

```
1 using System;
2 using System.Collections.Generic;
3 using System.Data.Entity;
4 using System.Diagnostics;
5 using System.Globalization;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows;
10 using System.Windows.Controls;
11 using System.Windows.Data;
12 using System.Windows.Documents;
13 using System.Windows.Input;
14 using System.Windows.Markup;
15 using System.Windows.Media;
16 using System.Windows.Media.Imaging;
17 using System.Windows.Navigation;
18 using System.Windows.Shapes;
19 using Infragistics.Controls.Schedules;
20 using AppProjects;
21 using System.ComponentModel;
22 using System.Collections.ObjectModel;
23
24 namespace DBProjects
25 {
26     /// <summary>
27     /// Логика взаимодействия для MainWindow.xaml
28     /// </summary>
29     public partial class MainWindow : Window
30     {
31         public ProjectDB content { get; set; }
32         public CollectionViewSource projectsViewSource { get; set; }
33         public CollectionViewSource categoriesViewSource { get; set; }
34         public CollectionViewSource companiesViewSource { get; set; }
35         public CollectionViewSource personsViewSource { get; set; }
36         public CollectionViewSource tasksViewSource { get; set; }
37         public CollectionViewSource categoriesProjectsViewSource { get;
38             set; }
39         public CollectionViewSource commentsViewSource { get; set; }
40         public CollectionViewSource listsViewSource { get; set; }
41         public CollectionViewSource viewObjectViewSource { get; set; }
42         public CollectionViewSource myTasksViewSource { get; set; }
43         public CollectionViewSource myCommentsViewSource { get; set; }
44         public Persons User { get; set; } = null;
45         public DateTime CurrentDate { get; set; } = DateTime.Now;
46         public ListBackedProjectViewModel viewmodel { get; set; }
47
48         public MainWindow()
49         {
50             InitializeComponent();
51             content = new ProjectDB();
52             projectsViewSource = ((CollectionViewSource)(this.FindResource
53                 ("projectsViewSource")));
54             categoriesViewSource = ((CollectionViewSource)(this.FindResource
55                 ("categoriesViewSource")));
56             categoriesProjectsViewSource = (CollectionViewSource)
```

```

        (FindResource("categoriesProjectsViewSource"));
54     companiesViewSource = ((CollectionViewSource)(this.FindResource
        ("companiesViewSource")));
55     personsViewSource = ((CollectionViewSource)(this.FindResource
        ("personsViewSource")));
56     tasksViewSource = ((CollectionViewSource)(this.FindResource
        ("tasksViewSource")));
57     viewObjectViewSource = ((CollectionViewSource)(this.FindResource
        ("viewObjectViewSource")));
58     commentsViewSource = ((CollectionViewSource)(this.FindResource
        ("commentsViewSource")));
59     listsViewSource = ((CollectionViewSource)(this.FindResource
        ("listsViewSource")));
60     myTasksViewSource = ((CollectionViewSource)(this.FindResource
        ("listsTasksViewSource")));
61     myCommentsViewSource = ((CollectionViewSource)(this.FindResource
        ("tasksCommentsViewSource")));
62     viewModel = new ListBackedProjectViewModel();
63     DlgLogin dlgLogin = null;
64     while (true)
65     {
66         dlgLogin = null;
67         dlgLogin = new DlgLogin(content);
68         dlgLogin.BoxDate.SelectedDate = CurrentDate;
69         if (dlgLogin.ShowDialog() == false)
70         {
71             Close();
72             return;
73         }
74         if(dlgLogin.BoxUser.SelectedValue != null)
75         {
76             if (((Persons)(dlgLogin.BoxUser.SelectedItem)).Pasword ==
                null)
77             {
78                 if (dlgLogin.BoxPassword.Password == "") break;
79             }
80             else if (((Persons)
                (dlgLogin.BoxUser.SelectedItem)).Pasword ==
                dlgLogin.BoxPassword.Password) break;
81         }
82         MessageBox.Show("Не верный пароль", "Ошибка авторизации",
            MessageBoxButton.OK, MessageBoxImage.Warning);
83     }
84     User = (from c in content.Persons where c.id == ((Persons)
        (dlgLogin.BoxUser.SelectedItem)).id select c).First();
85     CurrentDate = (DateTime)dlgLogin.BoxDate.SelectedDate;
86     MenuUser.Header = User.Email;
87 }
88
89 private void Window_Loaded(object sender, RoutedEventArgs e)
90 {
91     Company_ViewUpdate();
92     Project_ViewUpdate();
93     List_ViewUpdate();
94     Task_ViewUpdate();
95     MyTask_ViewUpdate();

```

```
196         MyComment_ViewUpdate();
197         Company_BoxIndustry.ItemsSource = IndustryQuery.ToList();
198         Company_BoxCity.ItemsSource = CityQuery.ToList();
199         MyTask_BoxOwner.ItemsSource = content.Persons.ToList();
200         Task_BoxOwner.ItemsSource = content.Persons.ToList();
201         Company_BoxOwner.ItemsSource = content.Persons.ToList();
202         List_BoxOwner.ItemsSource = content.Persons.ToList();
203         categoriesViewSource.Source = content.Categories.ToList();
204         personsViewSource.Source = content.Persons.ToList();
205         viewModel.DownloadDataSource(MyTaskQuery);
206         gantt.Project.DataContext = viewModel;
207     }
208
209     void Refresh(FrameworkElement item)
210     {
211         var buf = item.DataContext;
212         item.DataContext = item;
213         item.DataContext = buf;
214     }
215
216     private void MenuAddCategory_Click(object sender, RoutedEventArgs e)
217     {
218         DlgCategory winAddCategory = new DlgCategory(content);
219         if (winAddCategory.ShowDialog() == false) return;
220     }
221
222     private void ButtonEdit_Click(object sender, RoutedEventArgs e)
223     {
224         DlgCategory winAddCategory = new DlgCategory(content);
225         if (winAddCategory.ShowDialog() == false) return;
226     }
227
228
229     private void MenuAddPerson_Click(object sender, RoutedEventArgs e)
230     {
231         if (User.IsAdmin == null)
232         {
233             MessageBox.Show("Этот объект могут изменить только администратор", "Не достаточно прав для операции",
234                             MessageBoxButton.OKCancel, MessageBoxImage.Warning);
235             return;
236         }
237         DlgPerson dlg = new DlgPerson(content);
238         if (dlg.ShowDialog() == false) return;
239         MyTask_BoxOwner.ItemsSource = content.Persons.ToList();
240         Task_BoxOwner.ItemsSource = content.Persons.ToList();
241         Company_BoxOwner.ItemsSource = content.Persons.ToList();
242         List_BoxOwner.ItemsSource = content.Persons.ToList();
243         categoriesViewSource.Source = content.Categories.ToList();
244     }
245
246     private void MenuAddProject_Click(object sender, RoutedEventArgs e)
247     {
248         DlgProject dlg = new DlgProject(content);
249         if (dlg.ShowDialog() == false) return;
250     }
```

```

150
151     private void MenuAddTask_Click(object sender, RoutedEventArgs e)
152     {
153         DlgTask dlg = new DlgTask(content);
154         if (dlg.ShowDialog() == false) return;
155     }
156
157     private void MenuAddList_Click(object sender, RoutedEventArgs e)
158     {
159         DlgList dlg = new DlgList(content);
160         if (dlg.ShowDialog() == false) return;
161         Company_ViewUpdate();
162     }
163
164     private void MenuAddComment_Click(object sender, RoutedEventArgs e)
165     {
166         DlgComments dlg = new DlgComments(content);
167         if (dlg.ShowDialog() == false) return;
168     }
169
170     private void MenuAddRole_Click(object sender, RoutedEventArgs e)
171     {
172         DlgRole dlg = new DlgRole(content);
173         if (dlg.ShowDialog() == false) return;
174     }
175
176
177     IQueryable<Companies> CompanyQuery
178     {
179         get
180         {
181             var query = from c in content.Companies select c;
182             if (Company_BoxName.Text != "") query = from c in query where ↗
183                 c.Name.Contains(Company_BoxName.Text) select c;
184             if (Company_BoxIndustry.Text != "") query = from c in query ↗
185                 where c.Industry == Company_BoxIndustry.Text select c;
186             if (Company_BoxCity.Text != "") query = from c in query where ↗
187                 c.City == Company_BoxCity.Text select c;
188             if (Company_BoxPhone.Text != "") query = from c in query ↗
189                 where c.Fax.Contains(Company_BoxPhone.Text) select c;
190             if (Company_BoxNotes.Text != "") query = from c in query ↗
191                 where c.Notes.Contains(Company_BoxNotes.Text) select c;
192             if (Company_BoxCountry.Text != "") query = from c in query ↗
193                 where c.Country == Company_BoxCountry.Text select c;
194             if (Company_BoxOwner.SelectedItem != null) query = from c in ↗
195                 query where c.Id_Owner == ((Persons) ↗
196                 Company_BoxOwner.SelectedItem).id select c;
197             if (User == null) return query;
198             if (User.IsAdmin != null) return query;
199             var permission = (from c in content.ViewObject where ↗
200                 c.Id_Owner == User.id select new {Id_Company = ↗
201                 c.Id_Company }).Distinct();
202             query = (from c in query join p in permission on c.id equals ↗
203                 p.Id_Company select c).Distinct();
204             return query;
205         }
206     }

```

```

195     }
196
197     IQueryable<Tasks> MyTaskQuery
198     {
199
200         get
201         {
202             DateTime CurrentDate10 = CurrentDate.AddDays(10);
203             DateTime CurrentDate30 = CurrentDate.AddDays(30);
204             var query = from c in content.Tasks where c.Id_Owner ==      ↗
                User.id || c.Lists.Id_Owner == User.id ||      ↗
                c.Lists.Projects.Id_Owner == User.id ||      ↗
                c.Lists.Projects.Companies.Id_Owner == User.id select c;
205             if (MyTask_BoxTerm.Text == "На сегодня")
206             {
207                 query = from c in query where c.Start_Date <= CurrentDate ↗
                    select c;
208                 query = from c in query where c.Due_Date >= CurrentDate ↗
                    select c;
209             }
210             if (MyTask_BoxTerm.Text == "На 10 дней")
211             {
212                 query = from c in query where c.Start_Date <=      ↗
                    CurrentDate10 select c;
213                 query = from c in query where c.Due_Date >= CurrentDate ↗
                    select c;
214             }
215             if (MyTask_BoxTerm.Text == "На 30 дней")
216             {
217                 query = from c in query where c.Start_Date <=      ↗
                    CurrentDate30 select c;
218                 query = from c in query where c.Due_Date >= CurrentDate ↗
                    select c;
219             }
220             if (MyTask_BoxTerm.Text == "Просроченные") query = from c in ↗
                query where c.Due_Date < CurrentDate select c;
221             if (MyTask_BoxStatus.Text != "") query = from c in query where ↗
                c.Status == MyTask_BoxStatus.Text select c;
222             if (MyTask_BoxPriority.Text != "") query = from c in query ↗
                where c.Priority == MyTask_BoxPriority.Text select c;
223             if (MyTask_BoxNotes.Text != "") query = from c in query where ↗
                c.Description.Contains(MyTask_BoxNotes.Text) select c;
224             if (MyTask_BoxName.Text != "") query = from c in query where ↗
                c.Name.Contains(MyTask_BoxName.Text) select c;
225             if (MyTask_BoxOwner.SelectedValue != null) query = from c in ↗
                query where c.Id_Owner == ((Persons) ↗
                (MyTask_BoxOwner.SelectedItem)).id select c;
226             var permission = (from c in content.ViewObject where ↗
                c.Id_Owner == User.id select new { Id_Task = ↗
                c.Id_Task }).Distinct();
227             query = (from c in query join p in permission on c.Id equals ↗
                p.Id_Task select c).Distinct();
228             return query;
229         }
230     }
231

```

```

232     IQueryable<Tasks> LateTaskQuery
233     {
234         get
235         {
236             var query = from c in content.Tasks select c;
237             if (User != null) query = from c in content.Tasks where
                c.Id_Owner == User.id select c;
238             query = from c in content.Tasks where c.Status != "Выполнено"
                select c;
239             query = from c in content.Tasks where c.Due_Date <
                CurrentDate select c;
240             return query;
241         }
242     }
243
244     IQueryable<Projects> ProjectQuery
245     {
246         get
247         {
248             if (Company_DataGrid.SelectedValue == null) return from c in
                content.Projects where false select c; ;
249             var query = from c in content.Projects where c.Id_Company ==
                ((Companies)(Company_DataGrid.SelectedItem)).id select c;
250             if (Project_BoxName.Text != "") query = from c in query where
                c.Name.Contains(Project_BoxName.Text) select c;
251             if (Project_BoxNotes.Text != "") query = from c in query
                where c.Description.Contains(Project_BoxNotes.Text) select
                c;
252             if (Project_BoxYear.Text != "") query = from c in query where
                c.Year.ToString() == Project_BoxYear.Text select c;
253             if (Project_BoxCategory.SelectedItem != null) query = from c
                in query where c.Id_Category == ((Categories)
                Project_BoxCategory.SelectedItem).Id select c;
254             if (Project_BoxOwner.SelectedItem != null) query = from c in
                query where c.Id_Owner == ((Persons)
                Project_BoxOwner.SelectedItem).id select c;
255             if (User == null) return query;
256             if (User.IsAdmin != null) return query;
257             if (User.id == ((Companies)
                (Company_DataGrid.SelectedItem)).Id_Owner) return query;
258             var permission = (from c in content.ViewObject where
                c.Id_Owner == User.id select new { Id_Project =
                c.Id_Project }).Distinct();
259             query = (from c in query join p in permission on c.Id equals
                p.Id_Project select c).Distinct();
260             return query;
261         }
262     }
263
264     IQueryable<Lists> ListQuery
265     {
266         get
267         {
268             if (Project_DataGrid.SelectedValue == null) return from c in
                content.Lists where false select c;
269             var Parent = (Projects)(Project_DataGrid.SelectedItem);

```

```

270         var query = from c in content.Lists where c.Id_Project ==
                        Parent.Id select c;
271         if (List_BoxName.Text != "") query = from c in query where
                        c.Name.Contains(List_BoxName.Text) select c;
272         if (List_BoxNotes.Text != "") query = from c in query where
                        c.Note.Contains(List_BoxNotes.Text) select c;
273         if (List_BoxOwner.SelectedItem != null) query = from c in
                        query where c.Id_Owner == ((Persons)
                        List_BoxOwner.SelectedItem).id select c;
274         if (User == null) return query;
275         if (User.IsAdmin != null) return query;
276         if (User.id == Parent.Companies.Id_Owner) return query;
277         if (User.id == Parent.Id_Owner) return query;
278         var permission = (from c in content.ViewObject where
                        c.Id_Owner == User.id select new { Id_List =
                        c.Id_List }).Distinct();
279         query = (from c in query join p in permission on c.Id equals
                        p.Id_List select c).Distinct();
280         return query;
281     }
282 }
283
284 IQueryable<Comments> CommentQuery
285 {
286     get
287     {
288         if (Task_DataGrid.SelectedValue == null) return from c in
                        content.Comments where false select c;
289         var Parent = (Tasks)(Task_DataGrid.SelectedItem);
290         var query = from c in content.Comments where c.Id_Task ==
                        Parent.Id select c;
291         if (User == null) return query;
292         if (User.IsAdmin != null) return query;
293         if (User.id == Parent.Lists.Projects.Companies.Id_Owner)
                        return query;
294         if (User.id == Parent.Lists.Projects.Id_Owner) return query;
295         if (User.id == Parent.Lists.Id_Owner) return query;
296         if (User.id == Parent.Id_Owner) return query;
297         var permission = (from c in content.ViewObject where
                        c.Id_Owner == User.id select new { Id_Comment =
                        c.Id_Comment }).Distinct();
298         query = (from c in query join p in permission on c.Id equals
                        p.Id_Comment select c).Distinct();
299         return query;
300     }
301 }
302
303 IQueryable<Comments> MyCommentQuery
304 {
305     get
306     {
307         if (MyTask_DataGrid.SelectedValue == null) return from c in
                        content.Comments where false select c;
308         var query = from c in content.Comments where c.Id_Task ==
                        ((Tasks)(MyTask_DataGrid.SelectedItem)).Id select c;
309         query = from c in query where c.Id_Person == User.id ||

```

```

310         c.Tasks.Id_Owner == User.id ||
311         c.Tasks.Lists.Id_Owner == User.id ||
312         c.Tasks.Lists.Projects.Id_Owner == User.id ||
313         c.Tasks.Lists.Projects.Companies.Id_Owner == User.id
314         select c;
315     if (User == null) return query;
316     if (User.IsAdmin != null) return query;
317     var permission = (from c in content.ViewObject where
318         c.Id_Owner == User.id select new { Id_Comment =
319         c.Id_Comment }).Distinct();
320     query = (from c in query join p in permission on c.Id equals
321         p.Id_Comment select c).Distinct();
322     return query;
323 }
324 }
325
326 IQueryable<Tasks> TaskQuery
327 {
328     get
329     {
330         DateTime CurrentDate10 = CurrentDate.AddDays(10);
331         DateTime CurrentDate30 = CurrentDate.AddDays(30);
332         if (List_DataGrid.SelectedValue == null) return from c in
333             content.Tasks where false select c;
334         var query = from c in content.Tasks where c.Id_List ==
335             ((Lists)(List_DataGrid.SelectedItem)).Id select c;
336         if (Task_BoxTerm.Text == "На сегодня")
337         {
338             query = from c in query where c.Start_Date <= CurrentDate
339                 select c;
340             query = from c in query where c.Due_Date >= CurrentDate
341                 select c;
342         }
343         if (Task_BoxTerm.Text == "На 10 дней")
344         {
345             query = from c in query where c.Start_Date <=
346                 CurrentDate10 select c;
347             query = from c in query where c.Due_Date >= CurrentDate
348                 select c;
349         }
350         if (Task_BoxTerm.Text == "На 30 дней")
351         {
352             query = from c in query where c.Start_Date <=
353                 CurrentDate30 select c;
354             query = from c in query where c.Due_Date >= CurrentDate
355                 select c;
356         }
357         if (Task_BoxTerm.Text == "Просроченные")
358         {
359             query = from c in query where c.Due_Date < CurrentDate
360                 select c;
361         }
362         if (Task_BoxStatus.Text != "") query = from c in query where
363             c.Status == Task_BoxStatus.Text select c;
364         if (Task_BoxPriority.Text != "") query = from c in query
365             where c.Priority == Task_BoxPriority.Text select c;

```



```

351         if (Task_BoxNotes.Text != "") query = from c in query where
           c.Description.Contains(Task_BoxNotes.Text) select c;
352         if (Task_BoxName.Text != "") query = from c in query where
           c.Name.Contains(Task_BoxName.Text) select c;
353         if (Task_BoxOwner.SelectedItem != null) query = from c in
           query where c.Id_Owner == ((Persons)
           (Task_BoxOwner.SelectedItem)).id select c;
354         if (User == null) return query;
355         if (User.IsAdmin != null) return query;
356         var Parent = (Lists)(List_DataGrid.SelectedItem);
357         if (User.id == Parent.Projects.Companies.Id_Owner) return
           query;
358         if (User.id == Parent.Projects.Id_Owner) return query;
359         if (User.id == Parent.Id_Owner) return query;
360         var permission = (from c in content.ViewObject where
           c.Id_Owner == User.id select new { Id_Task =
           c.Id_Task }).Distinct();
361         query = from c in query join p in permission on c.Id equals
           p.Id_Task select c;
362         return query;
363     }
364 }
365
366 IQueryable<String> IndustryQuery
367 {
368     get
369     {
370         var query = from c in content.Companies select c.Industry;
371         return query.Distinct();
372     }
373 }
374
375 IQueryable<Persons> PersonQuery
376 {
377     get
378     {
379         var query = from c in content.Persons select c;
380         return query.Distinct();
381     }
382 }
383
384 IQueryable<String> CityQuery
385 {
386     get
387     {
388         var query = from c in content.Companies select c.City;
389         return query.Distinct();
390     }
391 }
392
393 IQueryable<int?> YearQuery
394 {
395     get
396     {
397         var query = from c in content.Projects select c.Year;
398         return query.Distinct();

```

```

399     }
400 }
401
402 IQueryable<Categories> CategoryQuery
403 {
404     get
405     {
406         return from c in content.Categories select c;
407     }
408 }
409
410 IQueryable<Persons> OwnerQuery
411 {
412     get
413     {
414         return from c in content.Persons select c;
415     }
416 }
417
418
419 IQueryable<String> CountryQuery
420 {
421     get
422     {
423         var query = from c in content.Companies select c.Country;
424         return query.Distinct();
425     }
426 }
427
428 public void Company_ViewUpdate()
429 {
430     int saveId = -1;
431     if (Company_DataGrid.SelectedValue != null) saveId = ((Companies) ↗
         (Company_DataGrid.SelectedItem)).id;
432     companiesViewSource.Source = CompanyQuery.ToList();
433     Company_DataGrid.SelectedItem = (from c in (List<Companies>) ↗
         (companiesViewSource.Source) where c.id==saveId select ↗
         c).FirstOrDefault();
434     if (Company_DataGrid.SelectedItem == null) if ↗
         (Company_DataGrid.Items.Count > 0) ↗
         Company_DataGrid.SelectedIndex = 0;
435 }
436
437 private void Company_ButtonAdd_Click(object sender, RoutedEventArgs ↗
    e)
438 {
439     if (!Company_CheckPermission()) return;
440     DlgCompany dlg = new DlgCompany(content);
441     if (User != null) dlg.item.Id_Owner = User.id;
442     if (dlg.ShowDialog() == false) return;
443     Company_ViewUpdate();
444     Company_BoxCity.ItemsSource = CityQuery.ToList();
445     Company_BoxIndustry.ItemsSource = IndustryQuery.ToList();
446     Company_DataGrid.ItemsSource = CompanyQuery.ToList();
447 }
448

```

```
449     private void Company_ButtonEdit_Click(object sender, RoutedEventArgs e)
450     {
451         if (Company_DataGrid.SelectedItem == null) return;
452         if (!Company_CheckPermission()) return;
453         DlgCompany dlg = new DlgCompany(content, (Companies)
454             Company_DataGrid.SelectedItem);
455         if (dlg.ShowDialog() == false) return;
456         Company_ViewUpdate();
457         Company_BoxCity.ItemsSource = CityQuery.ToList();
458         Company_BoxIndustry.ItemsSource = IndustryQuery.ToList();
459         Company_DataGrid.ItemsSource = CompanyQuery.ToList();
460     }
461     bool Company_CheckPermission()
462     {
463         if (User == null) return true;
464         if (User.IsAdmin != null) return true;
465         Companies item = (Companies)Company_DataGrid.SelectedItem;
466         if (Company_DataGrid.SelectedValue != null) if (User.id ==
467             item.Id_Owner) return true;
468         MessageBox.Show("Этот объект могут изменить только администратор
469             или владелец", "Не достаточно прав для операции",
470             MessageBoxButton.OKCancel, MessageBoxImage.Warning);
471         return false;
472     }
473     bool Project_CheckPermission()
474     {
475         if (User == null) return true;
476         if (User.IsAdmin != null) return true;
477         Projects item = (Projects)Project_DataGrid.SelectedItem;
478         if (Project_DataGrid.SelectedValue != null)
479         {
480             if (User.id == item.Id_Owner) return true;
481             if (User.id == item.Companies.Id_Owner) return true;
482         }
483         MessageBox.Show("Этот объект могут изменить только администратор
484             или владелец", "Не достаточно прав для операции",
485             MessageBoxButton.OKCancel, MessageBoxImage.Warning);
486         return false;
487     }
488     bool List_CheckPermission()
489     {
490         if (User == null) return true;
491         if (User.IsAdmin != null) return true;
492         Lists item = (Lists)List_DataGrid.SelectedItem;
493         if (List_DataGrid.SelectedValue != null)
494         {
495             if (User.id == item.Id_Owner) return true;
496             if (User.id == item.Projects.Id_Owner) return true;
497             if (User.id == item.Projects.Companies.Id_Owner) return true;
498         }
499         MessageBox.Show("Этот объект могут изменить только администратор
500             или владелец", "Не достаточно прав для операции",
```

```
        MessageBoxButton.OKCancel, MessageBoxImage.Warning);
497     return false;
498 }
499
500 bool Task_CheckPermission()
501 {
502     if (User == null) return true;
503     if (User.IsAdmin != null) return true;
504     Tasks item = (Tasks)Task_DataGrid.SelectedItem;
505     if (Task_DataGrid.SelectedValue != null)
506     {
507         if (User.id == item.Id_Owner) return true;
508         if (User.id == item.Lists.Id_Owner) return true;
509         if (User.id == item.Lists.Projects.Id_Owner) return true;
510         if (User.id == item.Lists.Projects.Companies.Id_Owner) return true;
511     }
512     MessageBox.Show("Этот объект могут изменить только администратор
    или владелец", "Не достаточно прав для операции",
    MessageBoxButton.OKCancel, MessageBoxImage.Warning);
513     return false;
514 }
515
516 bool Comment_CheckPermission()
517 {
518     if (User == null) return true;
519     if (User.IsAdmin != null) return true;
520     Comments item = (Comments)Comment_DataGrid.SelectedItem;
521     if (Comment_DataGrid.SelectedValue != null)
522     {
523         if (User.id == item.Id_Person) return true;
524         if (User.id == item.Tasks.Id_Owner) return true;
525         if (User.id == item.Tasks.Lists.Id_Owner) return true;
526         if (User.id == item.Tasks.Lists.Projects.Id_Owner) return true;
527         if (User.id == item.Tasks.Lists.Projects.Companies.Id_Owner) return true;
528     }
529     MessageBox.Show("Этот объект могут изменить только администратор
    или владелец", "Не достаточно прав для операции",
    MessageBoxButton.OK, MessageBoxImage.Warning);
530     return false;
531 }
532
533 private void Company_ButtonDelete_Click(object sender,
    RoutedEventArgs e)
534 {
535     if (Company_DataGrid.SelectedItem == null ) return;
536     if (!Company_CheckPermission()) return;
537     if (User.IsAdmin == null) return;
538     if (MessageBox.Show("Удалить компанию " + ((Companies)
    Company_DataGrid.SelectedItem).Name + " и все связанные с ней
    объекты?", "Удаление компании", MessageBoxButton.OKCancel,
    MessageBoxImage.Warning) == MessageBoxResult.Cancel)
539         return;
540     content.Companies.Remove(((Companies)
```

```

        (Company_DataGrid.SelectedItem));
541     content.SaveChanges();
542     Company_ViewUpdate();
543 }
544
545 private void Company_FilterButton_Click(object sender, EventArgs e)
546 {
547     Company_ViewUpdate();
548     Company_DataGrid.Focus();
549 }
550
551 private void Company_ButtonToProject_Click(object sender,           ↗
552     RoutedEventArgs e)
553 {
554     Project_Tab.Focus();
555 }
556
557 private void Company_DataGrid_SelectionChanged(object sender,       ↗
558     SelectionChangedEventArgs e)
559 {
560     Project_ViewUpdate();
561     Company_ButtonToProject.IsEnabled =
562         Company_DataGrid.SelectedValue != null;           ↗
563     Project_Tab.IsEnabled = Company_DataGrid.SelectedValue != null;
564     Company_ButtonDelete.IsEnabled = Company_DataGrid.SelectedValue !=
565         null;
566     Company_ButtonEdit.IsEnabled = Company_DataGrid.SelectedValue !=
567         null;
568 }
569
570 private void Company_DataGrid_Click(object sender, RoutedEventArgs e)
571 {
572     var hyperlink = e.OriginalSource as Hyperlink;
573     if (hyperlink != null)
574     {
575         Process.Start(hyperlink.NavigateUri.ToString());
576     }
577 }
578
579 private void Project_ButtonDelete_Click(object sender,           ↗
580     RoutedEventArgs e)
581 {
582     if (Project_DataGrid.SelectedValue == null) return;
583     if (!Company_CheckPermission()) return;
584     if (MessageBox.Show("Удалить компанию " + ((Projects)
585         Project_DataGrid.SelectedItem).Name + " и все связанные с ней
586         объекты?",
587         "Удаление компании", MessageBoxButton.OK,
588         MessageBoxImage.Warning) == MessageBoxResult.Cancel)
589         return;
590     content.Projects.Remove(((Projects)
591         (Project_DataGrid.SelectedItem)));
592     content.SaveChanges();
593     Project_ViewUpdate();
594 }
595

```

```

585     private void Project_ButtonEdit_Click(object sender, RoutedEventArgs e)
586     {
587         if (!Project_CheckPermission()) return;
588         if (Project_DataGrid.SelectedItem == null) return;
589         DlgProject dlg = new DlgProject(content, (Projects)
590             Project_DataGrid.SelectedItem);
591         if (dlg.ShowDialog() == false) return;
592         Project_ViewUpdate();
593     }
594     private void Project_ButtonAdd_Click(object sender, RoutedEventArgs e)
595     {
596         if (!Company_CheckPermission()) return;
597         DlgProject dlg = new DlgProject(content);
598         dlg.item.Id_Company = ((Companies)
599             Company_DataGrid.SelectedItem).id;
600         if (User != null) dlg.item.Id_Owner = User.id;
601         if (dlg.ShowDialog() == false) return;
602         Project_ViewUpdate();
603     }
604     private void Project_FilterButton_Click(object sender,
605         RoutedEventArgs e)
606     {
607         Project_ViewUpdate();
608         Project_DataGrid.Focus();
609     }
610     private void Project_ButtonToProject_Click(object sender,
611         RoutedEventArgs e)
612     {
613         List_Tab.Focus();
614     }
615     public void Project_ViewUpdate()
616     {
617         int saveId = -1;
618         if (Project_DataGrid.SelectedValue != null) saveId = ((Projects)
619             Project_DataGrid.SelectedItem).Id;
620         projectsViewSource.Source = ProjectQuery.ToList();
621         Project_DataGrid.SelectedItem = (from c in (List<Projects>)
622             projectsViewSource.Source) where c.Id == saveId select
623             c).FirstOrDefault();
624         if (Project_DataGrid.SelectedItem == null) if
625             (Project_DataGrid.Items.Count > 0)
626             Project_DataGrid.SelectedIndex = 0;
627         Project_BoxYear.ItemsSource = YearQuery.ToList();
628         Project_BoxCategory.ItemsSource = CategoryQuery.ToList();
629         Project_BoxOwner.ItemsSource = OwnerQuery.ToList();
630     }
631     private void Project_DataGrid_SelectionChanged(object sender,
632         SelectionChangedEventArgs e)
633     {

```

```

629         List_ViewUpdate();
630         Project_ButtonToList.IsEnabled = Project_DataGrid.SelectedItem != null;
631         List_Tab.IsEnabled = Project_DataGrid.SelectedItem != null;
632         Project_ButtonDelete.IsEnabled = Project_DataGrid.SelectedItem != null;
633         Project_ButtonEdit.IsEnabled = Project_DataGrid.SelectedItem != null;
634     }
635
636     private void List_ButtonEdit_Click(object sender, RoutedEventArgs e)
637     {
638         if (!List_CheckPermission()) return;
639         if (List_DataGrid.SelectedItem == null) return;
640         DlgList dlg = new DlgList(content, (Lists) List_DataGrid.SelectedItem);
641         if (dlg.ShowDialog() == false) return;
642         List_ViewUpdate();
643     }
644
645     private void List_ButtonDelete_Click(object sender, RoutedEventArgs e)
646     {
647         if (!Project_CheckPermission()) return;
648         if (List_DataGrid.SelectedItem == null) return;
649         if (MessageBox.Show("Удалить этап " + ((Lists) List_DataGrid.SelectedItem).Name + " и все связанные с ним объекты?", "Удаление компании", MessageBoxButton.OKCancel, MessageBoxImage.Warning) == MessageBoxResult.Cancel) return;
650         content.Lists.Remove(((Lists) (List_DataGrid.SelectedItem)));
651         content.SaveChanges();
652         List_ViewUpdate();
653     }
654
655     private void List_FilterButton_Click(object sender, RoutedEventArgs e)
656     {
657         List_ViewUpdate();
658     }
659
660     private void List_ButtonToProject_Click(object sender, RoutedEventArgs e)
661     {
662         Task_Tab.Focus();
663     }
664
665     public void List_ViewUpdate()
666     {
667         int saveId = -1;
668         if (List_DataGrid.SelectedItem != null) saveId = ((Lists) (List_DataGrid.SelectedItem)).Id;
669         listViewSource.Source = ListQuery.ToList();
670         List_DataGrid.SelectedItem = (from c in (List<Lists>) (listViewSource.Source) where c.Id == saveId select c).FirstOrDefault();
671         if (List_DataGrid.SelectedItem == null) if

```

```

        (List_DataGrid.Items.Count > 0) List_DataGrid.SelectedIndex =
        0;
672     }
673
674     public void Comment_ViewUpdate()
675     {
676         int saveId = -1;
677         if (Comment_DataGrid.SelectedItem != null) saveId = ((Comments)
        (Comment_DataGrid.SelectedItem)).Id;
678         commentsViewSource.Source = CommentQuery.ToList();
679         Comment_DataGrid.SelectedItem = (from c in (List<Comments>)
        (commentsViewSource.Source) where c.Id == saveId select
        c).FirstOrDefault();
680         if (Comment_DataGrid.SelectedItem == null) if
        (Comment_DataGrid.Items.Count > 0)
        Comment_DataGrid.SelectedIndex = 0;
681     }
682
683     private void List_ButtonAdd_Click(object sender, RoutedEventArgs e)
684     {
685         if (!Project_CheckPermission()) return;
686         DlgList dlg = new DlgList(content);
687         dlg.item.Id_Project = ((Projects)
        (Project_DataGrid.SelectedItem)).Id;
688         if (User != null) dlg.item.Id_Owner = User.id;
689         if (dlg.ShowDialog() == false) return;
690         List_ViewUpdate();
691     }
692
693     public void Task_ViewUpdate()
694     {
695         int saveId = -1;
696         if (Task_DataGrid.SelectedValue != null) saveId = ((Tasks)
        (Task_DataGrid.SelectedItem)).Id;
697         tasksViewSource.Source = TaskQuery.ToList();
698         Task_DataGrid.SelectedItem = (from c in (List<Tasks>)
        (tasksViewSource.Source) where c.Id == saveId select
        c).FirstOrDefault();
699         if (Task_DataGrid.SelectedItem == null) if
        (Task_DataGrid.Items.Count > 0) Task_DataGrid.SelectedIndex =
        0;
700     }
701
702
703     private void Task_ButtonAdd_Click(object sender, RoutedEventArgs e)
704     {
705         if (!List_CheckPermission()) return;
706         DlgTask dlg = new DlgTask(content);
707         dlg.item.Id_List = ((Lists)(List_DataGrid.SelectedItem)).Id;
708         if (User != null) dlg.item.Id_Owner = User.id;
709         if (dlg.ShowDialog() == false) return;
710         Task_ViewUpdate();
711     }
712
713     private void Task_ButtonEdit_Click(object sender, RoutedEventArgs e)
714     {

```



```
715         if (!Task_CheckPermission()) return;
716         if (Task_DataGrid.SelectedItem == null) return;
717         DlgTask dlg = new DlgTask(content, (Tasks) Task_DataGrid.SelectedItem);
718         if (dlg.ShowDialog() == false) return;
719         Task_ViewUpdate();
720     }
721
722
723     private void Task_ButtonDelete_Click(object sender, RoutedEventArgs e)
724     {
725         if (!List_CheckPermission()) return;
726         if (Task_DataGrid.SelectedItem == null) return;
727         if (MessageBox.Show("Удалить задачу " + ((Tasks) Task_DataGrid.SelectedItem).Name + " и все связанные с ней объекты?",
728             "Удаление компании", MessageBoxButton.OKCancel,
729             MessageBoxImage.Warning) == MessageBoxResult.Cancel)
730             return;
731         content.Tasks.Remove(((Tasks)(Task_DataGrid.SelectedItem)));
732         content.SaveChanges();
733         Task_ViewUpdate();
734     }
735
736     private void List_DataGrid_SelectionChanged_1(object sender, SelectionChangedEventArgs e)
737     {
738         Task_ViewUpdate();
739         List_ButtonToTask.IsEnabled = List_DataGrid.SelectedValue != null;
740         List_ButtonDelete.IsEnabled = List_DataGrid.SelectedValue != null;
741         List_ButtonEdit.IsEnabled = List_DataGrid.SelectedValue != null;
742         Task_Tab.IsEnabled = List_DataGrid.SelectedValue != null;
743     }
744
745     private void Company_BoxIndustry_SelectionChanged(object sender, SelectionChangedEventArgs e)
746     {
747     }
748
749     private void Task_DataGrid_SelectionChanged(object sender, SelectionChangedEventArgs e)
750     {
751         Comment_ViewUpdate();
752         Task_ButtonDelete.IsEnabled = Task_DataGrid.SelectedValue != null;
753         Task_ButtonEdit.IsEnabled = Task_DataGrid.SelectedValue != null;
754         Comment_ButtonAdd.IsEnabled = Task_DataGrid.SelectedValue != null;
755     }
756
757
```

```

758     private void MenuLogIn_Click(object sender, RoutedEventArgs e)
759     {
760         DlgLogin dlgLogin = new DlgLogin(content);
761         dlgLogin.BoxDate.SelectedDate = CurrentDate;
762         if (User != null) dlgLogin.BoxUser.SelectedValue = User.id;
763         dlgLogin.ShowDialog();
764         if (dlgLogin.DialogResult == false) return;
765         User = (from c in content.Persons where c.id == ((Persons)
766             (dlgLogin.BoxUser.SelectedItem)).id select c).First();
767         CurrentDate = (DateTime)dlgLogin.BoxDate.SelectedDate;
768     }
769     private void Comment_DataGrid_SelectionChanged(object sender,
770         SelectionChangedEventArgs e)
771     {
772         Comment_ButtonEdit.IsEnabled = Comment_DataGrid.SelectedValue !=
773             null;
774         Comment_ButtonDelete.IsEnabled = Comment_DataGrid.SelectedValue !=
775             null;
776     }
777     private void Comment_ButtonAdd_Click(object sender, RoutedEventArgs
778         e)
779     {
780         if (!Task_CheckPermission()) return;
781         if (Task_DataGrid.SelectedValue == null) return;
782         DlgComments dlg = new DlgComments(content);
783         dlg.item.Id_Task = ((Tasks)(Task_DataGrid.SelectedItem)).Id;
784         dlg.item.Create_Date = DateTime.Now;
785         dlg.item.Edit_Date = DateTime.Now;
786         if (User != null) dlg.item.Id_Person = User.id;
787         if (dlg.ShowDialog() == false) return;
788         Comment_ViewUpdate();
789         MyComment_ViewUpdate();
790     }
791     private void Comment_ButtonEdit_Click(object sender, RoutedEventArgs
792         e)
793     {
794         if (!Comment_CheckPermission()) return;
795         if (Comment_DataGrid.SelectedValue == null) return;
796         DlgComments dlg = new DlgComments(content, (Comments)
797             Comment_DataGrid.SelectedItem);
798         dlg.item.Edit_Date = DateTime.Now;
799         if (User != null) dlg.item.Id_Person = User.id;
800         if (dlg.ShowDialog() == false) return;
801         content.SaveChanges();
802         Comment_ViewUpdate();
803         MyComment_ViewUpdate();
804     }
805     private void Comment_ButtonDelete_Click(object sender,
806         RoutedEventArgs e)
807     {
808         if (!Task_CheckPermission()) return;
809     }

```

```

806
807     private void Comment_ButtonDelete_Click_1(object sender,           ↗
            RoutedEventArgs e)
808     {
809         if (!Task_CheckPermission()) return;
810         if (Comment_DataGrid.SelectedItem == null) return;
811         if (MessageBox.Show("Удалить комментарий?",
812             "Удаление компании", MessageBoxButton.OKCancel,           ↗
            MessageBoxImage.Warning) == MessageBoxResult.Cancel)       ↗
            return;
813         content.Comments.Remove(((Comments)                           ↗
            (Comment_DataGrid.SelectedItem)));
814         content.SaveChanges();
815         Comment_ViewUpdate();
816     }
817
818     public void MyTask_ViewUpdate()
819     {
820         int saveId = -1;
821         if (MyTask_DataGrid.SelectedValue != null) saveId = ((Tasks)   ↗
            (MyTask_DataGrid.SelectedItem)).Id;
822         myTasksViewSource.Source = MyTaskQuery.ToList();
823         MyTask_DataGrid.SelectedItem = (from c in (List<Tasks>)         ↗
            (myTasksViewSource.Source) where c.Id == saveId select       ↗
            c).FirstOrDefault();
824         if (MyTask_DataGrid.SelectedItem == null) if                   ↗
            (MyTask_DataGrid.Items.Count > 0) MyTask_DataGrid.SelectedIndex ↗
            = 0;
825         viewmodel = null;
826         viewmodel = new ListBackedProjectViewModel();
827         viewmodel.DownloadDataSource(MyTaskQuery);
828         gantt.Project.DataContext = viewmodel;
829         GetStatistic();
830     }
831
832     private void MyTask_DataGrid_SelectionChanged(object sender,       ↗
            SelectionChangedEventArgs e)
833     {
834         Comment_ViewUpdate();
835         MyComment_ViewUpdate();
836         MyTask_ButtonDelete.IsEnabled = MyTask_DataGrid.SelectedValue != ↗
            null;
837         MyTask_ButtonEdit.IsEnabled = MyTask_DataGrid.SelectedValue !=  ↗
            null;
838         MyComment_ButtonAdd.IsEnabled = MyTask_DataGrid.SelectedValue != ↗
            null;
839     }
840
841     private void MyTask_ButtonEdit_Click(object sender, RoutedEventArgs  ↗
            e)
842     {
843         if (!Task_CheckPermission()) return;
844         if (MyTask_DataGrid.SelectedItem == null) return;
845         DlgTask dlg = new DlgTask(content, (Tasks)                     ↗
            MyTask_DataGrid.SelectedItem);
846         if (dlg.ShowDialog() == false) return;

```

```
847         Task_ViewUpdate();
848         MyTask_ViewUpdate();
849     }
850
851     private void MyTask_ButtonDelete_Click(object sender, RoutedEventArgs e)
852     {
853         if (MyTask_DataGrid.SelectedItem == null) return;
854         if (MessageBox.Show("Удалить задачу " + ((Tasks)
855             MyTask_DataGrid.SelectedItem).Name + " и все связанные с ней
856             объекты?",
857                 "Удаление компании", MessageBoxButton.OKCancel,
858                 MessageBoxImage.Warning) == MessageBoxResult.Cancel)
859             return;
860         content.Tasks.Remove(((Tasks)(MyTask_DataGrid.SelectedItem));
861         content.SaveChanges();
862         Task_ViewUpdate();
863         MyTask_ViewUpdate();
864     }
865
866     private void MyComment_ButtonAdd_Click(object sender, RoutedEventArgs e)
867     {
868         if (MyTask_DataGrid.SelectedValue == null) return;
869         if (!Task_CheckPermission()) return;
870         DlgComments dlg = new DlgComments(content);
871         dlg.item.Id_Task = ((Tasks)(MyTask_DataGrid.SelectedItem)).Id;
872         dlg.item.Create_Date = DateTime.Now;
873         dlg.item.Edit_Date = DateTime.Now;
874         if (User != null) dlg.item.Id_Person = User.id;
875         if (dlg.ShowDialog() == false) return;
876         Comment_ViewUpdate();
877         MyComment_ViewUpdate();
878     }
879
880     private void MyComment_ButtonEdit_Click(object sender,
881         RoutedEventArgs e)
882     {
883         if (MyComment_DataGrid.SelectedValue == null) return;
884         if (!Comment_CheckPermission()) return;
885         DlgComments dlg = new DlgComments(content, (Comments)
886             MyComment_DataGrid.SelectedItem);
887         dlg.item.Edit_Date = DateTime.Now;
888         if (User != null) dlg.item.Id_Person = User.id;
889         if (dlg.ShowDialog() == false) return;
890         content.SaveChanges();
891         Comment_ViewUpdate();
892         MyComment_ViewUpdate();
893     }
894
895     private void MyComment_ButtonDelete_Click(object sender,
896         RoutedEventArgs e)
897     {
898         if (MyComment_DataGrid.SelectedItem == null) return;
899         if (!Task_CheckPermission()) return;
900         if (MessageBox.Show("Удалить комментарий?",
```

```

894         "Удаление компании", MessageBoxButton.OKCancel,
           MessageBoxImage.Warning) == MessageBoxResult.Cancel)
           return;
895         content.Comments.Remove(((Comments)
           (MyComment_DataGrid.SelectedItem));
896         content.SaveChanges();
897         Comment_ViewUpdate();
898         MyComment_ViewUpdate();
899     }
900
901     public void MyComment_ViewUpdate()
902     {
903         int saveId = -1;
904         if (MyComment_DataGrid.SelectedItem != null) saveId = ((Comments)
           (MyComment_DataGrid.SelectedItem)).Id;
905         myCommentsViewSource.Source = MyCommentQuery.ToList();
906         MyComment_DataGrid.SelectedItem = (from c in (List<Comments>)
           (myCommentsViewSource.Source) where c.Id == saveId select
           c).FirstOrDefault();
907         if (MyComment_DataGrid.SelectedItem == null) if
           (MyComment_DataGrid.Items.Count > 0)
           MyComment_DataGrid.SelectedIndex = 0;
908     }
909
910     private void MyTask_FilterButton_Click(object sender, RoutedEventArgs e)
911     {
912         MyTask_ViewUpdate();
913     }
914
915     private void MyComment_DataGrid_SelectionChanged(object sender,
           SelectionChangedEventArgs e)
916     {
917         MyComment_ButtonEdit.IsEnabled =
           MyComment_DataGrid.SelectedValue != null;
918         MyComment_ButtonDelete.IsEnabled =
           MyComment_DataGrid.SelectedValue != null;
919     }
920
921     private void MenuChangeUser_Click(object sender, RoutedEventArgs e)
922     {
923         Persons curUser = User;
924
925         DlgLogin dlgLogin = null;
926         while (true)
927         {
928             dlgLogin = new DlgLogin(content);
929             dlgLogin.BoxDate.SelectedDate = CurrentDate;
930             dlgLogin.ShowDialog();
931             if (dlgLogin.DialogResult == false) return;
932             if (dlgLogin.BoxUser.SelectedValue != null)
933             {
934                 if (((Persons)(dlgLogin.BoxUser.SelectedItem)).Pasword ==
           null)
935                 {
936                     if (dlgLogin.BoxPassword.Password == "") break;

```

```

937     }
938     else if (((Persons)
(dlgLogin.BoxUser.SelectedItem)).Pasword ==
dlgLogin.BoxPassword.Password) break;
939 }
940     MessageBox.Show("Не верный пароль", "Ошибка авторизации",
    MessageBoxButton.OKCancel, MessageBoxImage.Warning);
941 }
942     if (curUser.id == ((Persons)(dlgLogin.BoxUser.SelectedItem)).id)
return;
943     User = (from c in content.Persons where c.id == ((Persons)
(dlgLogin.BoxUser.SelectedItem)).id select c).First();
944     CurrentDate = (DateTime)dlgLogin.BoxDate.SelectedDate;
945     Window_Loaded(sender, e);
946     MenuUser.Header = User.Email;
947 }
948
949 private void MenProfile_Click(object sender, RoutedEventArgs e)
950 {
951     DlgPerson dlg = new DlgPerson(content, User);
952     dlg.CheckBox_Admin.IsEnabled = (User.IsAdmin != null);
953     if (dlg.ShowDialog() == false) return;
954     MyTask_BoxOwner.ItemsSource = content.Persons.ToList();
955     Task_BoxOwner.ItemsSource = content.Persons.ToList();
956     Company_BoxOwner.ItemsSource = content.Persons.ToList();
957     List_BoxOwner.ItemsSource = content.Persons.ToList();
958     categoriesViewSource.Source = content.Categories.ToList();
959 }
960
961 private void Button_Click(object sender, RoutedEventArgs e)
962 {
963     Gantt gantt = new Gantt(content);
964     gantt.Show();
965 }
966
967
968 public void GetStatistic()
969 {
970     IQueryable<Tasks> query = MyTaskQuery;
971     string s = "Всего отобрано: " + (from c in MyTaskQuery where
c.Name.Contains(Task_BoxName.Text) select c).Count().ToString()
+ "\n";
972     s = s + "В ожидании: " + (from c in MyTaskQuery where
c.Name.Contains(Task_BoxName.Text) where c.Status == "В
ожидании" select c).Count().ToString() + "\n";
973     s = s + "Выполняется " + (from c in MyTaskQuery where
c.Name.Contains(Task_BoxName.Text) where c.Status ==
"Выполняется" select c).Count().ToString() + "\n";
974     s = s + "Завершено: " + (from c in MyTaskQuery where
c.Name.Contains(Task_BoxName.Text) where c.Status ==
"Выполнено" select c).Count().ToString() + "\n";
975     s = s + "Просрочено: " + (from c in MyTaskQuery where
c.Name.Contains(Task_BoxName.Text) where c.Status !=
"Выполнено" && c.Due_Date < DateTime.Today select c).Count
().ToString() + "\n";
976

```

```
977         Statistic_Box.Text = s;
978     }
979
980 }
981
982 public class ObservableModel : INotifyPropertyChanged
983 {
984     public event PropertyChangedEventHandler PropertyChanged;
985     protected void NotifyPropertyChanged(String info)
986     {
987         if (PropertyChanged != null)
988         {
989             PropertyChanged(this, new PropertyChangedEventArgs(info));
990         }
991     }
992 }
993
994 public class TaskModel : ObservableModel
995 {
996     private string _taskId;
997     public string TaskID
998     {
999         get
1000         {
1001             return _taskId;
1002         }
1003         set
1004         {
1005             if (_taskId != value)
1006             {
1007                 _taskId = value;
1008                 this.NotifyPropertyChanged("TaskID");
1009             }
1010         }
1011     }
1012     private string _tasks;
1013     public string Tasks
1014     {
1015         get
1016         {
1017             return _tasks;
1018         }
1019         set
1020         {
1021             if (_tasks != value)
1022             {
1023                 _tasks = value;
1024                 this.NotifyPropertyChanged("Tasks");
1025             }
1026         }
1027     }
1028     private string _name;
1029     public string Name
1030     {
1031         get
1032     {
```

```
1033         return _name;
1034     }
1035     set
1036     {
1037         if (_name != value)
1038         {
1039             _name = value;
1040             this.NotifyPropertyChanged("Name");
1041         }
1042     }
1043 }
1044 private ProjectTaskConstraintType _constraintType;
1045 public ProjectTaskConstraintType ConstraintType
1046 {
1047     get
1048     {
1049         return _constraintType;
1050     }
1051     set
1052     {
1053         if (_constraintType != value)
1054         {
1055             _constraintType = value;
1056             this.NotifyPropertyChanged("ConstraintType");
1057         }
1058     }
1059 }
1060 private DateTime? _constraintDate;
1061 public DateTime? ConstraintDate
1062 {
1063     get
1064     {
1065         return _constraintDate;
1066     }
1067     set
1068     {
1069         if (_constraintDate != value)
1070         {
1071             _constraintDate = value;
1072             this.NotifyPropertyChanged("ConstraintDate");
1073         }
1074     }
1075 }
1076 private ProjectDurationFormat _durationFormat;
1077 public ProjectDurationFormat DurationFormat
1078 {
1079     get
1080     {
1081         return _durationFormat;
1082     }
1083     set
1084     {
1085         if (_durationFormat != value)
1086         {
1087             _durationFormat = value;
1088             this.NotifyPropertyChanged("DurationFormat");
1089         }
1090     }
1091 }
```



```
1089     }
1090     }
1091 }
1092 private TimeSpan _durationInHours;
1093 public TimeSpan DurationInHours
1094 {
1095     get
1096     {
1097         return _durationInHours;
1098     }
1099     set
1100     {
1101         if (_durationInHours != value)
1102         {
1103             _durationInHours = value;
1104             this.NotifyPropertyChanged("DurationInHours");
1105         }
1106     }
1107 }
1108 private DateTime _start;
1109 public DateTime Start
1110 {
1111     get
1112     {
1113         return _start;
1114     }
1115     set
1116     {
1117         if (_start != value)
1118         {
1119             _start = value;
1120             this.NotifyPropertyChanged("Start");
1121         }
1122     }
1123 }
1124 private bool _isMilestone = false;
1125 public bool IsMilestone
1126 {
1127     get
1128     {
1129         return _isMilestone;
1130     }
1131     set
1132     {
1133         if (_isMilestone != value)
1134         {
1135             _isMilestone = value;
1136             this.NotifyPropertyChanged("IsMilestone");
1137         }
1138     }
1139 }
1140 private bool _isInProgress = true;
1141 public bool IsInProgress
1142 {
1143     get
1144     {
```

```
1145         return _isInProgress;
1146     }
1147     set
1148     {
1149         if (_isInProgress != value)
1150         {
1151             _isInProgress = value;
1152             this.NotifyPropertyChanged("IsInProgress");
1153         }
1154     }
1155 }
1156 private DateTime? _deadlineDate;
1157 public DateTime? DeadlineDate
1158 {
1159     get
1160     {
1161         return _deadlineDate;
1162     }
1163     set
1164     {
1165         if (_deadlineDate != value)
1166         {
1167             _deadlineDate = value;
1168             this.NotifyPropertyChanged("DeadlineDate");
1169         }
1170     }
1171 }
1172 private bool _isUndetermined = false;
1173 public bool IsUndetermined
1174 {
1175     get
1176     {
1177         return _isUndetermined;
1178     }
1179     set
1180     {
1181         if (_isUndetermined != value)
1182         {
1183             _isUndetermined = value;
1184             this.NotifyPropertyChanged("IsUndetermined");
1185         }
1186     }
1187 }
1188 private string _resourceName;
1189 public string ResourceName
1190 {
1191     get
1192     {
1193         return _resourceName;
1194     }
1195     set
1196     {
1197         if (_resourceName != value)
1198         {
1199             _resourceName = value;
1200             this.NotifyPropertyChanged("ResourceName");
1201         }
1202     }
1203 }
```

```
1201     }
1202     }
1203 }
1204 }
1205
1206 public class ListBackedProjectViewModel : ObservableModel
1207 {
1208     public ListBackedProjectViewModel()
1209     {
1210         // this.DownloadDataSource();
1211     }
1212     private ObservableCollection<TaskModel> _tasks;
1213
1214
1215
1216     public ObservableCollection<TaskModel> Tasks
1217     {
1218         get
1219         {
1220             return _tasks;
1221         }
1222         set
1223         {
1224             if (value != null)
1225             {
1226                 _tasks = value;
1227             }
1228             NotifyPropertyChanged("Tasks");
1229         }
1230     }
1231     public void DownloadDataSource(IQueryable<Tasks> query)
1232     {
1233         ObservableCollection<TaskModel> dataSource = new ObservableCollection<TaskModel>\(\);
1234         var TaskList = query.ToList();
1235         foreach (var item in TaskList)
1236         {
1237             if (item.Due_Date == null || item.Start_Date == null)
1238                 continue;
1239             TaskModel task = new TaskModel();
1240             task.TaskID = item.Id.ToString();
1241             task.Name = item.Name.ToString();
1242             task.IsInProgress = true;
1243             task.Start = (DateTime)item.Start_Date;
1244             task.IsMilestone = false;
1245             TimeSpan span = (TimeSpan)(item.Due_Date - item.Start_Date);
1246             span = new TimeSpan(span.Ticks / 4);
1247             task.DurationInHours = span;
1248             task.IsUndetermined = true;
1249             task.ResourceName = item.Persons.Email;
1250             task.DurationFormat = ProjectDurationFormat.Days;
1251             // task.DeadlineDate = ((DateTime)task.Start).AddHours
1252             (task.DurationInHours.TotalHours()).ToUniversalTime()
1253             dataSource.Add(task);
1254         }
1255     }
```

```
1254         this._tasks = dataSource;
1255     }
1256
1257
1258     }
1259
1260
1261
1262 }
1263
```