide_help = new DispatcherTimer();
416     hide_help.Tick += hide_help_Tick;
417     hide_help.Interval = new TimeSpan(0, 0, 0, 0, 10);
418     hide_help.Start();
419     //fadeout homeblur
420     DispatcherTimer fadeout_home = new DispatcherTimer();
421     fadeout_home.Tick += fadeout_home_Tick;
422     fadeout_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
423     fadeout_home.Start();
424
425 }
426
427
428 private void help_MouseClick(object sender, RoutedEventArgs e)
429 {
430     blurFront.Visibility = Visibility.Visible;
431     //init
432     Help_Grid.Margin = new Thickness(0, te, 0, -te);
433     Help_Grid.Opacity = 0;
434     Help_Grid.Visibility = Visibility.Visible;
435     //fadein homeblur
436     DispatcherTimer fadein_home = new DispatcherTimer();
437     fadein_home.Tick += fadein_home_Tick;
438     fadein_home.Interval = new TimeSpan(0, 0, 0, 0, 10);
439     fadein_home.Start();
440
441     DispatcherTimer show_help = new DispatcherTimer();
442     show_help.Tick += show_help_Tick;
443     show_help.Interval = new TimeSpan(0, 0, 0, 0, 10);
444     show_help.Start();
445 }
```



```

446     int help_h = 20;
447     double help_o = 0;
448     private void show_help_Tick(object sender, EventArgs e)
449     {
450         //change alhpa
451         if (help_o < .9)
452         {
453             help_o += .05;
454             Help_Grid.Opacity = help_o;
455         }
456         else
457         {
458             Help_Grid.Opacity = 1;
459             (sender as DispatcherTimer).Stop();
460         }
461         //change position
462         if (help_h > 0)
463         {
464             help_h--;
465             Help_Grid.Margin = new Thickness(0, help_h, 0, 0);
466         }
467         else
468         {
469         }
470     }
471 }
472
473
474 private void hide_help_Tick(object sender, EventArgs e)
475 {
476     //change alhpa
477     if (help_o > 0)
478     {
479         help_o -= .05;
480         Help_Grid.Opacity = help_o;
481     }
482
483     //change position
484     if (help_h < te)
485     {
486         help_h++;
487         Help_Grid.Margin = new Thickness(0, help_h, 0, 0);
488     }
489     else
490     {
491         //help_Grid.Visibility = Visibility.Collapsed;
492         reset();
493         (sender as DispatcherTimer).Stop();
494     }
495 }
496 //new customer
497
498
499 private void new_customer_save_click(object sender, RoutedEventArgs e)
500 {
501     //validate info
502     ... ..

```

```

503     if (alpha.IsMatch(new_customer_tb_1.Text) == false)
504     {
505         nc_error.Content = "*First Name has invalid characters";
506         nc_error.Visibility = Visibility.Visible;
507     }
508     else if (alpha.IsMatch(new_customer_tb_2.Text) == false)
509     {
510         nc_error.Content = "*Last Name has invalid characters";
511         nc_error.Visibility = Visibility.Visible;
512     }
513     else if (beta.IsMatch(new_customer_tb_3.Text) == false)
514     {
515         nc_error.Content = "*Age has invalid characters";
516         nc_error.Visibility = Visibility.Visible;
517     }
518     else if (nc_rb_1.IsChecked == false && nc_rb_2.IsChecked == false)
519     {
520         nc_error.Content = "*Please select a gender";
521         nc_error.Visibility = Visibility.Visible;
522     }
523     else
524     {
525         //save customer to server
526         SqlConnection connect = new SqlConnection(getconnected());
527         SqlCommand command = new SqlCommand("INSERT into fecCustomer
528                                     (First, Last, Age, Gender, Attendance) VALUES (@First, @Last,
529                                     @Age, @Gender, @Attendance)");
530         try
531         {
532             command.Connection = connect;
533             command.CommandType = CommandType.Text;
534             connect.Open();
535             command.Parameters.Add("@First", SqlDbType.VarChar);
536             command.Parameters["@First"].Value = new_customer_tb_1.Text;
537             command.Parameters.Add("@Last", SqlDbType.VarChar);
538             command.Parameters["@Last"].Value = new_customer_tb_2.Text;
539             command.Parameters.Add("@Age", SqlDbType.VarChar);
540             command.Parameters["@Age"].Value = new_customer_tb_3.Text;
541             command.Parameters.Add("@Gender", SqlDbType.VarChar);
542             if (nc_rb_1.IsChecked == true)
543             {
544                 command.Parameters["@Gender"].Value = "male";
545             }
546             else
547             {
548                 command.Parameters["@Gender"].Value = "female";
549             }
550             command.Parameters.Add("@Attendance", SqlDbType.VarChar);
551             command.Parameters["@Attendance"].Value = "0";
552             command.ExecuteNonQuery();
553             connect.Close();
554             command.Dispose();
555             connect.Dispose();

```

```

556         new_customer_tb_1.Clear();
557         new_customer_tb_2.Clear();
558         new_customer_tb_3.Clear();
559         nc_rb_1.IsChecked = false;
560         nc_rb_2.IsChecked = false;
561
562         //go back to home
563         //./HIDE New_Customer_Grid
564         new_customer_back(null, null);
565
566     }
567
568     catch (Exception el)
569     {
570         connect.Close();
571         command.Dispose();
572         connect.Dispose();
573         //connection failed
574         nc_error.Content = "Could not connect to server";
575         nc_error.Visibility = Visibility.Visible;
576         Console.WriteLine(el.StackTrace);
577     }
578 }
579
580
581 private void new_customer_back(object sender, RoutedEventArgs e)
582 {
583
584
585     //hide new_customer pane
586     DispatcherTimer hide_new_customer = new DispatcherTimer();
587     hide_new_customer.Tick += hide_new_customer_Tick;
588     hide_new_customer.Interval = new TimeSpan(0, 0, 0, 0, 10);
589     hide_new_customer.Start();
590     //no need for fadeout blur
591     main_new_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
592 }
593
594
595 private void new_customer_MouseClick(object sender, RoutedEventArgs e)
596 {
597     //fade out New grid
598
599     new_back_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
600     Thread.Sleep(200);
601     blurFront.Visibility = Visibility.Visible;
602
603     //init
604     New_Customer_Grid.Margin = new Thickness(0, te, 0, -te);
605     New_Customer_Grid.Opacity = 0;
606     New_Customer_Grid.Visibility = Visibility.Visible;
607     blurFront.Opacity = 1;
608     blurFront.Visibility = Visibility.Visible;
609
610
611

```

```
612         show_new_customer.Tick += show_new_customer_Tick;
613         show_new_customer.Interval = new TimeSpan(0, 0, 0, 0, 10);
614         show_new_customer.Start();
615     }
616     int new_customer_h = 20;
617     double new_customer_o = 0;
618     private void show_new_customer_Tick(object sender, EventArgs e)
619     {
620         //change alhpa
621         if (new_customer_o < .9)
622         {
623             new_customer_o += .05;
624             New_Customer_Grid.Opacity = new_customer_o;
625         }
626         else
627         {
628             New_Customer_Grid.Opacity = 1;
629             (sender as DispatcherTimer).Stop();
630         }
631         //change position
632         if (new_customer_h > 0)
633         {
634             new_customer_h--;
635             New_Customer_Grid.Margin = new Thickness(0, new_customer_h, 0,
636             0);
637         }
638         else
639         {
640         }
641     }
642 }
643
644 private void hide_new_customer_Tick(object sender, EventArgs e)
645 {
646     //change alhpa
647     if (new_customer_o > 0)
648     {
649         new_customer_o -= .05;
650         New_Customer_Grid.Opacity = new_customer_o;
651     }
652     //change position
653     if (new_customer_h < te)
654     {
655         new_customer_h++;
656         New_Customer_Grid.Margin = new Thickness(0, new_customer_h, 0,
657         0);
658     }
659     else
660     {
661         //new_customer_Grid.Visibility = Visibility.Collapsed;
662         reset2();
663         (sender as DispatcherTimer).Stop();
664     }
665 }
```

```

666
667     /**(*(*(*(*
668     private void new_employee_save_click(object sender, EventArgs e)
669     {
670         //validate info
671         ne_error.Visibility = Visibility.Collapsed;
672         if (alpha.IsMatch(new_employee_tb_1.Text) == false)
673         {
674             ne_error.Content = "*First Name has invalid characters";
675             ne_error.Visibility = Visibility.Visible;
676         }
677         else if (alpha.IsMatch(new_employee_tb_2.Text) == false)
678         {
679             ne_error.Content = "*Last Name has invalid characters";
680             ne_error.Visibility = Visibility.Visible;
681         }
682         else if (beta.IsMatch(new_employee_tb_3.Text) == false)
683         {
684             ne_error.Content = "*Age has invalid characters";
685             ne_error.Visibility = Visibility.Visible;
686         }
687         else
688         {
689             //save employee to server
690             SqlConnection connect = new SqlConnection(getconnected());
691             SqlCommand command = new SqlCommand("INSERT into fecEmployee
692                                     (First, Last, Age, Shifts) VALUES (@First, @Last, @Age,
693                                     @Shifts)");
694             try
695             {
696                 command.Connection = connect;
697                 command.CommandType = CommandType.Text;
698                 connect.Open();
699                 command.Parameters.Add("@First", SqlDbType.VarChar);
700                 command.Parameters["@First"].Value = new_employee_tb_1.Text;
701                 command.Parameters.Add("@Last", SqlDbType.VarChar);
702                 command.Parameters["@Last"].Value = new_employee_tb_2.Text;
703                 command.Parameters.Add("@Age", SqlDbType.VarChar);
704                 command.Parameters["@Age"].Value = new_employee_tb_3.Text;
705                 command.Parameters.Add("@Shifts", SqlDbType.VarChar);
706                 string shifty = "";
707                 if (ne_rb_1.IsChecked == true)
708                     shifty += "-Monday-";
709                 else
710                     shifty += "-.-";
711                 if (ne_rb_2.IsChecked == true)
712                     shifty += "-Tuesday-";
713                 else
714                     shifty += "-.-";
715                 if (ne_rb_3.IsChecked == true)
716                     shifty += "-Wednesday-";
717                 else
718                     shifty += "-.-";
719                 if (ne_rb_4.IsChecked == true)

```

```
718         shifty += "-Thursday-";
719     else
720         shifty += "-.-";
721     if (ne_rb_5.IsChecked == true)
722         shifty += "-Friday-";
723     else
724         shifty += "-.-";
725     if (ne_rb_6.IsChecked == true)
726         shifty += "-Saturday-";
727     else
728         shifty += "-.-";
729     command.Parameters["@Shifts"].Value = shifty;
730
731     command.ExecuteNonQuery();
732     connect.Close();
733     command.Dispose();
734     connect.Dispose();
735
736     //reset fields
737     new_employee_tb_1.Clear();
738     new_employee_tb_2.Clear();
739     new_employee_tb_3.Clear();
740     ne_rb_1.IsChecked = false;
741     ne_rb_2.IsChecked = false;
742     ne_rb_3.IsChecked = false;
743     ne_rb_4.IsChecked = false;
744     ne_rb_5.IsChecked = false;
745     ne_rb_6.IsChecked = false;
746
747
748     //go back to home
749     //./HIDE New_Employee_Grid
750     new_employee_back(null, null);
751
752 }
753
754 catch (Exception el)
755 {
756     connect.Close();
757     command.Dispose();
758     connect.Dispose();
759     //connection failed
760     ne_error.Content = "Could not connect to server";
761     ne_error.Visibility = Visibility.Visible;
762     Console.WriteLine(el.StackTrace);
763 }
764
765 }
766 private void new_employee_back(object sender, RoutedEventArgs e)
767 {
768
769     //hide new_employee pane
770     DispatcherTimer hide_new_employee = new DispatcherTimer();
771
```

```

772         hide_new_employee.Tick += hide_new_employee_Tick;
773         hide_new_employee.Interval = new TimeSpan(0, 0, 0, 0, 10);
774         hide_new_employee.Start();
775         //no need for fadeout blur
776         main_new_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
777     }
778
779
780     private void new_employee_MouseClick(object sender, RoutedEventArgs e)
781     {
782         //fade out New grid
783
784         new_back_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
785         Thread.Sleep(200);
786         blurFront.Visibility = Visibility.Visible;
787
788         //init
789         New_Employee_Grid.Margin = new Thickness(0, te, 0, -te);
790         New_Employee_Grid.Opacity = 0;
791         New_Employee_Grid.Visibility = Visibility.Visible;
792         blurFront.Opacity = 1;
793         blurFront.Visibility = Visibility.Visible;
794
795
796         DispatcherTimer show_new_employee = new DispatcherTimer();
797         show_new_employee.Tick += show_new_employee_Tick;
798         show_new_employee.Interval = new TimeSpan(0, 0, 0, 0, 10);
799         show_new_employee.Start();
800     }
801     int new_employee_h = 20;
802     double new_employee_o = 0;
803     private void show_new_employee_Tick(object sender, EventArgs e)
804     {
805         //change alhpa
806         if (new_employee_o < .9)
807         {
808             new_employee_o += .05;
809             New_Employee_Grid.Opacity = new_employee_o;
810         }
811         else
812         {
813             New_Employee_Grid.Opacity = 1;
814             (sender as DispatcherTimer).Stop();
815         }
816         //change position
817         if (new_employee_h > 0)
818         {
819             new_employee_h--;
820             New_Employee_Grid.Margin = new Thickness(0, new_employee_h, 0,
821                 0);
822         }
823         else
824         {
825         }
826     }

```

```

827 }
828
829 private void hide_new_employee_Tick(object sender, EventArgs e)
830 {
831     //change alhpa
832     if (new_employee_o > 0)
833     {
834         new_employee_o -= .05;
835         New_Employee_Grid.Opacity = new_employee_o;
836     }
837
838     //change position
839     if (new_employee_h < te)
840     {
841         new_employee_h++;
842         New_Employee_Grid.Margin = new Thickness(0, new_employee_h, 0, 0);
843     }
844     else
845     {
846         //new_employee_Grid.Visibility = Visibility.Collapsed;
847         reset2();
848         (sender as DispatcherTimer).Stop();
849     }
850 }
851 //()()()()()()()()()()()()()()
852 private void edit_customer_plus_click(object sender, EventArgs e)
853 {
854     int num = Convert.ToInt32(edit_customer_attCount.Content);
855     Console.WriteLine(num.ToString() + "!");
856     num++;
857     edit_customer_attCount.Content = Convert.ToString(num);
858 }
859 private void edit_customer_minus_click(object sender, EventArgs e)
860 {
861     int num = Convert.ToInt32(edit_customer_attCount.Content);
862     Console.WriteLine(num.ToString() + "!");
863     num--;
864     if (num >= 0)
865     {
866         edit_customer_attCount.Content = Convert.ToString(num);
867     }
868     else
869     {
870         edit_customer_attCount.Content = "0";
871     }
872 }
873
874 private void edit_customer_find_click(object sender, EventArgs e)
875 {
876     if (ec_cover.Visibility != Visibility.Visible)
877     {
878         //validate info
879         ne_error.Visibility = Visibility.Collapsed;
880         if (alpha.IsMatch(edit_customer_tb_1.Text) == false)
881         {

```



```

882         ec_error.Content = "*First Name has invalid characters";
883         ec_error.Visibility = Visibility.Visible;
884     }
885     else if (alpha.IsMatch(edit_customer_tb_2.Text) == false)
886     {
887         ec_error.Content = "*Last Name has invalid characters";
888         ec_error.Visibility = Visibility.Visible;
889     }
890     else
891     {
892         //check if customer exists
893         SqlConnection connect = new SqlConnection(getconnected());
894         SqlCommand command = new SqlCommand("SELECT First, Last,
895                                         Age, Attendance FROM fecCustomer WHERE First = @First AND
896                                         Last = @Last");
897         try
898         {
899             command.Connection = connect;
900             command.CommandType = CommandType.Text;
901             connect.Open();
902             command.Parameters.Add("@First", SqlDbType.VarChar);
903             command.Parameters["@First"].Value =
904                 edit_customer_tb_1.Text;
905             command.Parameters.Add("@Last", SqlDbType.VarChar);
906             command.Parameters["@Last"].Value =
907                 edit_customer_tb_2.Text;
908
909             SqlDataReader sqlReader = command.ExecuteReader();
910             if (sqlReader.HasRows == true)
911             {
912                 //we got a result
913                 ec_error.Content = "Customer Found";
914                 ec_error.Visibility = Visibility.Visible;
915                 while(sqlReader.Read())
916                 {
917                     edit_customer_tb_3.Text = sqlReader.GetString
918                     (2);
919                     if (Convert.ToInt32(sqlReader.GetString(3)) > 0)
920                     {
921                         edit_customer_attCount.Content =
922                         sqlReader.GetString(3);
923                     }
924                     else
925                     {
926                         edit_customer_attCount.Content = "0";
927                     }
928                     break;
929                 }
930                 ec_cover.Visibility = Visibility.Visible;
931                 edit_customer_save_btn.Content = "Save";
932             }
933         }
934         else

```

```

931     {
932         //no result
933         ec_error.Content = "*Customer not found";
934         ec_error.Visibility = Visibility.Visible;
935     }
936
937     connect.Close();
938     command.Dispose();
939     connect.Dispose();
940
941
942
943     //go back to home
944     //./HIDE Edit_Customer_Grid
945     // edit_customer_back(null, null);
946
947
948 }
949 catch (Exception el)
950 {
951     connect.Close();
952     command.Dispose();
953     connect.Dispose();
954     //connection failed
955     ec_error.Content = "Could not connect to server";
956     ec_error.Visibility = Visibility.Visible;
957     Console.WriteLine(el.StackTrace);
958 }
959 }
960 }
961 else
962 {
963     //editing time!
964     //validate info
965     ec_error.Visibility = Visibility.Collapsed;
966     if (alpha.IsMatch(edit_customer_tb_1.Text) == false)
967     {
968         nc_error.Content = "*First Name has invalid characters";
969         nc_error.Visibility = Visibility.Visible;
970     }
971     else if (alpha.IsMatch(edit_customer_tb_2.Text) == false)
972     {
973         nc_error.Content = "*Last Name has invalid characters";
974         nc_error.Visibility = Visibility.Visible;
975     }
976     else if (beta.IsMatch(edit_customer_tb_3.Text) == false)
977     {
978         nc_error.Content = "*Age has invalid characters";
979         nc_error.Visibility = Visibility.Visible;
980     }
981     else
982     {
983         //save customer to server
984         SqlConnection connect = new SqlConnection(getconnected());
985         SqlCommand command = new SqlCommand("UPDATE fecCustomer SET
          First=@First Last=@Last Age=@Age Attendance=@Attendance

```

```

WHERE First = @First AND Last = @Last");
986 try
987 {
988     command.Connection = connect;
989     command.CommandType = CommandType.Text;
990     connect.Open();
991     command.Parameters.Add("@First", SqlDbType.VarChar);
992     command.Parameters["@First"].Value =
edit_customer_tb_1.Text;
993     command.Parameters.Add("@Last", SqlDbType.VarChar);
994     command.Parameters["@Last"].Value =
edit_customer_tb_2.Text;
995     command.Parameters.Add("@Age", SqlDbType.VarChar);
996     command.Parameters["@Age"].Value =
edit_customer_tb_3.Text;
997
998     command.Parameters.Add("@Attendance",
SqlDbType.VarChar);
999     command.Parameters["@Attendance"].Value =
Convert.ToString(edit_customer_attCount.Content);
1000     command.ExecuteNonQuery();
1001     connect.Close();
1002     command.Dispose();
1003     connect.Dispose();
1004
1005     //reset fields
1006     edit_customer_tb_1.Clear();
1007     edit_customer_tb_2.Clear();
1008     edit_customer_tb_3.Clear();
1009     edit_customer_save_btn.Content = "Find";
1010
1011     //go back to home
1012     //./HIDE edit_Customer_Grid
1013     edit_customer_back(null, null);
1014
1015 }
1016 catch (Exception el)
1017 {
1018     connect.Close();
1019     command.Dispose();
1020     connect.Dispose();
1021     //connection failed
1022     ec_error.Content = "Could not connect to server";
1023     ec_error.Visibility = Visibility.Visible;
1024     Console.WriteLine(el.StackTrace);
1025 }
1026 }
1027 }
1028 }
1029 }
1030 }
1031 private void edit_customer_back(object sender, RoutedEventArgs e)
1032 {
1033     ec_error.Visibility = Visibility.Collapsed;

```

```

1034         if (ec_cover.Visibility != Visibility.Visible)
1035         {
1036             //hide edit_customer pane
1037             DispatcherTimer hide_edit_customer = new DispatcherTimer();
1038             hide_edit_customer.Tick += hide_edit_customer_Tick;
1039             hide_edit_customer.Interval = new TimeSpan(0, 0, 0, 0, 10);
1040             hide_edit_customer.Start();
1041             //no need for fadeout blur
1042             main_edit_btn.RaiseEvent(new RoutedEventArgs
1043                 (Button.ClickEvent));
1044         }
1045         else
1046         {
1047             //dont exit pane, but reset to std edit view
1048             edit_customer_save_btn.Content = "Find";
1049             ec_cover.Visibility = Visibility.Collapsed;
1050             edit_customer_tb_1.Clear();
1051             edit_customer_tb_2.Clear();
1052         }
1053     }
1054
1055     private void edit_customer_MouseClick(object sender, RoutedEventArgs e)
1056     {
1057         //fade out New grid
1058         ec_cover.Visibility = Visibility.Collapsed;
1059         edit_customer_tb_1.Clear();
1060         edit_customer_tb_2.Clear();
1061         edit_customer_tb_3.Clear();
1062         edit_customer_save_btn.Content = "Find";
1063
1064         edit_back_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
1065
1066         blurFront.Visibility = Visibility.Visible;
1067
1068         //init
1069         Edit_Customer_Grid.Margin = new Thickness(0, te, 0, -te);
1070         Edit_Customer_Grid.Opacity = 0;
1071         Edit_Customer_Grid.Visibility = Visibility.Visible;
1072         blurFront.Opacity = 1;
1073         blurFront.Visibility = Visibility.Visible;
1074
1075         DispatcherTimer show_edit_customer = new DispatcherTimer();
1076         show_edit_customer.Tick += show_edit_customer_Tick;
1077         show_edit_customer.Interval = new TimeSpan(0, 0, 0, 0, 10);
1078         show_edit_customer.Start();
1079
1080     }
1081     int edit_customer_h = 20;
1082     double edit_customer_o = 0;
1083     private void show_edit_customer_Tick(object sender, EventArgs e)
1084     {
1085         //change alhpa
1086         if (edit_customer_o < 9)
1087

```

```

1088     {
1089         edit_customer_o += .05;
1090         Edit_Customer_Grid.Opacity = edit_customer_o;
1091     }
1092     else
1093     {
1094         Edit_Customer_Grid.Opacity = 1;
1095         (sender as DispatcherTimer).Stop();
1096     }
1097     //change position
1098     if (edit_customer_h > 0)
1099     {
1100         edit_customer_h--;
1101         Edit_Customer_Grid.Margin = new Thickness(0, edit_customer_h, 0, 0);
1102     }
1103     else
1104     {
1105     }
1106 }
1107
1108
1109
1110 private void hide_edit_customer_Tick(object sender, EventArgs e)
1111 {
1112     //change alhpa
1113     if (edit_customer_o > 0)
1114     {
1115         edit_customer_o -= .05;
1116         Edit_Customer_Grid.Opacity = edit_customer_o;
1117     }
1118
1119     //change position
1120     if (edit_customer_h < te)
1121     {
1122         edit_customer_h++;
1123         Edit_Customer_Grid.Margin = new Thickness(0, edit_customer_h, 0, 0);
1124     }
1125     else
1126     {
1127         //edit_customer_Grid.Visibility = Visibility.Collapsed;
1128         reset2();
1129         (sender as DispatcherTimer).Stop();
1130     }
1131 }
1132
1133 /**
1134 //EDIT EMPLOYEE
1135 /**
1136
1137 private void edit_employee_find_click(object sender, EventArgs e)
1138 {
1139     if (ee_cover.Visibility != Visibility.Visible)
1140     {
1141         //validate info
1142         ee_error.Visibility = Visibility.Collapsed;

```

```

1143     if (alpha.IsMatch(edit_employee_tb_1.Text) == false)
1144     {
1145         ee_error.Content = "*First Name has invalid characters";
1146         ee_error.Visibility = Visibility.Visible;
1147     }
1148     else if (alpha.IsMatch(edit_employee_tb_2.Text) == false)
1149     {
1150         ee_error.Content = "*Last Name has invalid characters";
1151         ee_error.Visibility = Visibility.Visible;
1152     }
1153     else
1154     {
1155         //check if customer exists
1156         SqlConnection connect = new SqlConnection(getconnected());
1157         SqlCommand command = new SqlCommand("SELECT First, Last,
Age, Shifts FROM fecEmployee WHERE First = @First AND Last =
@Last");
1158         try
1159         {
1160             command.Connection = connect;
1161             command.CommandType = CommandType.Text;
1162             connect.Open();
1163             command.Parameters.Add("@First", SqlDbType.VarChar);
1164             command.Parameters["@First"].Value =
edit_employee_tb_1.Text;
1165             command.Parameters.Add("@Last", SqlDbType.VarChar);
1166             command.Parameters["@Last"].Value =
edit_employee_tb_2.Text;
1167
1168             SqlDataReader sqlReader = command.ExecuteReader();
1169             if (sqlReader.HasRows == true)
1170             {
1171                 //we got a result
1172                 ee_error.Content = "Employee Found";
1173                 ee_error.Visibility = Visibility.Visible;
1174                 while (sqlReader.Read())
1175                 {
1176                     edit_employee_tb_3.Text = Convert.ToString
(sqlReader.GetValue(2));
1177                     //take shifts and parse
1178                     String[] sifts = sqlReader.GetString(3).Split
('-');
1179                     sifts = sifts.Where(x => !string.IsNullOrEmpty
(x)).ToArray();
1180                     if (sifts[0] != ".")
1181                         ee_rb_1.IsChecked = true;
1182                     if (sifts[1] != ".")
1183                         ee_rb_2.IsChecked = true;
1184                     if (sifts[2] != ".")
1185                         ee_rb_3.IsChecked = true;
1186                     if (sifts[3] != ".")
1187                         ee_rb_4.IsChecked = true;
1188                     if (sifts[4] != ".")
1189

```

```
1190         if (sifts[5] != ".")
1191             ee_rb_6.IsChecked = true;
1192
1193         break;
1194     }
1195     ee_cover.Visibility = Visibility.Visible;
1196     edit_employee_save_btn.Content = "Save";
1197
1198
1199
1200
1201     }
1202     else
1203     {
1204         //no result
1205         ee_error.Content = "*Employee not found";
1206         ee_error.Visibility = Visibility.Visible;
1207     }
1208
1209     connect.Close();
1210     command.Dispose();
1211     connect.Dispose();
1212
1213
1214
1215     //go back to home
1216     //./HIDE Edit_Employee_Grid
1217     // edit_employee_back(null, null);
1218
1219
1220     }
1221     catch (Exception el)
1222     {
1223         connect.Close();
1224         command.Dispose();
1225         connect.Dispose();
1226         //connection failed
1227         ee_error.Content = "Could not connect to server";
1228         ee_error.Visibility = Visibility.Visible;
1229         Console.WriteLine(el.StackTrace);
1230     }
1231 }
1232 }
1233 else
1234 {
1235     //editing time!
1236     //validate info
1237     ee_error.Visibility = Visibility.Collapsed;
1238     if (alpha.IsMatch(edit_employee_tb_1.Text) == false)
1239     {
1240         ee_error.Content = "*First Name has invalid characters";
1241         ee_error.Visibility = Visibility.Visible;
1242     }
1243     else if (alpha.IsMatch(edit_employee_tb_2.Text) == false)
1244     {
1245         ee_error.Content = "*Last Name has invalid characters";
```

```

1246         ee_error.Visibility = Visibility.Visible;
1247     }
1248     else if (beta.IsMatch(edit_employee_tb_3.Text) == false)
1249     {
1250         ee_error.Content = "*Age has invalid characters";
1251         ee_error.Visibility = Visibility.Visible;
1252     }
1253     else
1254     {
1255         //save customer to server
1256         SqlConnection connect = new SqlConnection(getconnected());
1257         SqlCommand command = new SqlCommand("UPDATE fecEmployee SET
First=@First, Last=@Last, Age=@Age, Shifts=@Shifts WHERE
First = @First AND Last = @Last");
1258         try
1259         {
1260             command.Connection = connect;
1261             command.CommandType = CommandType.Text;
1262             connect.Open();
1263             command.Parameters.Add("@First", SqlDbType.VarChar);
1264             command.Parameters["@First"].Value =
edit_employee_tb_1.Text;
1265             command.Parameters.Add("@Last", SqlDbType.VarChar);
1266             command.Parameters["@Last"].Value =
edit_employee_tb_2.Text;
1267             command.Parameters.Add("@Age", SqlDbType.VarChar);
1268             command.Parameters["@Age"].Value =
edit_employee_tb_3.Text;
1269             string shifty = "";
1270             if (ee_rb_1.IsChecked == true)
1271                 shifty += "-Monday-";
1272             else
1273                 shifty += "-.-";
1274             if (ee_rb_2.IsChecked == true)
1275                 shifty += "-Tuesday-";
1276             else
1277                 shifty += "-.-";
1278             if (ee_rb_3.IsChecked == true)
1279                 shifty += "-Wednesday-";
1280             else
1281                 shifty += "-.-";
1282             if (ee_rb_4.IsChecked == true)
1283                 shifty += "-Thursday-";
1284             else
1285                 shifty += "-.-";
1286             if (ee_rb_5.IsChecked == true)
1287                 shifty += "-Friday-";
1288             else
1289                 shifty += "-.-";
1290             if (ee_rb_6.IsChecked == true)
1291                 shifty += "-Saturday-";
1292             else
1293                 shifty += "-.-";

```



```

1294         command.Parameters.Add("@Shifts", SqlDbType.VarChar);
1295         command.Parameters["@Shifts"].Value = shift;
1296         command.ExecuteNonQuery();
1297         connect.Close();
1298         command.Dispose();
1299         connect.Dispose();
1300
1301         //reset fields
1302         edit_employee_tb_1.Clear();
1303         edit_employee_tb_2.Clear();
1304         edit_employee_tb_3.Clear();
1305         ee_rb_1.IsChecked = false;
1306         ee_rb_2.IsChecked = false;
1307         ee_rb_3.IsChecked = false;
1308         ee_rb_4.IsChecked = false;
1309         ee_rb_5.IsChecked = false;
1310         ee_rb_6.IsChecked = false;
1311         ec_cover.Visibility = Visibility.Collapsed;
1312         edit_employee_save_btn.Content = "Find";
1313
1314         //go back to home
1315         //./HIDE edit_Customer_Grid
1316         edit_employee_back(null, null);
1317
1318     }
1319
1320     catch (Exception el)
1321     {
1322         connect.Close();
1323         command.Dispose();
1324         connect.Dispose();
1325         //connection failed
1326         ee_error.Content = "Could not connect to server";
1327         ee_error.Visibility = Visibility.Visible;
1328         Console.WriteLine(el.StackTrace);
1329     }
1330 }
1331
1332 }
1333
1334 private void edit_employee_back(object sender, RoutedEventArgs e)
1335 {
1336     ee_error.Visibility = Visibility.Collapsed;
1337     if (ec_cover.Visibility != Visibility.Visible)
1338     {
1339
1340         //hide edit_employee pane
1341         DispatcherTimer hide_edit_employee = new DispatcherTimer();
1342         hide_edit_employee.Tick += hide_edit_employee_Tick;
1343         hide_edit_employee.Interval = new TimeSpan(0, 0, 0, 0, 10);
1344         hide_edit_employee.Start();
1345         //no need for fadeout blur
1346         main_edit_btn.RaiseEvent(new RoutedEventArgs
            (Button.ClickEvent));
1347

```

```

1348     else
1349     {
1350         //dont exit pane, but reset to std edit view
1351         edit_employee_save_btn.Content = "Find";
1352         ee_cover.Visibility = Visibility.Collapsed;
1353         edit_employee_tb_1.Clear();
1354         edit_employee_tb_2.Clear();
1355     }
1356 }
1357
1358
1359 private void edit_employee_MouseClick(object sender, RoutedEventArgs e)
1360 {
1361     //fade out New grid
1362     ee_cover.Visibility = Visibility.Collapsed;
1363     edit_employee_tb_1.Clear();
1364     edit_employee_tb_2.Clear();
1365     edit_employee_tb_3.Clear();
1366     edit_employee_save_btn.Content = "Find";
1367
1368     edit_back_btn.RaiseEvent(new RoutedEventArgs(Button.ClickEvent));
1369
1370     blurFront.Visibility = Visibility.Visible;
1371
1372     //init
1373     Edit_Employee_Grid.Margin = new Thickness(0, te, 0, -te);
1374     Edit_Employee_Grid.Opacity = 0;
1375     Edit_Employee_Grid.Visibility = Visibility.Visible;
1376     blurFront.Opacity = 1;
1377     blurFront.Visibility = Visibility.Visible;
1378
1379
1380     DispatcherTimer show_edit_employee = new DispatcherTimer();
1381     show_edit_employee.Tick += show_edit_employee_Tick;
1382     show_edit_employee.Interval = new TimeSpan(0, 0, 0, 0, 10);
1383     show_edit_employee.Start();
1384 }
1385 int edit_employee_h = 20;
1386 double edit_employee_o = 0;
1387 private void show_edit_employee_Tick(object sender, EventArgs e)
1388 {
1389     //change alhpa
1390     if (edit_employee_o < .9)
1391     {
1392         edit_employee_o += .05;
1393         Edit_Employee_Grid.Opacity = edit_employee_o;
1394     }
1395     else
1396     {
1397         Edit_Employee_Grid.Opacity = 1;
1398         (sender as DispatcherTimer).Stop();
1399     }
1400     //change position
1401     if (edit_employee_h > 0)
1402     {

```

```

1403         edit_employee_h--;
1404         Edit_Employee_Grid.Margin = new Thickness(0, edit_employee_h, 0, 0);
1405     }
1406     else
1407     {
1408     }
1409 }
1410
1411 }
1412
1413 private void hide_edit_employee_Tick(object sender, EventArgs e)
1414 {
1415     //change alhpa
1416     if (edit_employee_o > 0)
1417     {
1418         edit_employee_o -= .05;
1419         Edit_Employee_Grid.Opacity = edit_employee_o;
1420     }
1421
1422     //change position
1423     if (edit_employee_h < te)
1424     {
1425         edit_employee_h++;
1426         Edit_Employee_Grid.Margin = new Thickness(0, edit_employee_h, 0, 0);
1427     }
1428     else
1429     {
1430         //edit_employee_Grid.Visibility = Visibility.Collapsed;
1431         reset2();
1432         (sender as DispatcherTimer).Stop();
1433     }
1434 }
1435
1436 //VIEW REPORT
1437 public Dictionary<string, Process> OpenedProcesses = new Dictionary<string, Process>(StringComparer.OrdinalIgnoreCase);
1438 private void view_customers_Report(object sender, RoutedEventArgs e)
1439 {
1440
1441     SqlConnection connect = new SqlConnection(getconnected());
1442     SqlCommand command = new SqlCommand("SELECT First, Last, Gender, Age, Attendance FROM fecCustomer");
1443     try
1444     {
1445         string filePath = System.Environment.CurrentDirectory + "\\customer_report.txt";
1446         command.Connection = connect;
1447         command.CommandType = CommandType.Text;
1448         connect.Open();
1449         SqlDataReader sqlReader = command.ExecuteReader();
1450         if (sqlReader.HasRows == true)
1451         {
1452             //we got results
1453             File.Delete(filePath);

```

```

1454 StreamWriter file = new StreamWriter(filePath, true);
1455 int id = 1;
1456 while (sqlReader.Read())
1457 {
1458     //take shifts and parse
1459     /*
1460     String[] sifts = sqlReader.GetString(3).Split('-');
1461     sifts = sifts.Where(x => !string.IsNullOrEmpty
1462     (x)).ToArray();
1463     sifts = sifts.Where(x => (x != ".")).ToArray();
1464     */
1465     //write to text file line
1466     file.WriteLine(id.ToString() + ": " +
1467     (sqlReader.GetValue(0)).ToString() + " " +
1468     (sqlReader.GetValue(1)).ToString() + " Gender-" +
1469     (sqlReader.GetValue(2)).ToString() + " Age-" +
1470     (sqlReader.GetValue(3)).ToString() + " Attendance-" +
1471     (sqlReader.GetValue(4)).ToString());
1472     id++;
1473 }
1474 //show text file
1475 file.Close();
1476 Process proc;
1477 if (this.OpenedProcesses.TryGetValue(filePath, out proc))
1478 {
1479     v_error.Content = "The file is already open!";
1480     v_error.Visibility = Visibility.Visible;
1481     // it could be possible to activate the window of the
1482     open process but that is another question on its own.
1483 }
1484 else
1485 {
1486     proc = System.Diagnostics.Process.Start(filePath);
1487     this.OpenedProcesses[filePath] = proc;
1488 }
1489 }
1490 else
1491 {
1492     //no result
1493     v_error.Content = "*No customers exist";
1494     v_error.Visibility = Visibility.Visible;
1495 }
1496 }
1497 connect.Close();
1498 command.Dispose();
1499 connect.Dispose();
1500
1501 //go back to home
1502 //./HIDE Edit_Employee_Grid
1503 // edit_employee_back(null, null);

```

```

1502
1503
1504     }
1505     catch (Exception e1)
1506     {
1507         connect.Close();
1508         command.Dispose();
1509         connect.Dispose();
1510         //connection failed
1511         v_error.Content = "Could not connect to server";
1512         v_error.Visibility = Visibility.Visible;
1513         Console.WriteLine(e1.StackTrace);
1514     }
1515 }
1516 private void view_employees_Report(object sender, RoutedEventArgs e)
1517 {
1518     SqlConnection connect = new SqlConnection(getconnected());
1519     SqlCommand command = new SqlCommand("SELECT First, Last, Age, Shifts
FROM fecEmployee");
1520
1521     try
1522     {
1523         string filePath = System.Environment.CurrentDirectory + "\\
\\employee_report.txt";
1524
1525         command.Connection = connect;
1526         command.CommandType = CommandType.Text;
1527         connect.Open();
1528         SqlDataReader sqlReader = command.ExecuteReader();
1529         if (sqlReader.HasRows == true)
1530         {
1531             //we got results
1532             File.Delete(filePath);
1533             StreamWriter file = new StreamWriter(filePath, true);
1534             int id = 1;
1535             while (sqlReader.Read())
1536             {
1537                 //take shifts and parse
1538
1539                 String[] sifts = sqlReader.GetString(3).Split('-');
1540                 sifts = sifts.Where(x => !string.IsNullOrEmpty
(x)).ToArray();
1541                 sifts = sifts.Where(x => (x != ".")).ToArray();
1542
1543                 //write to text file line
1544                 file.WriteLine(id.ToString() + ": " +
(sqlReader.GetValue(0)).ToString() + " " +
(sqlReader.GetValue(1)).ToString() + " Age-" +
(sqlReader.GetValue(2)).ToString() + " Shifts-" +
String.Join(", ", sifts));
1545                 id++;
1546             }
1547             //show text file
1548             file.Close();
1549             Process proc;

```

```
1549         if (this.OpenedProcesses.TryGetValue(filePath, out proc)
1550             && !proc.HasExited)
1551         {
1552             v_error.Content = "The file is already open!";
1553             v_error.Visibility = Visibility.Visible;
1554             // it could be possible to activate the window of the
1555             // open process but that is another question on its own.
1556         }
1557         else
1558         {
1559             proc = System.Diagnostics.Process.Start(filePath);
1560             this.OpenedProcesses[filePath] = proc;
1561         }
1562     else
1563     {
1564         //no result
1565         ee_error.Content = "*No Employees exist";
1566         ee_error.Visibility = Visibility.Visible;
1567     }
1568
1569     connect.Close();
1570     command.Dispose();
1571     connect.Dispose();
1572
1573
1574
1575     //go back to home
1576     //./HIDE Edit_Employee_Grid
1577     // edit_employee_back(null, null);
1578
1579 }
1580
1581 catch (Exception el)
1582 {
1583     connect.Close();
1584     command.Dispose();
1585     connect.Dispose();
1586     //connection failed
1587     ee_error.Content = "Could not connect to server";
1588     ee_error.Visibility = Visibility.Visible;
1589     Console.WriteLine(el.StackTrace);
1590 }
1591
1592 }
1593
1594 }
1595
```