**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет» (ГГТУ) Ликино-Дулевский политехнический колледж- филиал ГГТУ**

**О Т Ч Ё Т**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

по ПМ.02 Осуществление интеграции программных модулей

Обучающегося\_\_\_\_\_\_Мелега Алексея Валерьевича \_\_\_\_\_\_\_\_\_\_\_

Ф.И.О.

Курс \_\_\_\_4\_\_\_\_\_ группа \_\_\_ИСП.20А\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность 09.02.07 Информационные системы и программирование \_\_\_

Место практики Ликино-Дулевский политехнический колледж – филиал ГГТУ

Период практики с 09.02.2024 г. по 22.02.2024\_г.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководители практики

от колледжа \_\_\_\_\_\_\_\_ Кузьмина Елена Евгеньевна \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_ Пронина Алла Юрьевна \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_ Тишкин Валентин Васильевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_

г. Ликино-Дулево

2024 г.

Содержание

[1. Разработка спецификации 3](#_Toc159501062)

[2. Разработка пользовательского интерфейса 7](#_Toc159501063)

[3. Алгоритм работы приложения 24](#_Toc159501064)

[4. Проектирование ER-диаграммы 25](#_Toc159501065)

[5. Проектирование и разработка базы данных 26](#_Toc159501066)

[6. Руководство системного программиста 36](#_Toc159501067)

[7. Руководство пользователя 53](#_Toc159501068)

[8. Отладка и тестирование 59](#_Toc159501069)

[9. Текст программы 61](#_Toc159501070)

[Заключение 81](#_Toc159501071)

[Список литературы 82](#_Toc159501072)

# Разработка спецификации

**Наименование модуля:** Конференция по информационной безопасности

**Цель модуля:** Система предназначена для организаторов конференций информационной безопасности.

**Функциональные возможности модулями:**

* Добавление, редактирование и удаление данных из таблиц:
  + - «Участники»
    - «Мод**е**раторы»
    - «Организаторы»
    - «Жюри»
    - «Активность»
* Вывод отчета в Word о мероприятиях
* Авторизация для организаторов
* Экспорт списка победителей в формате Excel.

**Взаимодействие с другими модулями**

Импорт данных из файла Excel в базу данных SQL представляет собой процесс загрузки данных из электронной таблицы Excel в таблицы БД.

Привязка базы данных SQL к приложению требует создания модели Entity Framework для взаимодействия с данными.

**Взаимодействие с пользователем**

УЧАСТНИК

Когда Участник зарегистрировался в системе, он может просматривать и изменять информацию своего профиля, работать в группе с другими участниками в рамках активностей мероприятия.

МОДЕРАТОР

Модератор в основном координируют участников в рамках своей активности. Модератор может зарегистрироваться в системе, может зарегистрироваться на мероприятие, в рамках мероприятия выбрать активность для модерации.

ОРГАНИЗАТОР

Организаторы управляют всеми участниками мероприятий, информацией о событиях в системе, добавляют мероприятия и активности в рамках мероприятий.

ЖЮРИ

Жюри могут просматривать информацию о активностях, которую они оценивают, например: участники, оценка и т.д.

**Входные данные**

Таблица №1 «Словарь данных»

|  |  |  |  |
| --- | --- | --- | --- |
| **ключ** | **поле** | **обязательное** | **примечание** |
| **Активность** | | | |
| Первичный | IdActivity | Да | Идентификатор |
| Внешний | IdModerator | Да | Внешний ключ к таблице модераторы |
| Внешний | IdJuri | Да | Первичный ключ к таблице жюри |
|  | ActivName | Да | Доп. событие |
|  | Time | Да | Время начала |
| Внешний | IdPerson | Да | Первичный ключ к таблице участники |
|  | Days | Да | Дни |
| Внешний | IdEvent | Да | Первичный ключ к таблице событие |
| **Жюри** | | | |
| Первичный | IdJuri | Да | Идентификатор |
|  | Email | Да | Почта |
|  | DateBirthday | Да | День рождение |
| Внешний | IdCountry | Да | Внешний ключ к таблице страны |
|  | Phone | Да | Телефон |
|  | Password | Да | Пароль |
|  | Photo | Да | Фото |
| Внешний | IdDirictory | Да | Внешний ключ к таблице направление |
| **Модераторы** | | | |
| Первичный | IdModerator | Да | Идентификатор |
|  | FIO | Да | ФИО |
|  | Pol | Да | Пол |
|  | Email | Да | Почта |
|  | DateBirthday | Да | День рождения |
| Внешний | IdCountry | Да | Внешний ключ к стране |
|  | Phone | Да | Телефон |
|  | Password | Да | Пароль |
|  | Photo | Да | Фото |
| Внешний | IdDirictory | Да | Внешний ключ к таблице направления |
| Внешний | IdEventmoder | Да | Внешний ключ к таблице событие модератора |
| **Организаторы** | | | |
| Первичный | IdOrganizer | Да | Идентификатор |
|  | Fio | Да | ФИО |
|  | Email | Да | Почта |
|  | DateBirthday | Да | День рождения |
| Внешний | IdCountry | Да | Внешний ключ к таблице страны |
|  | Phone | Да | Телефон |
|  | Password | Да | Пароль |
|  | Pol | Да | Пол |
|  | Photo | Да | Фото |
| **Участники** | | | |
| Первичный | IdPerson | Да | Идентификатор |
|  | Fio | Да | ФИО |
|  | Email | Да | Почта |
|  | DateBirthday | Да | Дата рождения |
| Внешний | IdCountry | Да | Внешний ключ к таблице страны |
|  | Password | Да | Пароль |
|  | Pol | Да | Пол |
|  | Photo | Да | Фото |

**Выходные данные**

«Вывод данных в Word»

|  |  |  |  |
| --- | --- | --- | --- |
| **ключ** | **поле** | **обязательное** | **примечание** |
| **Активность** | | | |
| Первичный | IdActivity | Да | Идентификатор |
|  | ActivName | Да | Доп. событие |
| Внешний | IdPerson | Да | Внешний ключ к таблице участники |
| Внешний | IdEvent | Да | Первичный ключ к таблице событие |

«Вывод данных в Excel»

|  |  |  |  |
| --- | --- | --- | --- |
| **ключ** | **поле** | **обязательное** | **примечание** |
| **Активное событие** | | | |
| Первичный | IdActivEvent | Да | Идентификатор |
| Внешний | IdEvent | Да | Внешний ключ к таблице события |
| Внешний | IdPerson | Да | Внешний ключ к таблице участники |

**Навигация внутри модуля**

Сначала пользователь переходит на страницу авторизации. После успешной авторизации, пользователь перейдет на страницу «Главное меню». С «Главного меню» он может перейти на такие страницы как «Участники», «Модераторы», «Жюри», «Организаторы», «Активность». Если нужно вернуться в «Главное меню» надо нажать ПКМ по таблице и выбрать в контекстном меню функцию «Главное меню».

**Требования к макету**

Все компоненты системы должны иметь единый согласованный

внешний вид, соответствующий руководству по стилю, а также

следующим требованиям:

− последовательный пользовательский интерфейс, позволяющий

перемещаться между существующими окнами в приложении (в том

числе обратно, например, с помощью кнопки «Назад»);

− соответствующий заголовок на каждом окне приложения.

# Разработка пользовательского интерфейса

Страница «Авторизация»

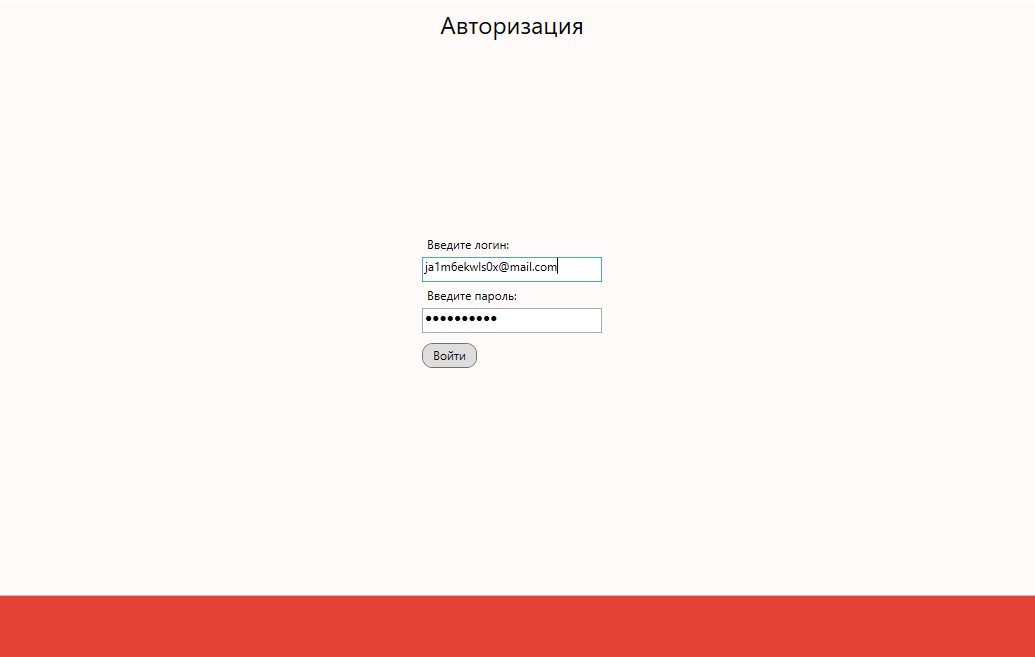


Рис.1 «Авторизация»

Код:

<Grid>

<StackPanel HorizontalAlignment="Center">

<Label Content="Авторизация" FontSize="24" />

</StackPanel>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">

<Label Content="Введите логин:" />

<TextBox Name="login" Width="180" Height="25" />

<Label Content="Введите пароль:" />

<PasswordBox Name="password" Width="180" Height="25"/>

<StackPanel Orientation="Horizontal" Margin="0,10,0,0">

<Button Name="btnenter" Content="Войти" Width="55" Height="25" Margin="0,0,5,0" Click="btnenter\_Click">

<Button.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="10"></Setter>

</Style>

</Button.Resources>

</Button>

</StackPanel>

</StackPanel>

</Grid>

Страница «Главное меню»

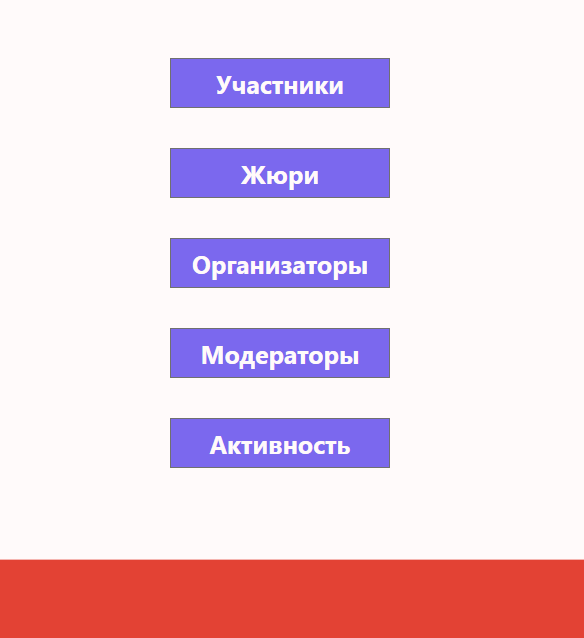


Рис.2 «Главное меню»

Код:

<Grid>

<StackPanel Orientation="Vertical" HorizontalAlignment="Center" VerticalAlignment="Center" Margin="15">

<Button Name="btnToPerson" Content="Участники" Style="{StaticResource MenuStyleBtn}" Click="btnToPerson\_Click"/>

<Button Name="btnToJuri" Content="Жюри" Style="{StaticResource MenuStyleBtn}" Click="btnToJuri\_Click"/>

<Button Name="btnToOrginiz" Content="Организаторы" Style="{StaticResource MenuStyleBtn}" Click="btnToOrginiz\_Click"/>

<Button Name="btnToModer" Content="Модераторы" Style="{StaticResource MenuStyleBtn}" Click="btnToModer\_Click"/>

<Button Name="btnToActiv" Content="Активность" Style="{StaticResource MenuStyleBtn}" Click="btnToActiv\_Click"/>

</StackPanel>

</Grid>

Страница «Участники»

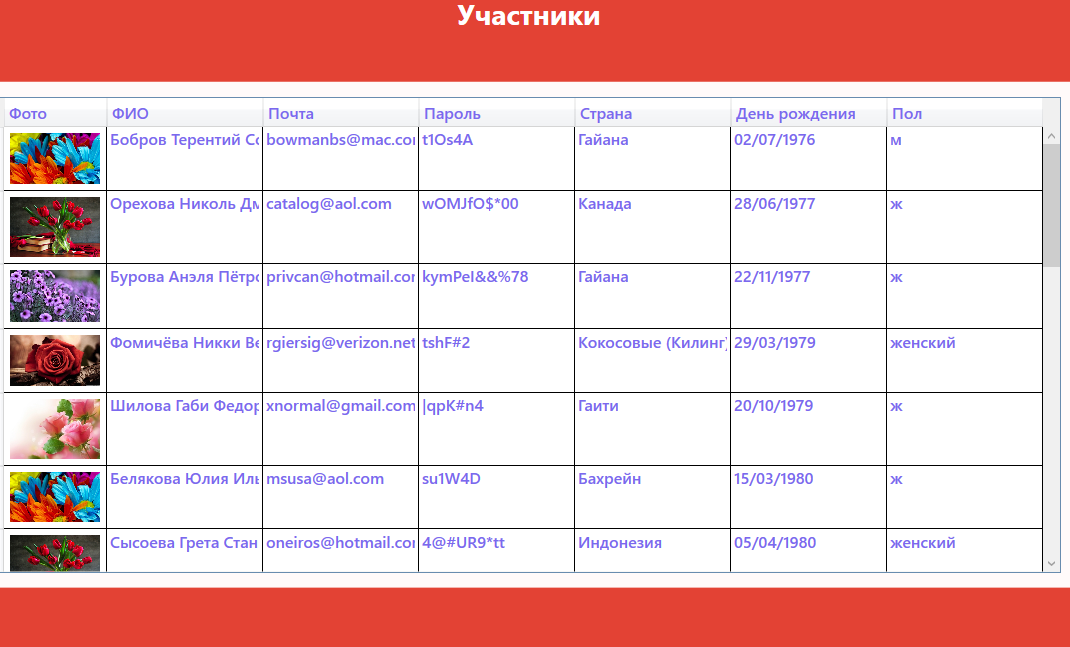


Рис.3 «Участники»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1.2\*"/>

<RowDefinition Height="7\*"/>

</Grid.RowDefinitions>

<StackPanel Background="#E34234">

<TextBlock Text="Участники" HorizontalAlignment="Center" Background="#E34234" Style="{StaticResource StyleTxtSubtitle}" Foreground="White"/>

</StackPanel>

<DataGrid Name="dtgPerson" Style="{StaticResource StyleDataGrid}" Grid.Row="1">

<DataGrid.Columns>

<DataGridTemplateColumn Width="auto" Header="Фото">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Width="90" Stretch="UniformToFill" HorizontalAlignment="Left" Margin="5">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>

\Resources\foto14.jpg

</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="ФИО" Width="0.5\*" Binding="{Binding Fio}"/>

<DataGridTextColumn Header="Почта" Width="0.5\*" Binding="{Binding Email}"/>

<DataGridTextColumn Header="Пароль" Width="0.5\*" Binding="{Binding Password}"/>

<DataGridTextColumn Header="Страна" Width="0.5\*" Binding="{Binding Country.NameCountry}"/>

<DataGridTextColumn Header="День рождения" Width="0.5\*" Binding="{Binding DateBirthday, StringFormat={}{0:dd/MM/yyyy}}"/>

<DataGridTextColumn Header="Пол" Width="0.5\*" Binding="{Binding Pol}"/>

</DataGrid.Columns>

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Добавить" Name="MenuAdd" Click="MenuAdd\_Click"/>

<MenuItem Header="Редактировать" Name="MenuEdit" Click="MenuEdit\_Click"/>

<MenuItem Header="Обновить" Name="MenuUpdate" Click="MenuUpdate\_Click"/>

<Separator/>

<MenuItem Header="Удалить" Name="MenuDel" Click="MenuDel\_Click"/>

<MenuItem Header="Главное меню" Name="MenuGlav" Click="MenuGlav\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

</DataGrid>

</Grid>

Страница добавления «Участники»

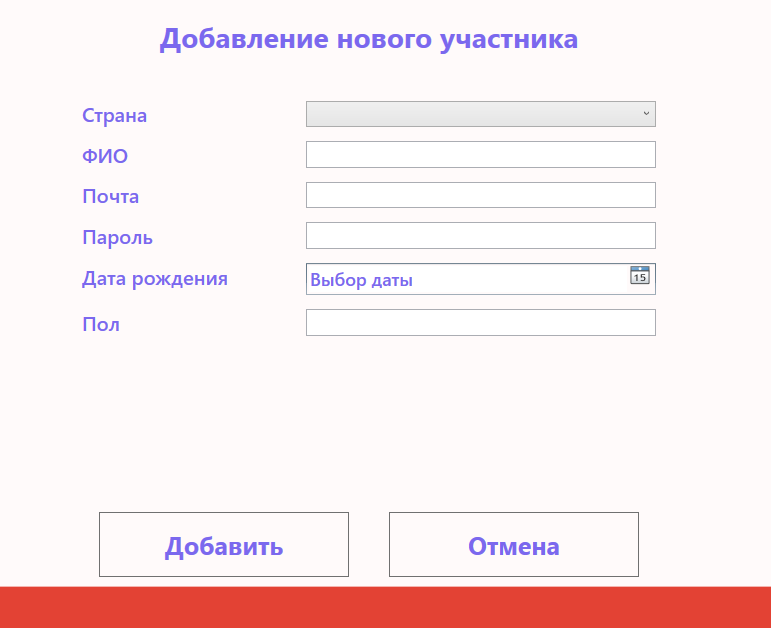


Рис.4 «Добавление участников»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Добавление нового участника" Style="{StaticResource StyleTxtSubtitle}" Name="Titletxt"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" Width="600" Margin="15">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Страна" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBCountry" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdCountry}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="ФИО" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="Txtfio" Text="{Binding Fio}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Почта" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtEmail" Text="{Binding Email}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Пароль" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtPassword" Text="{Binding Password}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Дата рождения" Style="{StaticResource StyleTxt}" Width="210"/>

<DatePicker x:Name="datePicker" IsDropDownOpen="False" SelectedDateFormat="Short" SelectedDate="{Binding DateBirthday}" Style="{StaticResource StyleDatePicker}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Пол" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtPol" Text="{Binding Pol}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Vertical" VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" Grid.Row="2" >

<Button Style="{StaticResource MenuStyleBtn2}" Content="Добавить" Name="BtnAdd" Click="BtnAdd\_Click" Margin="20"/>

<Button Style="{StaticResource MenuStyleBtn2}" Content="Отмена" Name="BtnCancel" Click="BtnCancel\_Click" Margin="20"/>

</StackPanel>

</StackPanel>

</Grid>

Страница «Жюри»



Рис.6 «Жюри»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1.2\*"/>

<RowDefinition Height="7\*"/>

</Grid.RowDefinitions>

<StackPanel Background="#E34234">

<TextBlock Text="Жюри" HorizontalAlignment="Center" Background="#E34234" Style="{StaticResource StyleTxtSubtitle}" Foreground="White"/>

</StackPanel>

<DataGrid Name="dtgJuri" Style="{StaticResource StyleDataGrid}" Grid.Row="1">

<DataGrid.Columns>

<DataGridTemplateColumn Width="auto" Header="Фото">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Width="90" Stretch="UniformToFill" HorizontalAlignment="Left" Margin="5">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>

\Resources\foto14.jpg

</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="Почта" Width="0.5\*" Binding="{Binding Email}"/>

<DataGridTextColumn Header="Пароль" Width="0.5\*" Binding="{Binding Password}"/>

<DataGridTextColumn Header="Страна" Width="0.5\*" Binding="{Binding Country.NameCountry}"/>

<DataGridTextColumn Header="День рождения" Width="0.5\*" Binding="{Binding DateBirthday, StringFormat={}{0:dd/MM/yyyy}}"/>

<DataGridTextColumn Header="Телефон" Width="0.5\*" Binding="{Binding Phone}"/>

<DataGridTextColumn Header="Направление" Width="0.5\*" Binding="{Binding Dirictory.IdDirictory}"/>

</DataGrid.Columns>

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Добавить" Name="MenuAdd" Click="MenuAdd\_Click"/>

<MenuItem Header="Редактировать" Name="MenuEdit" Click="MenuEdit\_Click"/>

<MenuItem Header="Обновить" Name="MenuUpdate" Click="MenuUpdate\_Click"/>

<Separator/>

<MenuItem Header="Удалить" Name="MenuDel" Click="MenuDel\_Click"/>

<MenuItem Header="Главное меню" Name="MenuGlav" Click="MenuGlav\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

</DataGrid>

</Grid>

Страница добавления «Жюри»

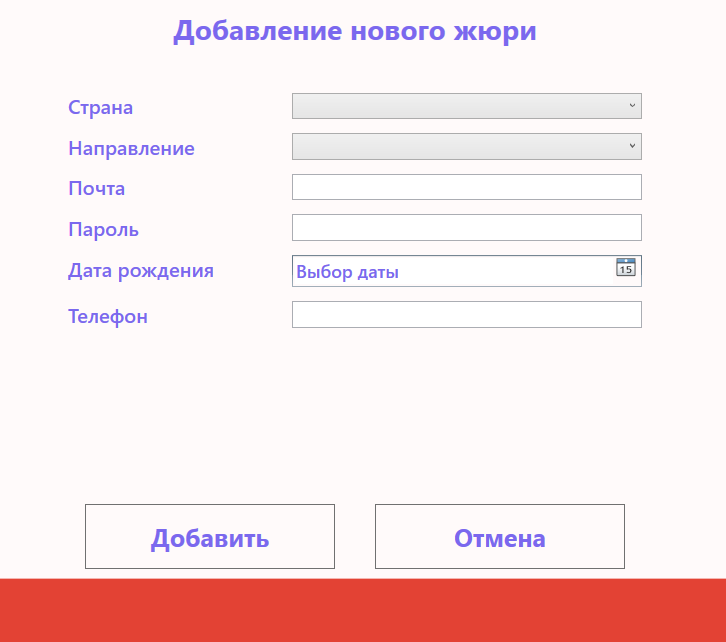


Рис.7 «Добавление жюри»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Добавление нового жюри" Style="{StaticResource StyleTxtSubtitle}" Name="Titletxt"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" Width="600" Margin="15">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Страна" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBCountry" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdCountry}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Направление" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBDiric" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdDirictory}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Почта" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtEmail" Text="{Binding Email}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Пароль" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtPassword" Text="{Binding Password}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Дата рождения" Style="{StaticResource StyleTxt}" Width="210"/>

<DatePicker x:Name="datePicker" IsDropDownOpen="False" SelectedDateFormat="Short" SelectedDate="{Binding DateBirthday}" Style="{StaticResource StyleDatePicker}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Телефон" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="Txtphone" Text="{Binding Phone}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Vertical" VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" Grid.Row="2" >

<Button Style="{StaticResource MenuStyleBtn2}" Content="Добавить" Name="BtnAdd" Click="BtnAdd\_Click" Margin="20"/>

<Button Style="{StaticResource MenuStyleBtn2}" Content="Отмена" Name="BtnCancel" Click="BtnCancel\_Click" Margin="20"/>

</StackPanel>

</StackPanel>

</Grid>

Страница «Организаторы»



Рис.8 «Организаторы»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1.2\*"/>

<RowDefinition Height="7\*"/>

</Grid.RowDefinitions>

<StackPanel Background="#E34234">

<TextBlock Text="Организаторы" HorizontalAlignment="Center" Background="#E34234" Style="{StaticResource StyleTxtSubtitle}" Foreground="White"/>

</StackPanel>

<DataGrid Name="dtgOrganiz" Style="{StaticResource StyleDataGrid}" Grid.Row="1">

<DataGrid.Columns>

<DataGridTemplateColumn Width="auto" Header="Фото">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Width="90" Stretch="UniformToFill" HorizontalAlignment="Left" Margin="5">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>

\Resources\foto14.jpg

</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="ФИО" Width="0.5\*" Binding="{Binding Fio}"/>

<DataGridTextColumn Header="Почта" Width="0.5\*" Binding="{Binding Email}"/>

<DataGridTextColumn Header="Пароль" Width="0.5\*" Binding="{Binding Password}"/>

<DataGridTextColumn Header="Страна" Width="0.5\*" Binding="{Binding Country.NameCountry}"/>

<DataGridTextColumn Header="День рождения" Width="0.5\*" Binding="{Binding DateBirthday, StringFormat={}{0:dd/MM/yyyy}}"/>

<DataGridTextColumn Header="Телефон" Width="0.5\*" Binding="{Binding Phone}"/>

<DataGridTextColumn Header="Пол" Width="0.5\*" Binding="{Binding Pol}"/>

</DataGrid.Columns>

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Добавить" Name="MenuAdd" Click="MenuAdd\_Click"/>

<MenuItem Header="Редактировать" Name="MenuEdit" Click="MenuEdit\_Click"/>

<MenuItem Header="Обновить" Name="MenuUpdate" Click="MenuUpdate\_Click"/>

<Separator/>

<MenuItem Header="Удалить" Name="MenuDel" Click="MenuDel\_Click"/>

<MenuItem Header="Главное меню" Name="MenuGlav" Click="MenuGlav\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

</DataGrid>

</Grid>

Страница добавления «Организаторы»

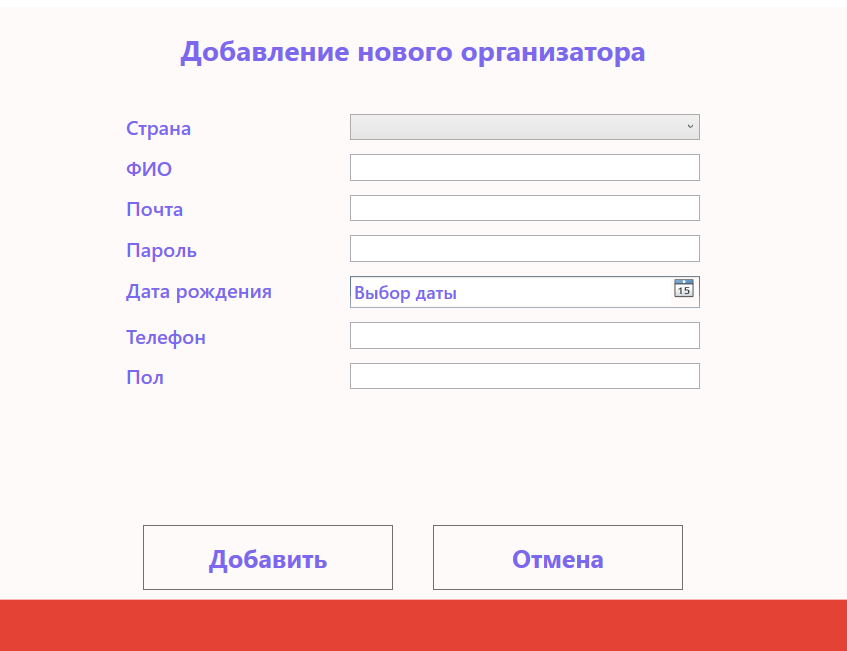


Рис.9 «Добавление организатора»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Добавление нового организатора" Style="{StaticResource StyleTxtSubtitle}" Name="Titletxt"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" Width="600" Margin="15">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Страна" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBCountry" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdCountry}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="ФИО" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="Txtfio" Text="{Binding Fio}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Почта" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtEmail" Text="{Binding Email}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Пароль" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtPassword" Text="{Binding Password}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Дата рождения" Style="{StaticResource StyleTxt}" Width="210"/>

<DatePicker x:Name="datePicker" IsDropDownOpen="False" SelectedDateFormat="Short" SelectedDate="{Binding DateBirthday}" Style="{StaticResource StyleDatePicker}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Телефон" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="Txtphone" Text="{Binding Phone}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Пол" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtPol" Text="{Binding Pol}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Vertical" VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" Grid.Row="2" >

<Button Style="{StaticResource MenuStyleBtn2}" Content="Добавить" Name="BtnAdd" Click="BtnAdd\_Click" Margin="20"/>

<Button Style="{StaticResource MenuStyleBtn2}" Content="Отмена" Name="BtnCancel" Click="BtnCancel\_Click" Margin="20"/>

</StackPanel>

</StackPanel>

</Grid>

Страница «Модераторы»



Рис.10 «Модераторы»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1.2\*"/>

<RowDefinition Height="7\*"/>

</Grid.RowDefinitions>

<StackPanel Background="#E34234">

<TextBlock Text="Модераторы" HorizontalAlignment="Center" Background="#E34234" Style="{StaticResource StyleTxtSubtitle}" Foreground="White"/>

</StackPanel>

<DataGrid Name="dtgModerator" Style="{StaticResource StyleDataGrid}" Grid.Row="1">

<DataGrid.Columns>

<DataGridTemplateColumn Width="auto" Header="Фото">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Width="90" Stretch="UniformToFill" HorizontalAlignment="Left" Margin="5">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>

\Resources\foto14.jpg

</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Header="ФИО" Width="0.5\*" Binding="{Binding FIO}"/>

<DataGridTextColumn Header="Почта" Width="0.5\*" Binding="{Binding Email}"/>

<DataGridTextColumn Header="Пароль" Width="0.5\*" Binding="{Binding Password}"/>

<DataGridTextColumn Header="Страна" Width="0.5\*" Binding="{Binding Country.NameCountry}"/>

<DataGridTextColumn Header="День рождения" Width="0.5\*" Binding="{Binding DateBirthday, StringFormat={}{0:dd/MM/yyyy}}"/>

<DataGridTextColumn Header="Телефон" Width="0.5\*" Binding="{Binding Phone}"/>

<DataGridTextColumn Header="Направление" Width="0.5\*" Binding="{Binding Dirictory.IdDirictory}"/>

<DataGridTextColumn Header="Пол" Width="0.5\*" Binding="{Binding Pol}"/>

<DataGridTextColumn Header="Событие модератора" Width="0.5\*" Binding="{Binding Eventmoder.NameMod}"/>

</DataGrid.Columns>

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Добавить" Name="MenuAdd" Click="MenuAdd\_Click"/>

<MenuItem Header="Редактировать" Name="MenuEdit" Click="MenuEdit\_Click"/>

<MenuItem Header="Обновить" Name="MenuUpdate" Click="MenuUpdate\_Click"/>

<Separator/>

<MenuItem Header="Удалить" Name="MenuDel" Click="MenuDel\_Click"/>

<MenuItem Header="Глдавное меню" Name="MenuGlav" Click="MenuGlav\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

</DataGrid>

</Grid>

Страница добавления «Модераторы»

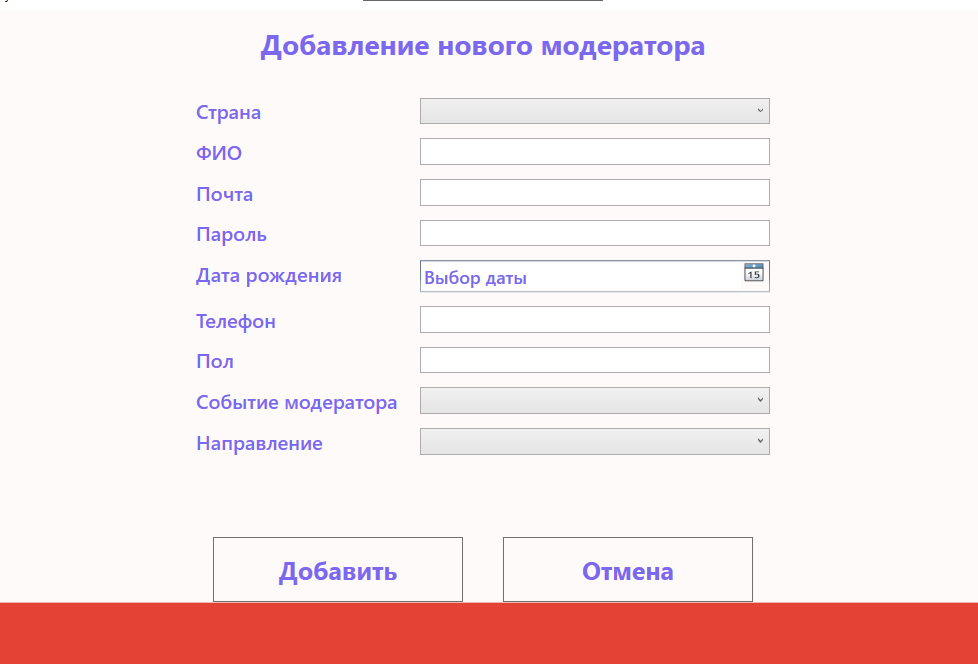


Рис.11 «Добавление модератора»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="7\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Добавление нового модератора" Style="{StaticResource StyleTxtSubtitle}" Name="Titletxt"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" Width="600" Margin="15">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Страна" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBCountry" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdCountry}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="ФИО" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="Txtfio" Text="{Binding Fio}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Почта" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtEmail" Text="{Binding Email}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Пароль" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtPassword" Text="{Binding Password}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Дата рождения" Style="{StaticResource StyleTxt}" Width="210"/>

<DatePicker x:Name="datePicker" IsDropDownOpen="False" SelectedDateFormat="Short" SelectedDate="{Binding DateBirthday}" Style="{StaticResource StyleDatePicker}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Телефон" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="Txtphone" Text="{Binding Phone}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Пол" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtPol" Text="{Binding Pol}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Событие модератора" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBEventModer" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdEventmoder}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Направление" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBDirictory" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdDirictory}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Vertical" VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" Grid.Row="2" >

<Button Style="{StaticResource MenuStyleBtn2}" Content="Добавить" Name="BtnAdd" Click="BtnAdd\_Click" Margin="20"/>

<Button Style="{StaticResource MenuStyleBtn2}" Content="Отмена" Name="BtnCancel" Click="BtnCancel\_Click" Margin="20"/>

</StackPanel>

</StackPanel>

</Grid>

Страница «Активность»



Рис.12 «Активность»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1.2\*"/>

<RowDefinition Height="7\*"/>

</Grid.RowDefinitions>

<StackPanel Background="#E34234">

<TextBlock Text="Активность" HorizontalAlignment="Center" Background="#E34234" Style="{StaticResource StyleTxtSubtitle}" Foreground="White"/>

</StackPanel>

<DataGrid Name="dtgActiv" Style="{StaticResource StyleDataGrid}" Grid.Row="1">

<DataGrid.Columns>

<DataGridTextColumn Header="Событие" Width="0.5\*" Binding="{Binding Event.NameEvent}"/>

<DataGridTextColumn Header="Модератор" Width="0.5\*" Binding="{Binding Moderator.FIO}"/>

<DataGridTextColumn Header="Жюри" Width="0.5\*" Binding="{Binding Juri.Email}"/>

<DataGridTextColumn Header="Дни" Width="0.5\*" Binding="{Binding Days}"/>

<DataGridTextColumn Header="Доп. Событие" Width="0.5\*" Binding="{Binding ActivName}"/>

<DataGridTextColumn Header="Время" Width="0.5\*" Binding="{Binding Time}"/>

<DataGridTextColumn Header="Победитель" Width="0.5\*" Binding="{Binding Person.Fio}"/>

</DataGrid.Columns>

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Добавить" Name="MenuAdd" Click="MenuAdd\_Click"/>

<MenuItem Header="Редактировать" Name="MenuEdit" Click="MenuEdit\_Click"/>

<MenuItem Header="Обновить" Name="MenuUpdate" Click="MenuUpdate\_Click"/>

<Separator/>

<MenuItem Header="Удалить" Name="MenuDel" Click="MenuDel\_Click"/>

<MenuItem Header="Главное меню" Name="MenuGlav" Click="MenuGlav\_Click"/>

<MenuItem Header="Вывод данных в Word" Name="MenuWord" Click="MenuWord\_Click"/>

<MenuItem Header="Вывод данных в Excel" Name="MenuExcel" Click="MenuExcel\_Click"/>

</ContextMenu>

</DataGrid.ContextMenu>

</DataGrid>

</Grid>

Страница добавления «Активность»

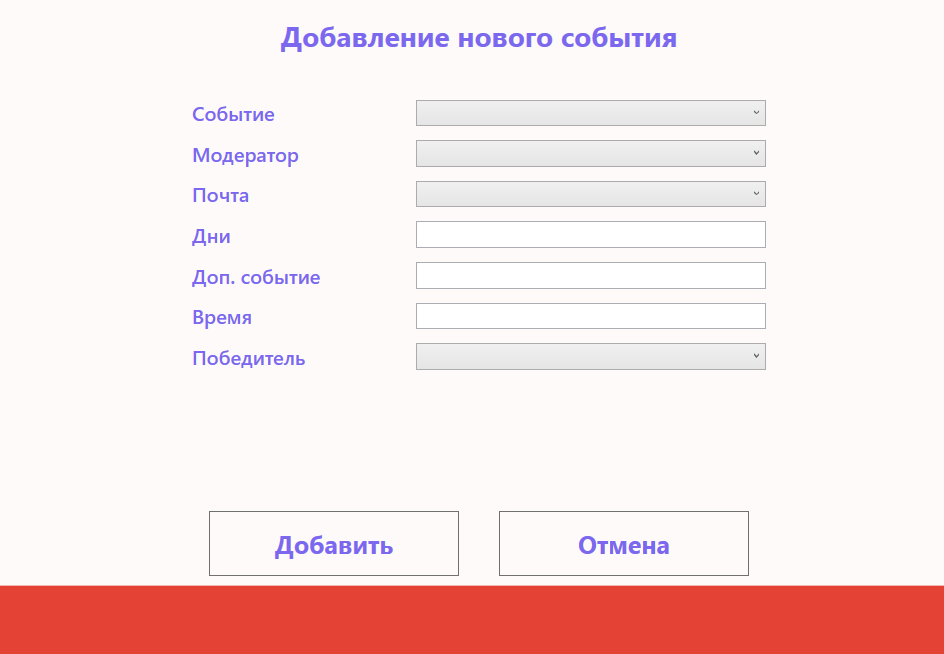


Рис.13 «Добавление активности»

Код:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Orientation="Vertical" Grid.Row="0" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Добавление нового события" Style="{StaticResource StyleTxtSubtitle}" Name="Titletxt"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" Width="600" Margin="15">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Событие" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBEvent" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdEvent}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Модератор" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBModer" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdModerator}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Почта" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBJuri" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdJuri}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Дни" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="Txtdays" Text="{Binding Days}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Доп. событие" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="ActivName" Text="{Binding ActivName}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Время" Style="{StaticResource StyleTxt}" Width="210"/>

<TextBox Name="TxtTime" Text="{Binding Time}" Style="{StaticResource StyleTxb}"/>

</StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<TextBlock Text="Победитель" Style="{StaticResource StyleTxt}" Width="210"/>

<ComboBox Name="CMBPerson" Width="350" Margin="7" FontSize="14"

SelectedValue="{Binding IdPerson}" Style="{StaticResource StyleCmb}"/>

</StackPanel>

</StackPanel>

<StackPanel Grid.Row="2" Orientation="Vertical" VerticalAlignment="Center" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" Grid.Row="2" >

<Button Style="{StaticResource MenuStyleBtn2}" Content="Добавить" Name="BtnAdd" Click="BtnAdd\_Click" Margin="20"/>

<Button Style="{StaticResource MenuStyleBtn2}" Content="Отмена" Name="BtnCancel" Click="BtnCancel\_Click" Margin="20"/>

</StackPanel>

</StackPanel>

</Grid>

# Алгоритм работы приложения

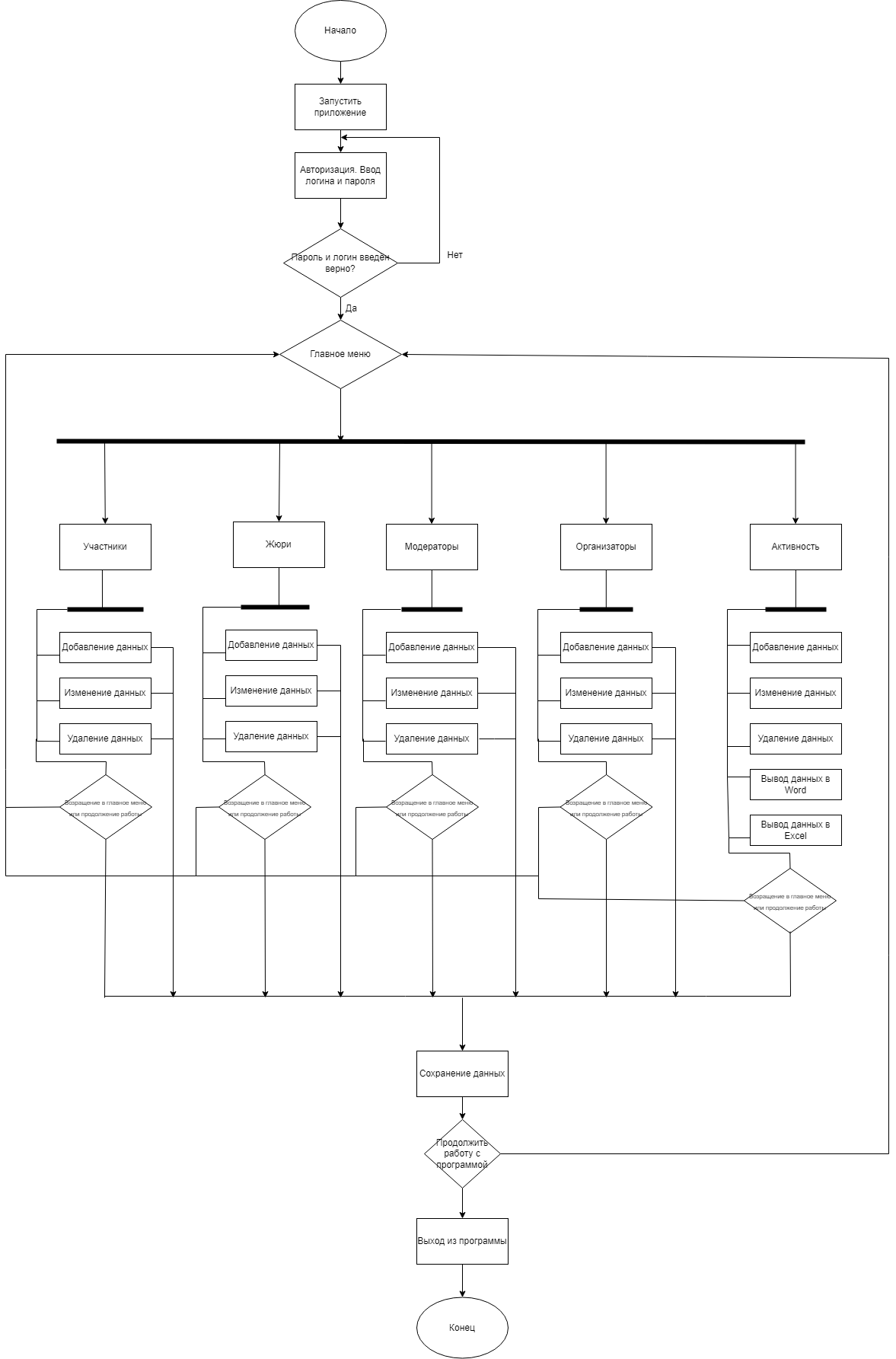


Рис.14 «Блок схема работы приложения»

# Проектирование ER-диаграммы

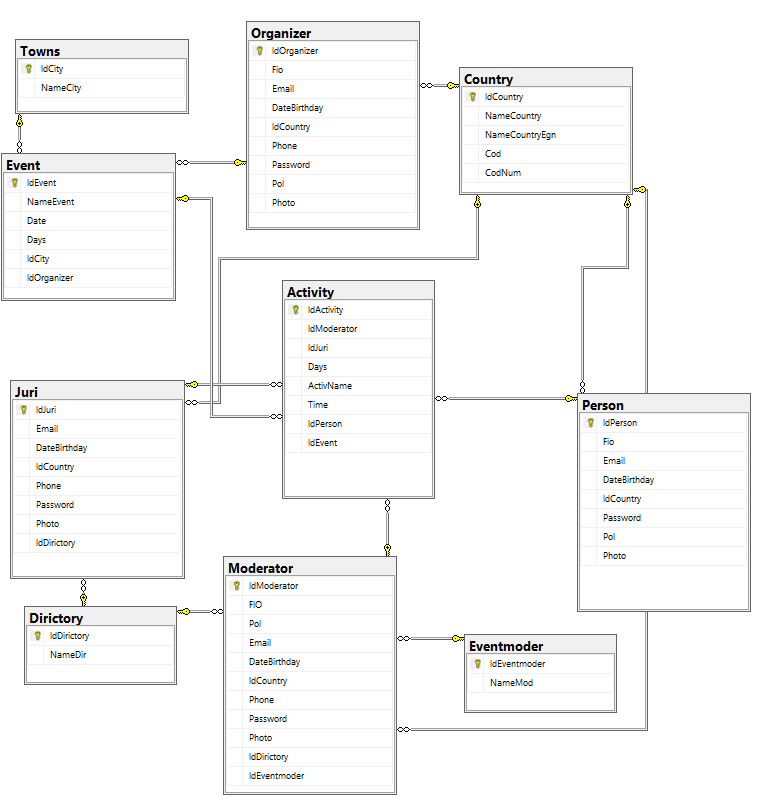


Рис.15 «ER- диаграмма»

# Проектирование и разработка базы данных

1. Запустили приложение Microsoft SQL Server
2. Создали базу данных «Holiday»

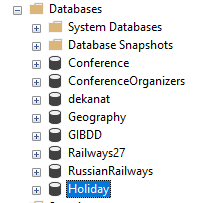


Рис.16 «Holiday»

1. Создаём таблицы и задаем им атрибуты

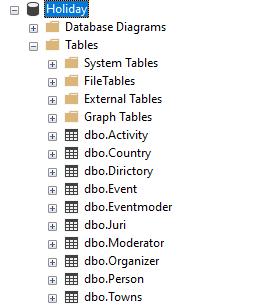


Рис.17 «Таблицы БД»

1. Создаём диаграмму «dbo.Diagram» и связываем таблицы между собой

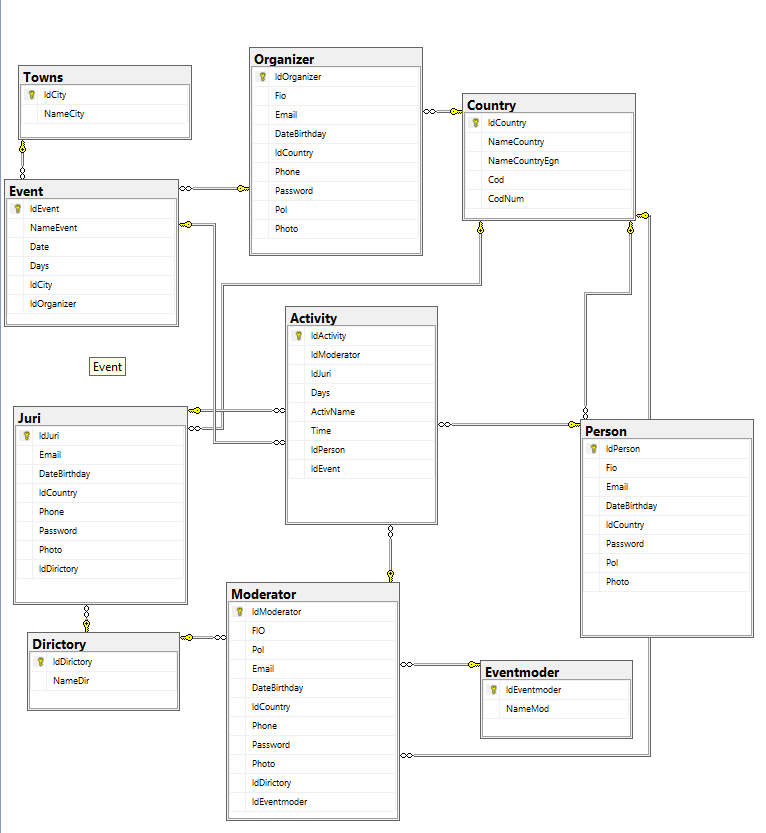


Рис.18 «Схема базы данных»

1. Заполнением данными таблицы с помощью импорта данных с Excel
2. Правой кнопкой мыши нажимаем на базу данных «Holiday». Выбираем «задачи» потом «импорт данных»

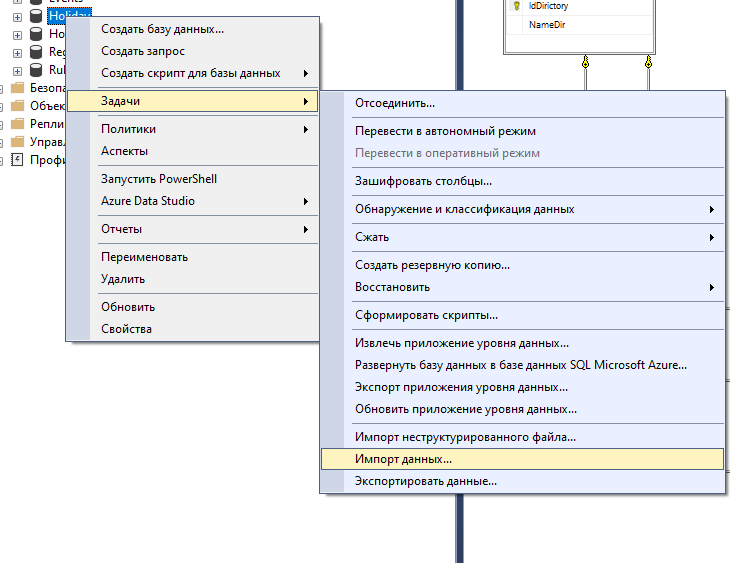


Рис.19 «Импорт данных»

1. Нажимаем «Далее»

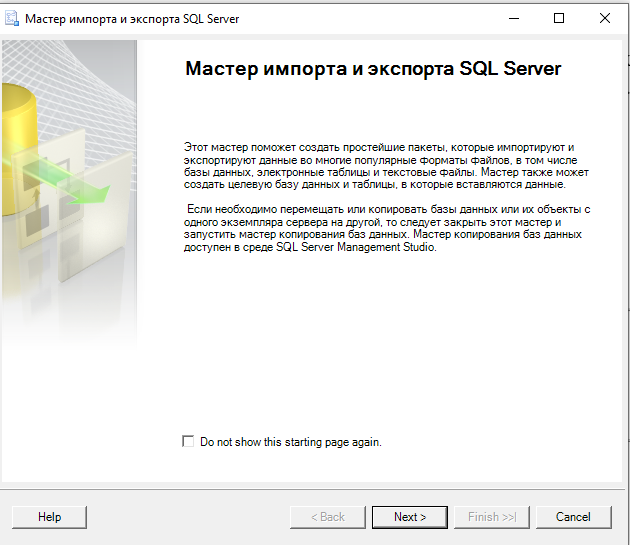


Рис.20 «Мастер импорта»

1. Выбираем источник данных Excel, выбираем путь к файлу Excel из которой берутся данные и выбираем версию Excel 2016

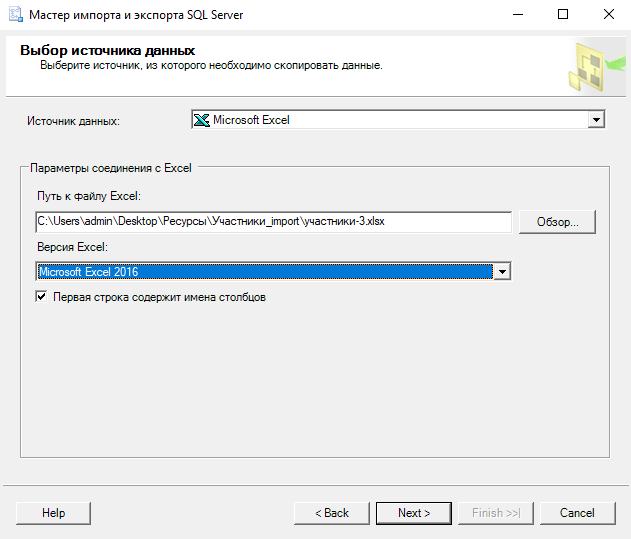


Рис.21 «Microsoft Excel»

1. Выбираем назначение «SQL Server»

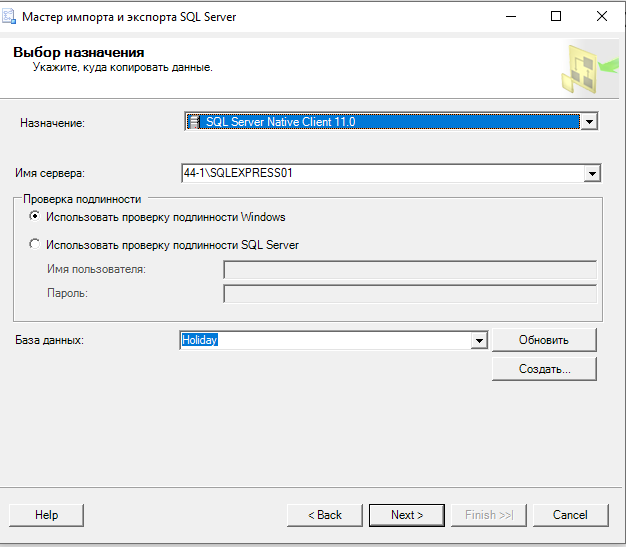


Рис.22 «SQL Server»

1. Нажимаем «Далее»

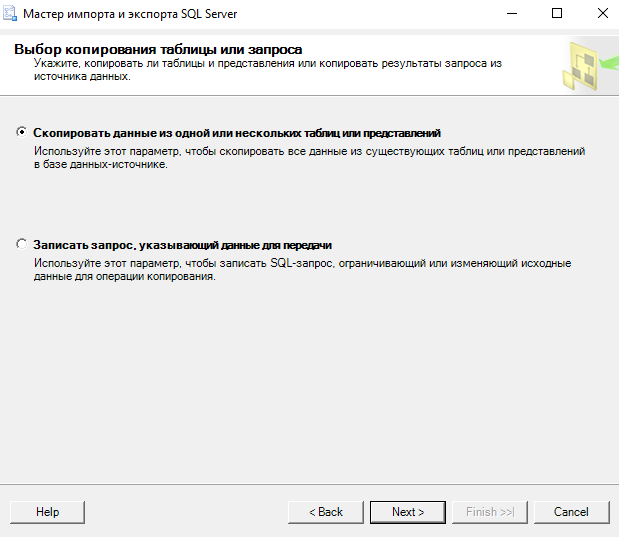


Рис.23 «Выбор копирования»

1. Выбираем нужную таблицу и изменяем нужные значения

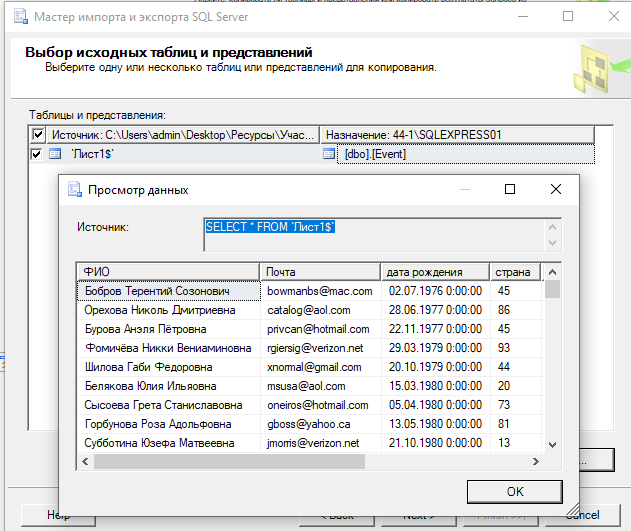


Рис.24 «Выбор исходных таблиц»

1. Нажимаем кнопку «Next» и пропускаем ошибки.

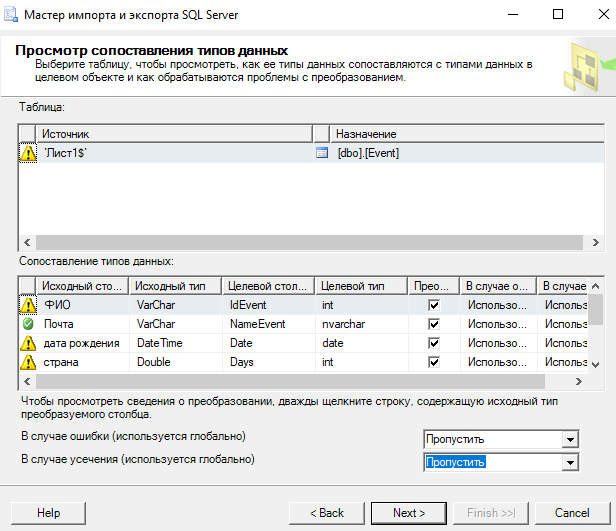


Рис.25 «Просмотр составления типоав данных»

1. Нажимаем кнопку «Finish»

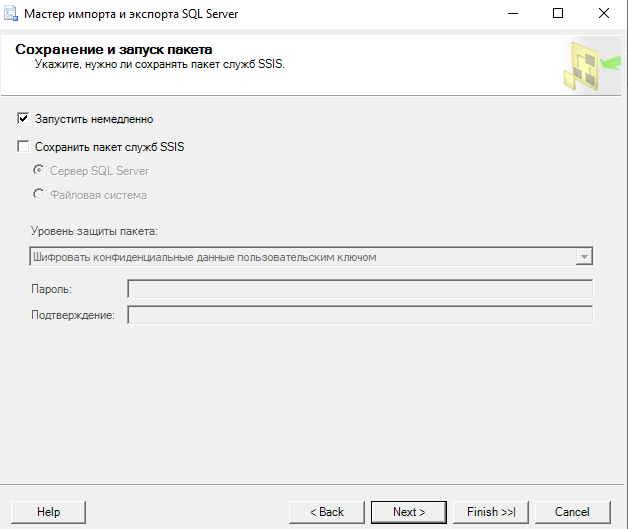


Рис.26 «Сохранение и запуск пакета»

Запросы:

1)Запрос вывода данных из Городов выбираем название и идентификатор

SELECT [IdCity]

,[NameCity]

FROM [Holiday].[dbo].[Towns]



Рис.27 «Запрос вывода данных»

2)Запрос вывода данных с условием из таблицы Городов выбираем конкретный город

SELECT [NameCity] FROM [dbo].[Towns] WHERE [NameCity] = 'Абакан'

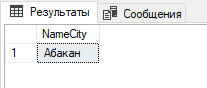


Рис.28 «Запрос вывода данных с условием»

3)Запрос вывода данных исключение повтора данных из таблицы сотрудники выводится пол

SELECT DISTINCT [Pol]

FROM [dbo].[Person]

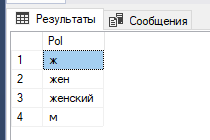


Рис.29 «Запрос выводы данных исключение»

4)Запрос вывода данных с условием что идентификаторы выводятся меньше трёх

SELECT [NameDir] FROM [dbo].[Dirictory] WHERE [IdDirictory] < 3

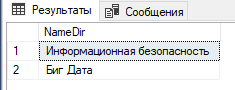


Рис.30 «Запрос вывода данных с условием»

5)Запрос вывода данных с сортировкой по дням

SELECT \* FROM [dbo].[Event] ORDER BY [Days];

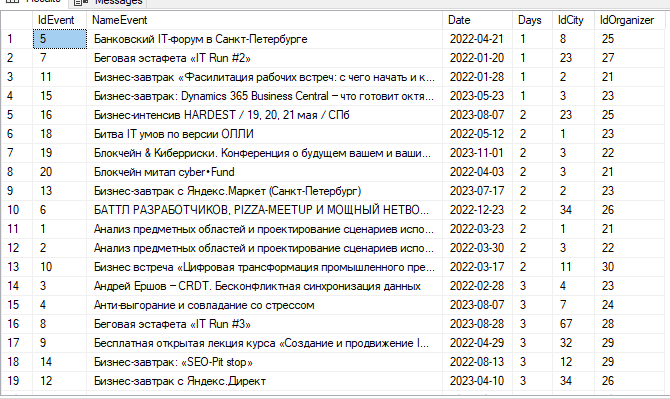


Рис.31 «Запрос вывода данных с сортировкой»

**Входные и выходные данные**

Таблица №2 «Словарь данных»

|  |  |  |  |
| --- | --- | --- | --- |
| **ключ** | **поле** | **обязательное** | **примечание** |
| **Активность** | | | |
| Первичный | IdActivity | Да | Идентификатор |
| Внешний | IdModerator | Да | Внешний ключ к таблице модераторы |
| Внешний | IdJuri | Да | Первичный ключ к таблице жюри |
|  | ActivName | Да | Доп. событие |
|  | Time | Да | Время начала |
| Внешний | IdPerson | Да | Первичный ключ к таблице участники |
|  | Days | Да | Дни |
| Внешний | IdEvent | Да | Первичный ключ к таблице событие |
| **Направление** | | | |
| Первичный | IdDirictory | Да | Идентификатор |
|  | NameDir | Да | Название направления |
| **Событие** | | | |
| Первичный | IdEvent | Да | Идентификатор |
|  | NameEvent | Да | Название событие |
|  | Date | Да | Дата |
|  | Days | Да | Дни |
| Внешний | IdCity | Да | Внешний ключ к таблице города |
| Внешний | IdOrganizer | Да | Внешний ключ к таблице организаторы |
| **Событие модераторов** | | | |
| Первичный | IdEventmoder | Да | Идентификатор |
|  | NameMod | Да | Название событие |
| **Жюри** | | | |
| Первичный | IdJuri | Да | Идентификатор |
|  | Email | Да | Почта |
|  | DateBirthday | Да | День рождение |
| Внешний | IdCountry | Да | Внешний ключ к таблице страны |
|  | Phone | Да | Телефон |
|  | Password | Да | Пароль |
|  | Photo | Да | Фото |
| Внешний | IdDirictory | Да | Внешний ключ к таблице направление |
| **Модераторы** | | | |
| Первичный | IdModerator | Да | Идентификатор |
|  | FIO | Да | ФИО |
|  | Pol | Да | Пол |
|  | Email | Да | Почта |
|  | DateBirthday | Да | День рождения |
| Внешний | IdCountry | Да | Внешний ключ к стране |
|  | Phone | Да | Телефон |
|  | Password | Да | Пароль |
|  | Photo | Да | Фото |
| Внешний | IdDirictory | Да | Внешний ключ к таблице направления |
| Внешний | IdEventmoder | Да | Внешний ключ к таблице событие модератора |
| **Организаторы** | | | |
| Первичный | IdOrganizer | Да | Идентификатор |
|  | Fio | Да | ФИО |
|  | Email | Да | Почта |
|  | DateBirthday | Да | День рождения |
| Внешний | IdCountry | Да | Внешний ключ к таблице страны |
|  | Phone | Да | Телефон |
|  | Password | Да | Пароль |
|  | Pol | Да | Пол |
|  | Photo | Да | Фото |
| **Участники** | | | |
| Первичный | IdPerson | Да | Идентификатор |
|  | Fio | Да | ФИО |
|  | Email | Да | Почта |
|  | DateBirthday | Да | Дата рождения |
| Внешний | IdCountry | Да | Внешний ключ к таблице страны |
|  | Password | Да | Пароль |
|  | Pol | Да | Пол |
|  | Photo | Да | Фото |
| **Города** | | | |
| Первичный | IdCity | Да | Идентификатор |
|  | NameCity | Да | Название города |
| **Страны** | | | |
| Первичный | IdCountry | Да | Идентификатор |
|  | NameCountry | Да | Название страны |
|  | NameCountryEgn | Да | Название страны на иностранном языке |
|  | Cod | Да | Код |
|  | CodNum | Да | Номер кода |

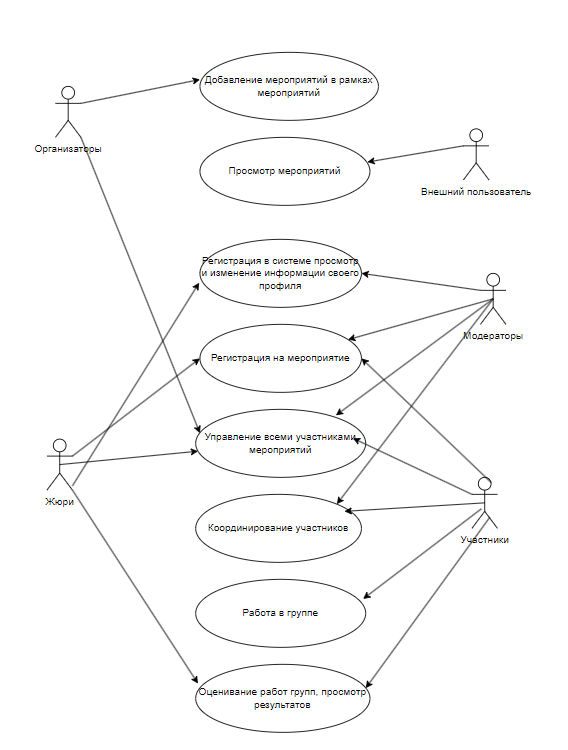


Рис.32 «Диаграмма прецедентов»

# Руководство системного программиста

1. **ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ**
   1. **Назначение и функции программы**

Программа предназначена для организаторов конференций информационной безопасности.

Функции программы:

* Добавление, редактирование и удаление данных из таблиц:
  + - «Участники»
    - «Модераторы»
    - «Организаторы»
    - «Жюри»
    - «Активность»
* Вывод отчета в Word о мероприятиях
* Авторизация для организаторов
* Экспорт списка победителей в формате Excel.
  1. **Сведения о технических и программных средствах, обеспечивающие выполнение программы**
     1. **Требования к программному обеспечению**

Для корректной работы программы необходимо: ОС Windows 10 и выше, Microsoft Excel 2016, Microsoft Word 2016 (и выше) и Microsoft SQL Management Studio.

* + 1. **Требования к техническим средствам**

Таблица №3 «Минимальные требования»

|  |  |
| --- | --- |
| Процессор | Intel(R) Core (TM) i5-4460 CPU @ 3.20GHz |
| Оперативная Память | 8 ГБ, DDR4, DIMM, 2133 МГц; |
| Разрешение экрана | 1920x1080 |
| Размер экрана | От 20 дюймов |
| Устройства ввода | Мышь, клавиатура |
| Дисковое пространство | 1 Гб |
| Подключаемые ус-ва | Принтер |
| Операционная система | Windows 10 и выше |

1. **СТРУКТУРА ПРОГРАММЫ**
   1. **Структура программы и её составные части**

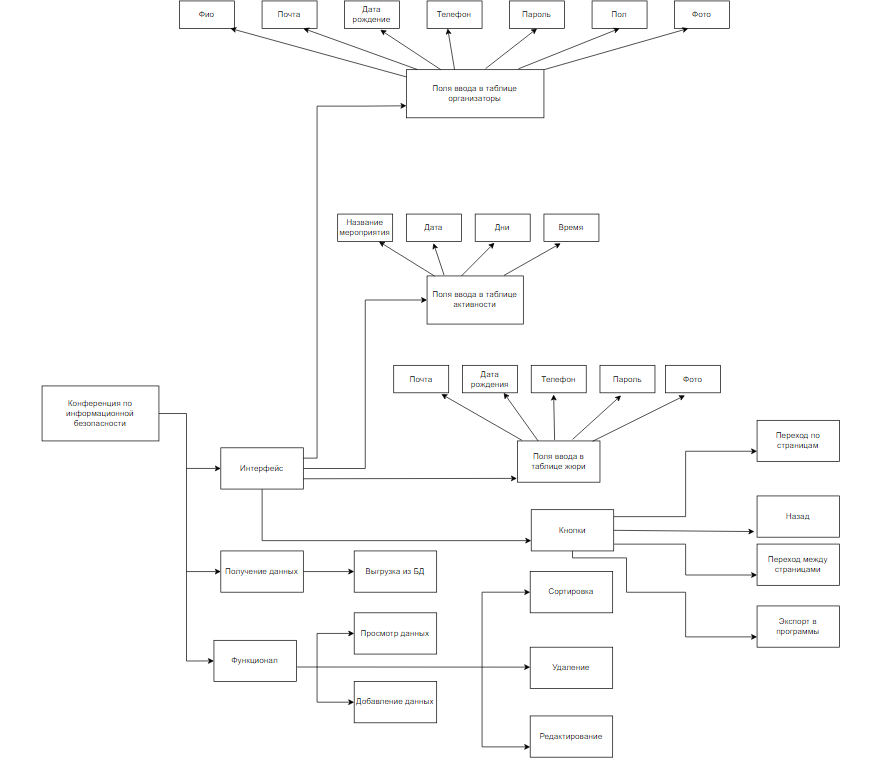


Рис.33 «Структура программы»

* 1. **Связи между составными частями программы**

Сначала пользователь переходит на страницу авторизации. После успешной авторизации, пользователь перейдет на страницу «Главное меню». С «Главного меню» он может перейти на такие страницы как «Участники», «Модераторы», «Жюри», «Организаторы», «Активность». Если нужно вернуться в «Главное меню» надо нажать ПКМ по таблице и выбрать в контекстном меню функцию «Главное меню».

* 1. **Связи программы с другими программами**

Чтобы сохранять данные, нужно экспортировать в следующие программы: Microsoft Excel и Microsoft Word. И для сохранения базы данных использую приложение SQL Server Management Studio.

1. **НАСТРОЙКА ПРОГРАММЫ И СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ**
   1. **Настройка работы БД в SQL Server Management Studio**
2. Находим Management Studio в меню «ПУСК», запускаем.
3. Открывается окно соединения с сервером SQL Server, где выбираем:

— Ядро СУБД в поле «Тип сервера»;

– имя сервера, указанное вами при установке в поле имя сервера;

— Проверка подлинности Windows в поле «Проверка подлинности».

1. Нажимаем «Соединить».

В результате Management Studio подключится к SQL Server, и произойдет открытие основного окна программы. Обязательно изучите такой элемент, как «Обозреватель объектов» — он даёт возможность работать со всеми структурными элементами БД на сервере посредством интерфейса, похожего на проводник Windows.

* 1. **Сохранение БД**

1. Создаём схему БД в SQL Server Management Studio, состоящую из таблиц.
2. Заполняем таблицы данными.
3. БД можно сохранить с помощью резервной копии БД или с помощью скрпта.

**Создание резервной копии БД:**

1. Слева в списке находим свою БД и нажимаем по ней ПКМ.

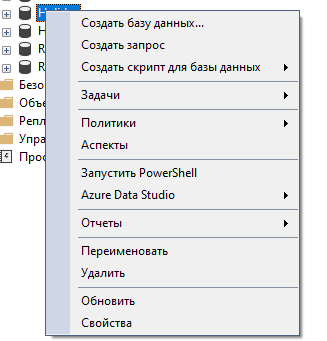


Рис.34 «Контекстное меню»

1. Нажимаем на пункт «Задачи» и в раскрывающимся списке выбираем пункт «Создать резервную копию».

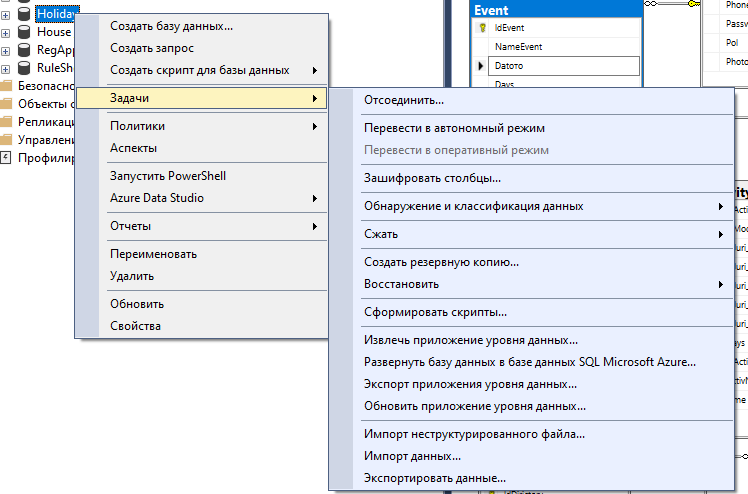


Рис.35 «Создание копии»

1. В открывшемся окне оставляем настройки сохранения по стандарту и нажимаем ОК.

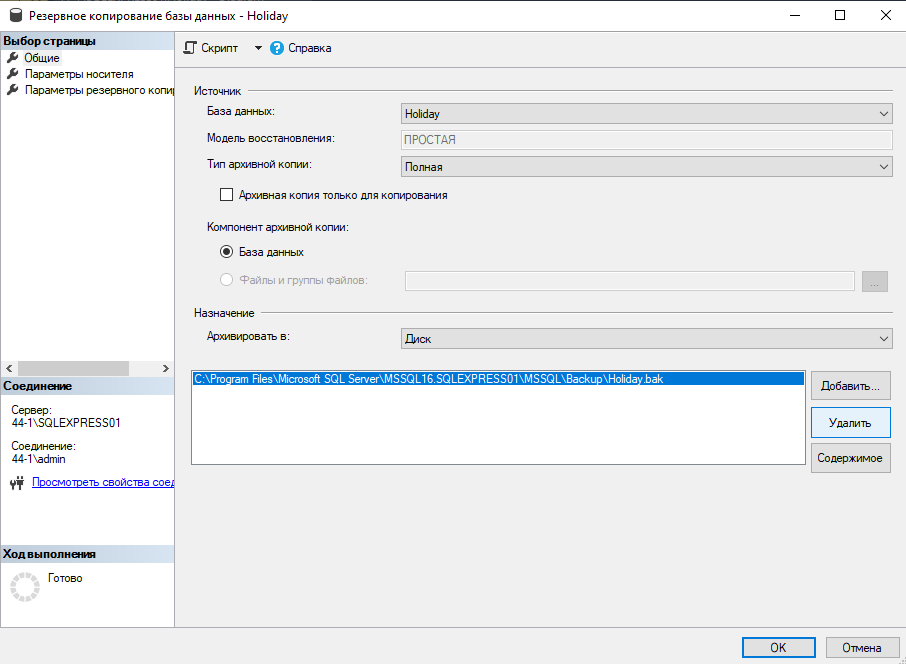


Рис.36 «Настройка сохранения копии»

1. Далее в папке куда была сохранена копия находим свою БД с расширением .bak.



Рис.37 «Резервная копия БД»

**Создание скрипта БД:**

1. Нажимаем ПКМ по своей БД – пункт «Задачи» - пункт «Сформировать скрипты».

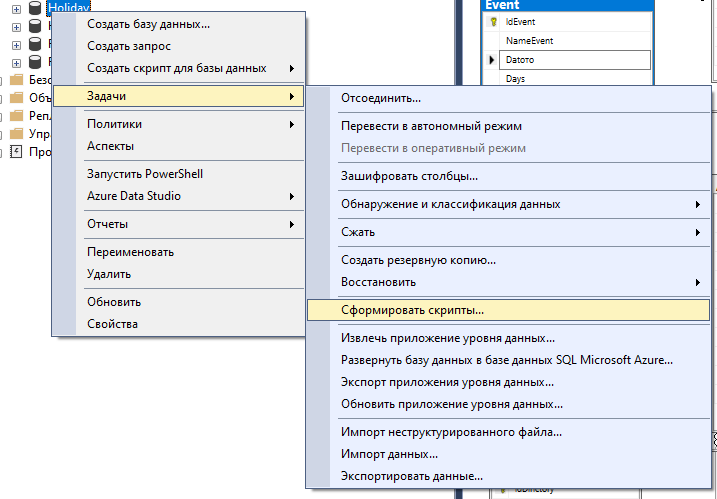


Рис.38 «Создание скрипта»

1. В открывшемся окне выбираем переключатель «Создать скрипт для всей БД и всех её объектов» и нажимаем Далее.

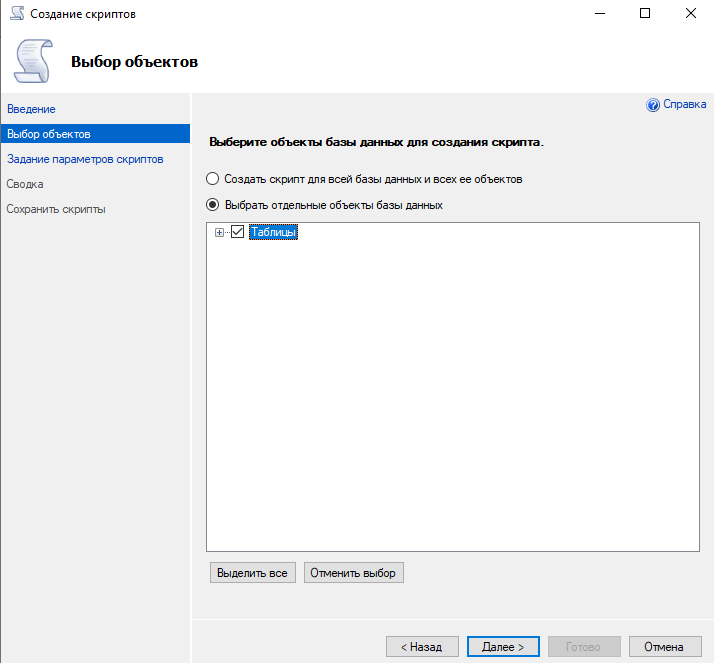


Рис.39 «Выбор объектов»

1. В следующем окне выбираем переключатель «Сохранить как файл скрипта» и оставляем все стандартные настройки, кроме поля «Имя файла», в этом поле мы выбираем путь сохранения скрипта и имя скрипта, и нажимаем Далее.

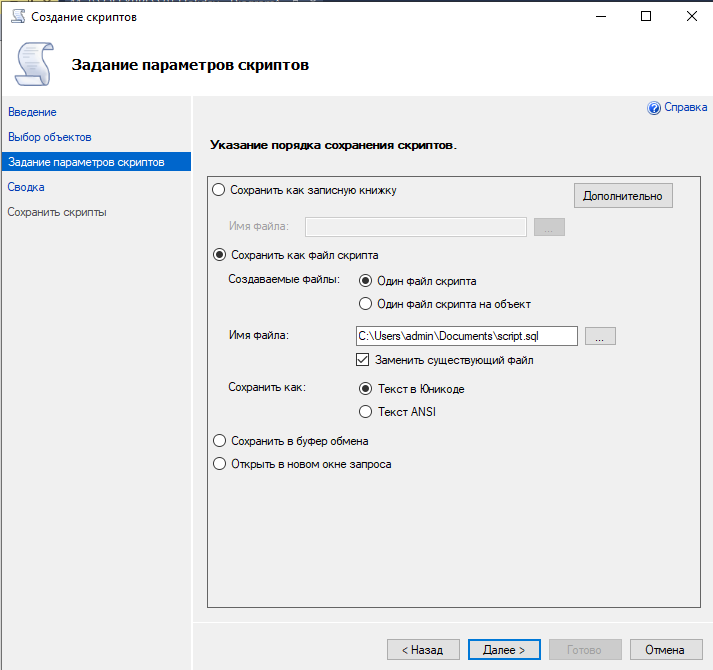


Рис.40 «Задание параметров скрипта»

1. В окне «Сводка» жмём Далее.

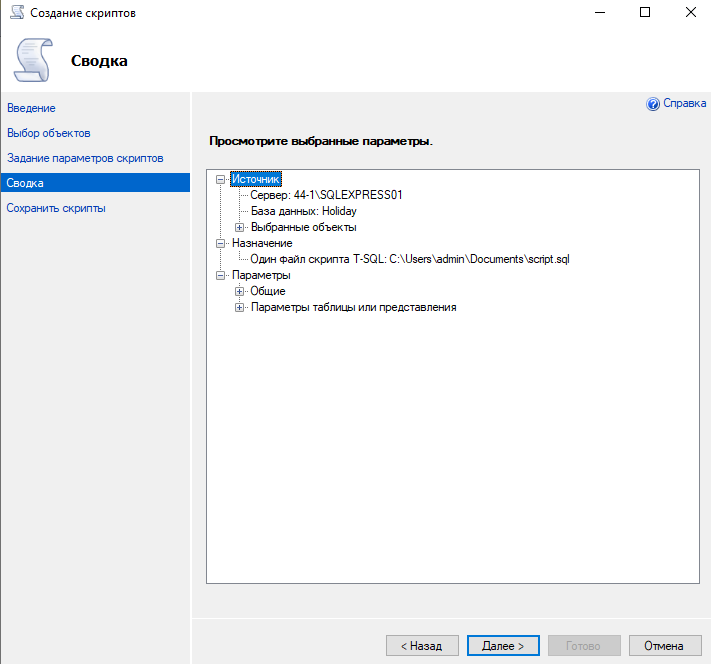


Рис.41 «Сводка»

1. В окне «Сохранить скрипты» ждём создания и сохранения скрипта и нажимаем Готово.

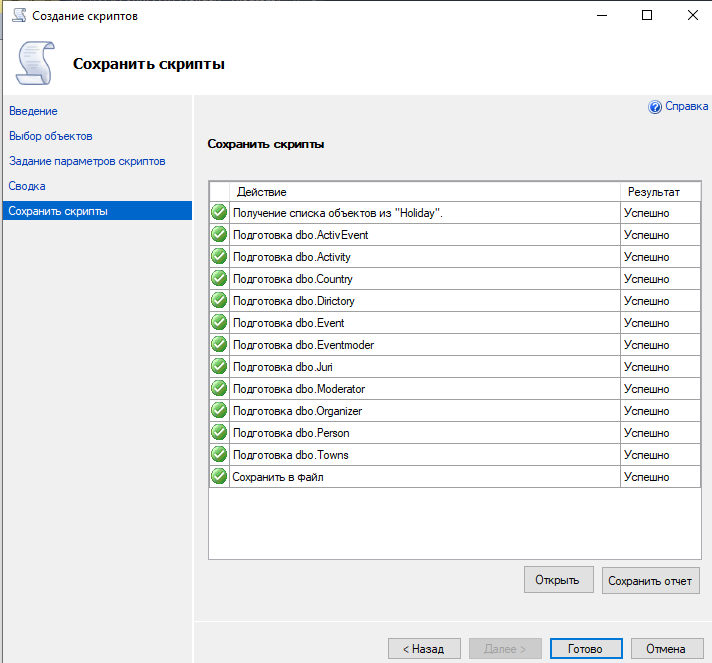


Рис.42 «Сохранение скрипта»

1. Находим скрипт БД с расширением .sql.



Рис.43 «Скрипт БД»

* 1. **Подключение БД к программе**

1. Запускаем Visual Studio Community – создаём проект .NET Framework – создаём справа в обозревателе решений папку где будет хранится БД.

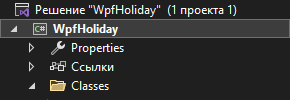


Рис.44 «Создание папки в программе»

1. Нажимаем ПКМ по папке – пункт «Добавить» - пункт «Создать элемент».

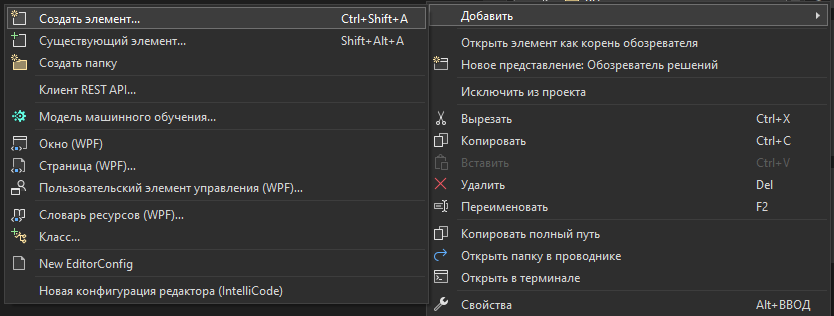


Рис.45 «Добавление элемента в папку»

1. В открывшемся окне слева выбираем пункт «Данные» и там находим пункт «Модель ADO.NET EDM» и снизу в поле ввода задаём ей имя, и нажимаем Добавить.

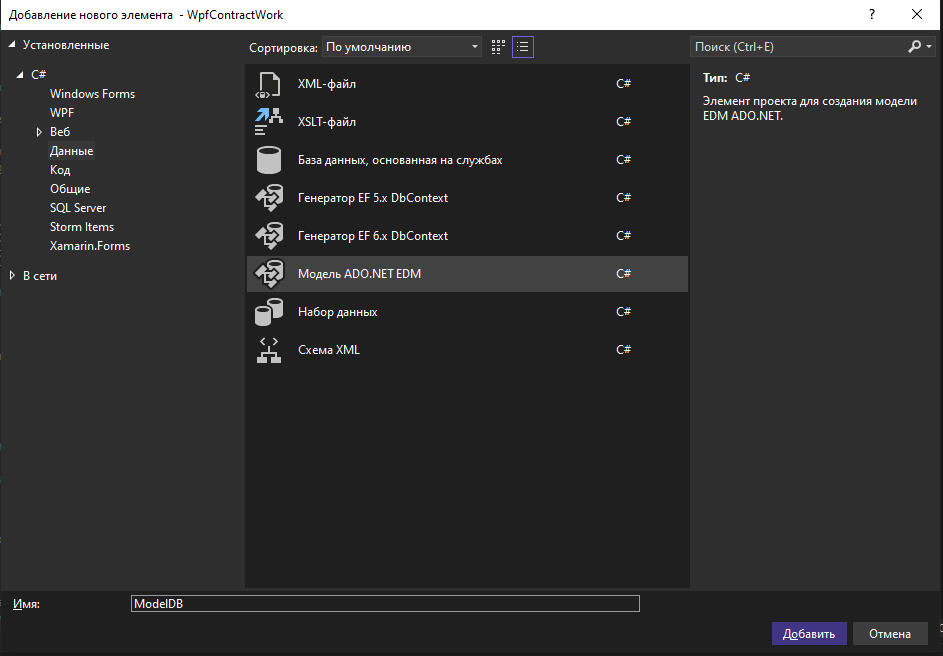


Рис.47 «Создание модели БД»

1. В открывшемся окне выбираем «Конструктор EF из БД» и жмём Далее.

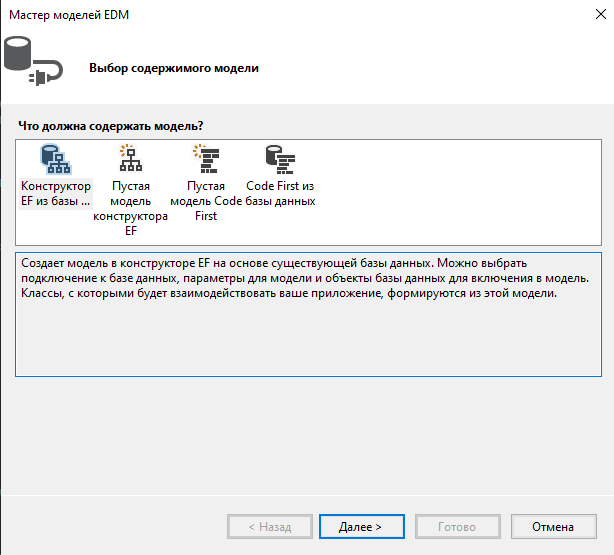


Рис.48 «Выбор модели»

1. Нажимаем «Создать соединение» и в строке «Имя сервера» ставим название своего сервера из приложения SQL Server Management Studio(нажав в приложении на свойства и скопировав имя сервера), и выбираем свою БД, далее жмём ОК.

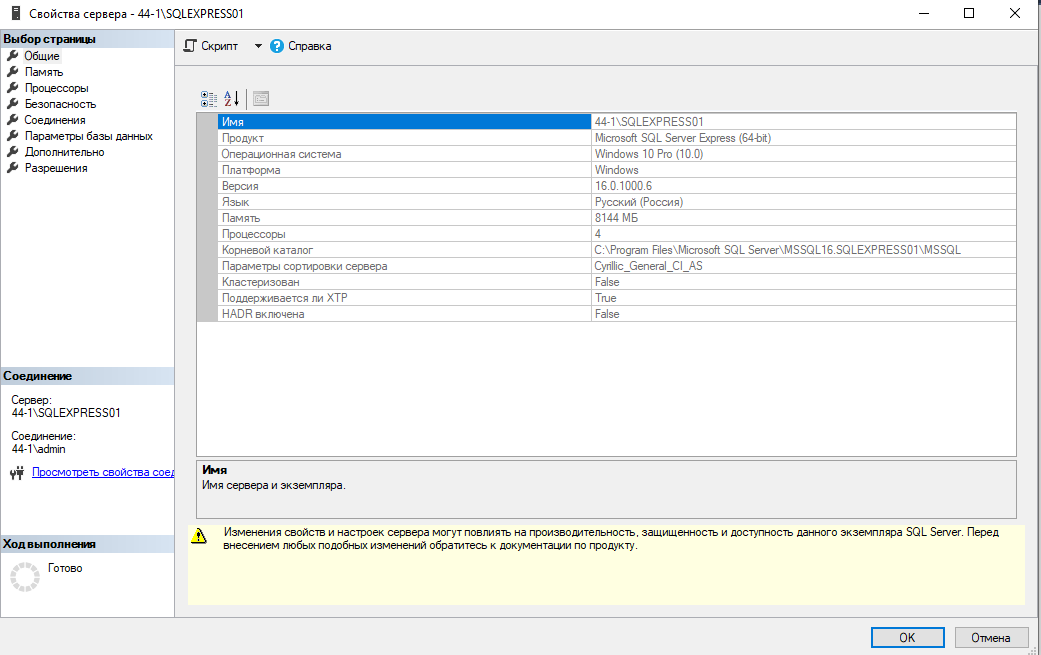


Рис.49 «Имя сервера»

1. После в строке задаём название БД и нажимаем Далее.

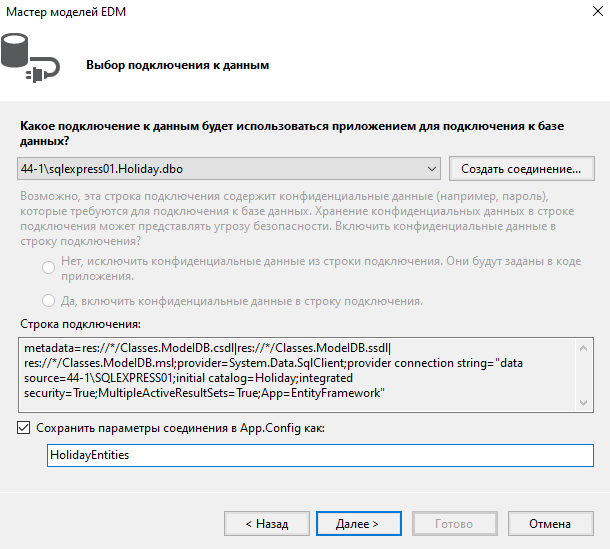


Рис.50 «Подключение БД»

1. Выбираем нужные таблицы и нажимаем Готово.

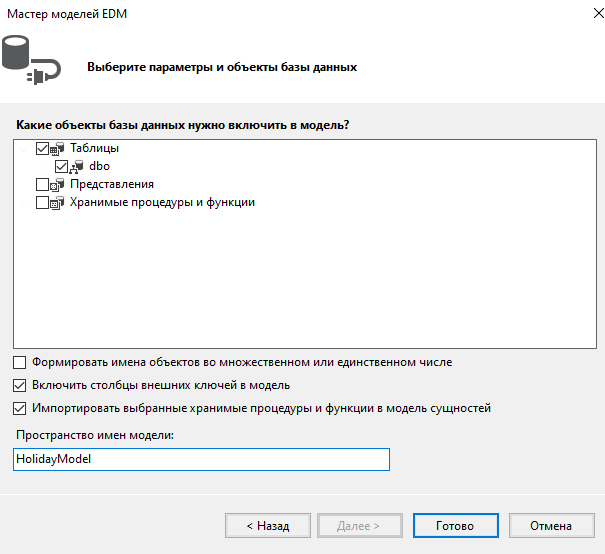


Рис.51 «Выборка таблиц для модели»

1. После откроется диаграмма модели БД и в папке появится модель БД.



Рис.52 «Модель в папке»

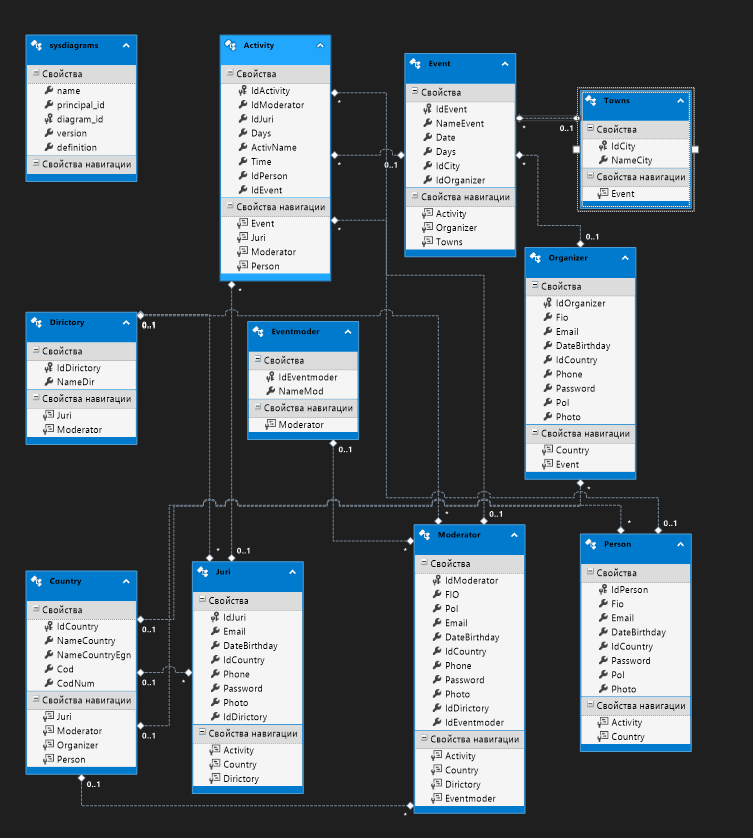


Рис.53 «Диаграмма модели БД»

* 1. **Перенос БД и её повторное подключение**

1. На новом ПК заходим в SQL Server Management Studio и ПКМ нажимаем на «Базы данных» и выбираем «Восстановить базу данных»

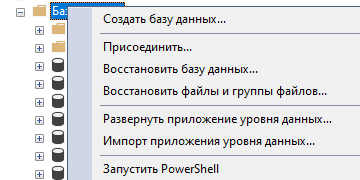


Рис.54 «Восстанавливаем БД»

1. Нажимаем на флажок «Устройство» и выбираем нашу резервную копию и нажимаем кнопку «Ок». БД восстановлена

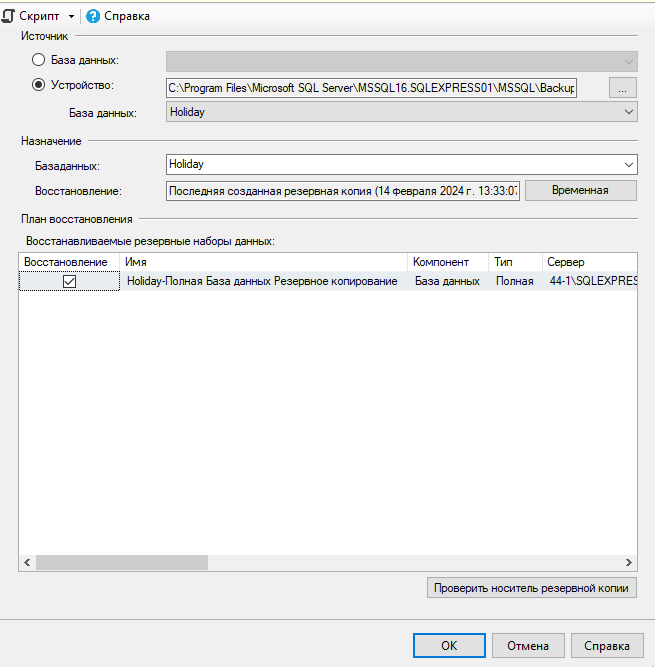


Рис.55 «Резервная копия БД»

1. **ПРОВЕРКА ПРОГРАММЫ И СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ**
   1. **Способы проверки**

Проверка работоспособности функции «Авторизация» проводится следующим образом (таблица №4).

Таблица №4 «Проверка авторизации»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Запустить программу «WpfHoliday» после этого нужно ввести логин и пароль на страницы авторизации. | Успешный запуск программы |
| 2 | **Проверка отображения информации пользователю ПО**  После успешной авторизации мы переходим страницу «Главное меню». | Отображение главного меню |

Проверка работоспособности функции «Отображения информации» проводится следующим образом (таблица №5).

Таблица №5 «Проверка отображения информации»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  С главного меню переходим на страницу «Участники». |  |
| 2 | **Проверка отображения информации пользователю ПО**  После перехода мы попадаем на страницу «Участники» с данными. | Отображение информации в виде таблицы с данными |

Проверка работоспособности функции «Добавление данных» проводится следующим образом (таблица №6).

Таблица №6 «Проверка добавления данных»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Нажимаем ПКМ по таблице и выбираем в контекстном меню функцию «Добавить». | Открытие окна с добавлением данных |
| 2 | **Проверка добавления информации**  После нажатия на кнопку и открытия окна «Добавление» вводим информацию в пустые строки ввода  Нажимаем кнопку «Добавить» | Успешный ввод данных в пустые поля, сохранение информации и её отображение в таблице с данными |

Проверка работоспособности функции «Удаление данных» проводится следующим образом (таблица №7).

Таблица №7 «Проверка удаления данных»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Выделяем запись или несколько записей в таблице данных и нажимаем на кнопку «Удалить» | Успешное выделение записей и вывод сообщения об удалении данных |
| 2 | **Проверка удалении информации**  После нажатия на кнопку и открытия всплывающего сообщения  Подтверждаем удаление данных, нажав кнопку ОК | Закрытие всплывающего сообщения и удаление данных из таблицы с данными |

Проверка работоспособности функции «Редактирование данных» проводится следующим образом (таблица №8).

Таблица №8 «Проверка редактирования данных»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Нажимаем ПКМ по таблице и выбираем в контекстном меню функцию «Добавить». | Открытие окна редактирование записей |
| 2 | **Проверка удалении информации**  После нажатия на кнопку и открытия окна редактирования записей  Изменяем информацию и нажимаем «Изменить» | Успешное изменение информации, сохранение и отображение информации в таблице с данными |

* 1. **Применяемые технические и организационные меры, используемые для обнаружения модификации ПО или любого расхождения между оригиналом и версией, полученной пользователем, в объеме, достаточном для правильной настройки и безопасного применения программы**

Организационные меры, используемые для обнаружения модификации ПО или любого расхождения между оригиналом и версией, полученной пользователем, должны в себя включать:

- средства маркировки дистрибутива (при его передаче пользователю на носителе);

- средства контрольного суммирования поставляемого дистрибутива программы;

- средства аутентификации (верификации) контрольных сумм при передаче дистрибутива и контрольных сумм по каналам связи;

- определение и утверждение конфигурация эталонных параметров ПО (оригинал);

- определение порядка установки, настройки и эксплуатации ПО в соответствии с эксплуатационной документацией на ПО ПМ ИЭТР, определены роли и права этих ролей на выполнение установки, настройки и эксплуатации ПО;

- определен порядок действий при обнаружении расхождений между оригиналом и версией, полученной пользователем, и их устранении;

- определен порядок аудита версий ПО, установленного у пользователей, на соответствие оригиналу.

Технические меры, используемые для обнаружения модификации ПО или любого расхождения между оригиналом и версией, полученной пользователем, включают:

- настройку прав разрешенных действий пользователей ПО;

- контроль и протоколирование действий пользователей.

На этапе разработки ПО указанные меры реализуются путем выполнения следующих шагов:

- исследование существующих у разработчика процессов в границах области действия мер по разработке безопасного ПО, связанных с идентификацией инструментальных средств разработки, отладки и тестирования;

- выбор способов обнаружения модификации ПО или любого расхождения между оригиналом и версией, полученной пользователем, для программ (и их частей), которые распространяются на физических носителях;

- выбор способов обнаружения модификации ПО или любого расхождения между оригиналом и версией, полученной пользователем, для программ (и их частей), которые распространяются по каналам связи;

- разработка процедуры обнаружения модификации файлов программы (или отдельных ее частей, например, обновлений);

- назначение работников, ответственных за реализацию меры по разработке безопасного ПО (в части контроля несанкционированной модификации кода), ознакомление их с документацией, касающейся реализации меры по разработке безопасного ПО.

При передаче дистрибутива и обновлений на носителе основными мерами контроля модификаций являются:

- маркировка дистрибутивного комплекта отличительными знаками (защитными знаками);

- использование контрольных сумм для идентифицированного перечня файлов.

При передаче дистрибутива и обновлений по каналам связи помимо контрольных сумм должны также использоваться средства двухключевой криптографии (хеш-суммы с подписью доверенным сертификатом), обеспечивающие возможность проверить доверенность источника при проверке контрольной суммы (защита от подделки контрольной суммы. Проверка осуществляется путем проверки сертификатов, выдаваемых доверенным удостоверяющим центром. Инфраструктура открытых ключей является элементом среды функционирования и предоставляется заказчиком (разработчик формирует подпись с помощью собственного сертификата, который должен быть включен заказчиком в число доверенных для обеспечения контроля дистрибуции).

Независимо от способа передачи дистрибутива и обновлений к мерам контроля несанкционированных модификаций относятся:

- организационно-технические меры обеспечения доверенной загрузки рабочей среды на компьютерах пользователей;

- использование штатных средств дистрибуции (установочного комплекта), содержащих проверку целостности и контрольных сумм модулей дистрибутива и обновлений.

**5. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ**

Дополнительные возможности – отсутствуют.

# Руководство пользователя

**1.Назначение программы:**

Программа предназначена для заключения договоров с физическими лицами для исполнения определённых услуг.

**2.Условия выполнения программы:**

Таблица№9 «Минимальные требования»

|  |  |
| --- | --- |
| Процессор | Intel(R) Core (TM) i5-4460 CPU @ 3.20GHz |
| Оперативная Память | 8 ГБ, DDR4, DIMM, 2133 МГц; |
| Разрешение экрана | 1920x1080 |
| Размер экрана | От 20 дюймов |
| Устройства ввода | Мышь, клавиатура |
| Дисковое пространство | 1 Гб |
| Подключаемые ус-ва | Принтер |
| Операционная система | Windows 10 и выше |

Требования к программным средствам:

Для корректной работы программы необходимо: ОС Windows 10 и выше, Microsoft Excel 2016, Microsoft Word 2019 и Microsoft SQL Management Studio.

**3.Выполнение программы:**

При открытии программы (исполняемый «.exe» файл) появится страница авторизации, нужно ввести логин и пароль от организатора и нажимаем кнопку «Войти», после этого происходит переход на страницу «Главное меню» с возможностью перехода на другие страницы: «Участники», «Жюри», «Организаторы», «Модераторы», «Активность».

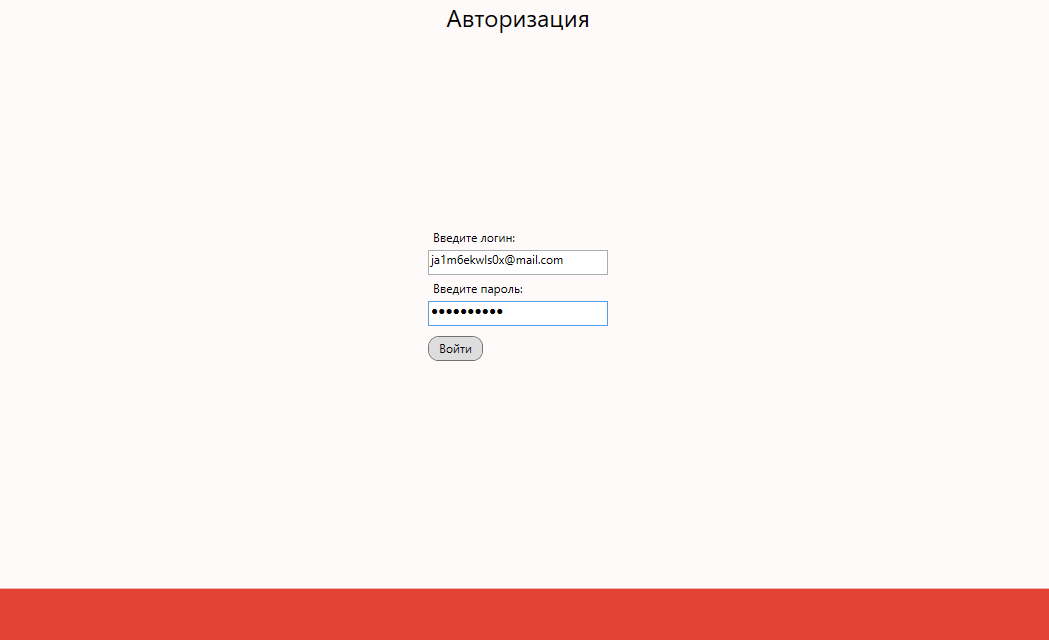


Рис.56 «Авторизация»

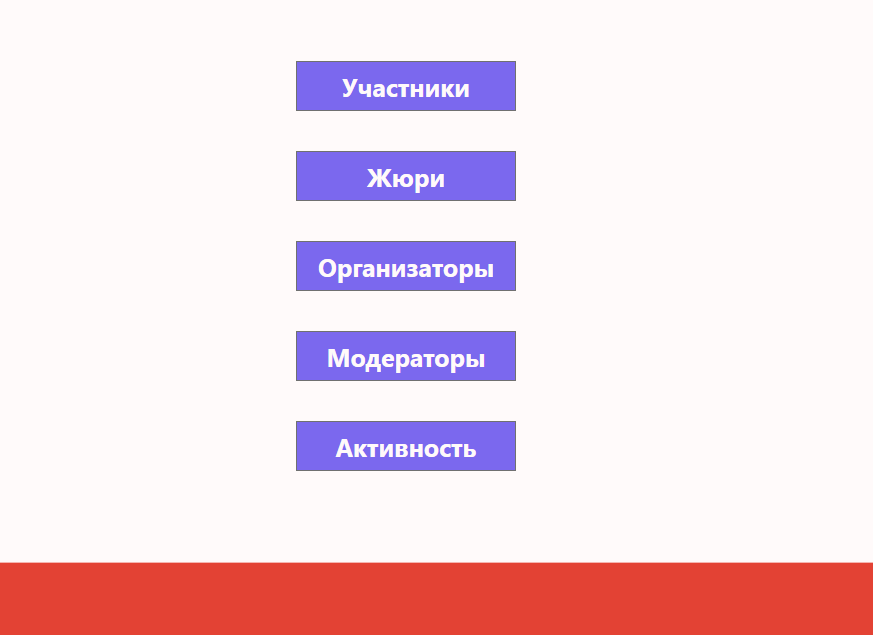


Рис.57 «Главное меню»

**Добавление:**

Для добавления необходимо нажать ПКМ по таблице и выбрать в контекстном меню функцию «Добавить», после этого переходим на страницу добавления и заполняем пустые поля.

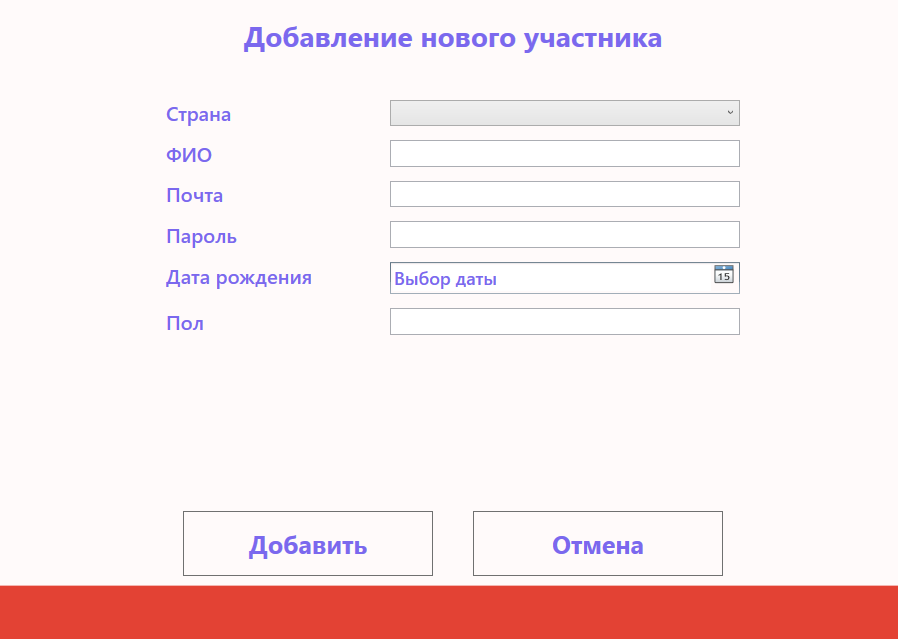


Рис.58 «Добавление записи»

**Удаление:**

Для удаления записей нужно нажать ПКМ по записи и выбрать в контекстном меню функцию «Удалить» и подтвердить удаление записи.



Рис.59 «Удаление»

**Просмотр списка Участники**



Рис.60 «Участники»

В данном списке приведены все участники, в котором можно посмотреть: ФИО, почту, пароль, страна, день рождения и пол.

**Сортировка по «От А до Я» и «От Я до А»**

Чтобы воспользоваться сортировкой нужно нажать по заголовку столбца.

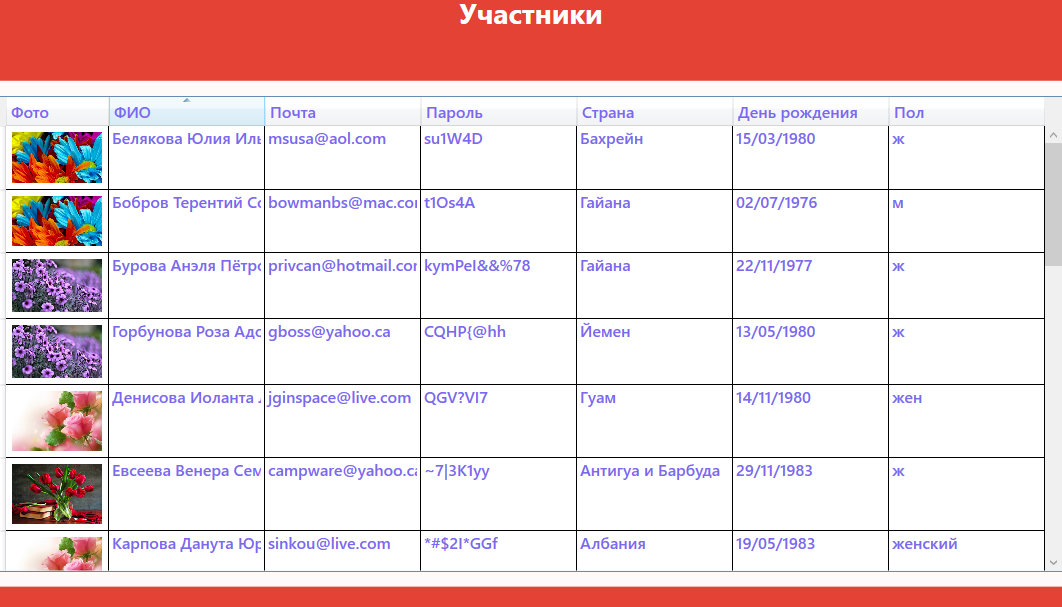


Рис.61 «Сортировка От А до Я»



Рис.62 «Сортировка От Я до А»

**Прочее**

**Экспорт отчетов в Excel:**

Для того чтобы отправить отчёт в Excel нужно: нужно нажать ПКМ по таблице и выбрать в контекстном меню функцию «Вывод данных в Excel». После нажатия будет открыт Excel в котором будут находиться записи таблицы.

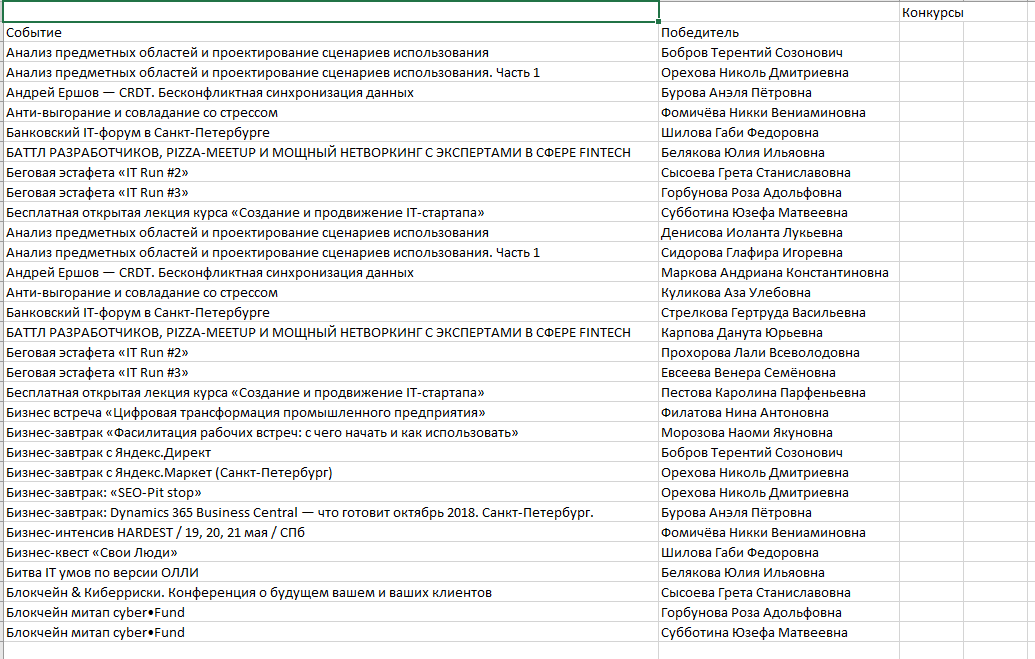


Рис.63 «Экспорт в Excel»

**Экспорт отчетов в Word:**

Если необходимо отправить отчёт в Word нужно: нужно нажать ПКМ по записи и выбрать в контекстном меню функцию «Вывод данных в Word». После нажатия будет открыт готовый шаблон вWord.

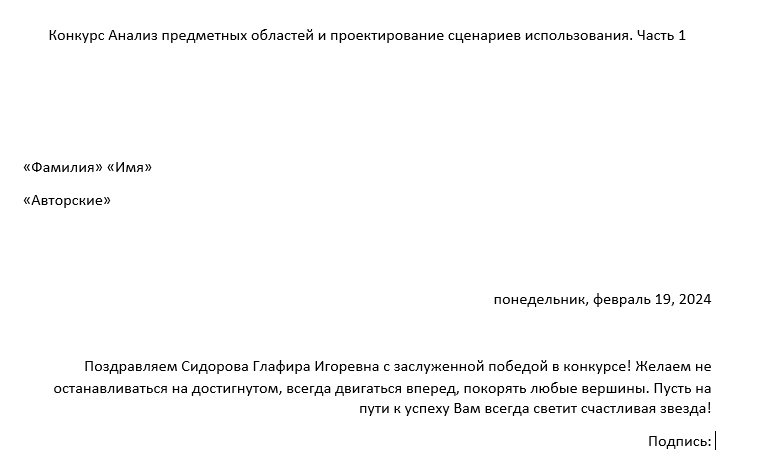


Рис.64 «Экспорт в Word»

**4.Сообщение оператору:**

Попытка добавления новой записи в таблицы с неуказанными данными

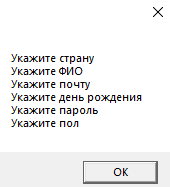


Рис.65 «Попытка добавления»

Подтверждение удаления

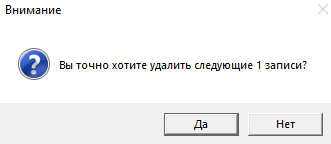


Рис.67 «Подтверждение удаления»

Информация после удаления записи

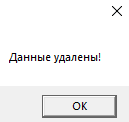


Рис.68 «Информация при удалении»

Информация после изменения записи

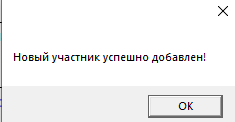


Рис.69 «Информация при изменении»

Информация после добавления записи

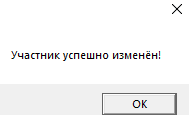


Рис.70 «Информация после добавления»

# Отладка и тестирование

**Приложения**

Тестовый сценарий №1

|  |  |
| --- | --- |
| **Тестовый пример #** | WpfHoliday |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | На странице «Активность» нажмите ПКМ по нужной записи и выберите функцию «Вывод данных в Word». |
| **Краткое изложение теста** | Должна выводиться документ Word |
| **Этапы теста** | 1. Запустите программу через Visual Studio.  2. Попадаем на страницу «Авторизацию».  3. После успешной авторизации переходим на страницу «Главное меню»  3. Затем переходим на страницу «Активность».  4. Нажмите ПКМ по нужной записи и выберите функцию «Вывод данных в Word». |
| **Тестовые данные** | Документ Word |
| **Ожидаемый результат** | Должен появиться документ Word с готовым шаблоном |
| **Фактический результат** | Выводит ошибку и пустой документ Word |
| **Статус** | Незачет. |
| **Предварительное условие** | Установите новую версию Word 2019. |
| **Постусловие** | Выдоится документ Word c готовым шаблоном |
| **Примечания/комментарии** | Пользователь должен обязательно выбрать нужную ему запись иначе будет шаблон с пустыми данными |

Тестовый сценарий №2

|  |  |
| --- | --- |
| **Тестовый пример #** | WpfHoliday |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | На странице «Авторизация» заполните логин и пароль от организатора. |
| **Краткое изложение теста** | Должен произойти переход на страницу «Главное меню» |
| **Этапы теста** | 1. Запустите программу через Visual Studio.  2. Попадаем на страницу «Авторизацию».  3. После успешной авторизации переходим на страницу «Главное меню» |
| **Тестовые данные** | Страницы Wpf, ввод логина и пароля |
| **Ожидаемый результат** | Должен произойти переход на страницу «Главное меню» |
| **Фактический результат** | Произошел переход на страницу «Главное меню» |
| **Статус** | Зачет. |
| **Предварительное условие** | Подготовьте логин и пароль от организатора |
| **Постусловие** | Выдоится страница «Главное меню» |
| **Примечания/комментарии** |  |

|  |  |  |
| --- | --- | --- |
| Ошибки | Скриншот ошибки | Отладка |
| **Ошибка подключения к базе данных.**  Отсутствует подключение к базе данных |  | Подключение к базе данных |
| **Орфографическая ошибка.**  Неверно написанный текст на элементе кнопка. |  | Верно написать текст на элементе кнопка. |
| **Синтаксическая ошибка.**  Отсутвует точка с запятой. |  | Требуется дописать точку с запятой. |
| **Отсутствие ссылок на дополнительные пакеты.**  Отсутствует ссылка наWpf\_Holiday.Classes; |  | Добавить ссылку дополнительного пакета Wpf\_Holiday.Classes; |
| **Отсутствие пути на расположение файла.**  Отсутствует ссылка на документ Microsoft.Office.Word |  | Добавить правильный путь до документа. |

# Текст программы

**ClassFrame.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace Wpf\_Holiday.Classes

{

internal class ClassFrame

{

public static Frame frmObj;

public static HolidayEntities db;

}

}

**Users.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Wpf\_Holiday.Classes

{

internal class Users

{

public static Users GetUsers { get; set; }

public int IdOrganizer { get; set; }

public string Fio { get; set; }

public string Email { get; set; }

public int IdCountry { get; set; }

public string Phone { get; set; }

public string Password { get; set; }

public string Pol { get; set; }

public Users(Organizer organizers)

{

IdOrganizer = organizers.IdOrganizer;

Fio = organizers.Fio;

Email = organizers.Email;

IdCountry = (int)organizers.IdCountry;

Phone = organizers.Phone;

Password = organizers.Password;

Pol = organizers.Pol;

GetUsers = this;

}

}

}

**Pages**

**Authorization.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для Authorization.xaml

/// </summary>

public partial class Authorization : Page

{

public Authorization()

{

InitializeComponent();

login.Text = "";

password.Password = "";

}

private void btnenter\_Click(object sender, RoutedEventArgs e)

{

if (login.Text == "") MessageBox.Show("Заполните поле логин", "Ошибка ввода данных", MessageBoxButton.OK, MessageBoxImage.Error);

if (password.Password == "") MessageBox.Show("Заполните поле пароль", "Ошибка ввода данных", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

Organizer organizers = ClassFrame.db.Organizer.FirstOrDefault(x => x.Email == login.Text && x.Password == password.Password);

if (organizers != null)

{

Users \_user = new Users(organizers);

ClassFrame.frmObj.Navigate(new PageMainMenu());

}

else

{

MessageBox.Show("Неверный логин или пароль", "Ошибка ввода данных", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

}

}

**PageActiv.xaml.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

using Word = Microsoft.Office.Interop.Word;

using Excel = Microsoft.Office.Interop.Excel;

using System.Runtime.Remoting.Metadata.W3cXsd2001;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageActiv.xaml

/// </summary>

public partial class PageActiv : Page

{

public PageActiv()

{

InitializeComponent();

dtgActiv.ItemsSource = HolidayEntities.GetContext().Activity.ToList();

}

private void MenuAdd\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddActiv(null));

}

private void MenuEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddActiv((Activity)dtgActiv.SelectedItem));

}

private void MenuUpdate\_Click(object sender, RoutedEventArgs e)

{

dtgActiv.ItemsSource = HolidayEntities.GetContext().Activity.ToList();

}

private void MenuDel\_Click(object sender, RoutedEventArgs e)

{

var salesForRemoving = dtgActiv.SelectedItems.Cast<Activity>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {salesForRemoving.Count()} записи?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

HolidayEntities.GetContext().Activity.RemoveRange(salesForRemoving);

HolidayEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

dtgActiv.ItemsSource = HolidayEntities.GetContext().Activity.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void MenuGlav\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMainMenu());

}

private void MenuWord\_Click(object sender, RoutedEventArgs e)

{

Activity salesForRemoving = (Activity)dtgActiv.SelectedItem;

Word.Selection wrdSelection;

Word.MailMerge wrdMailMerge;

Word.MailMergeFields wrdMergeFields;

string StrToAdd;

// Create an instance of Word and make it visible.

wrdApp = new Word.Application();

wrdApp.Visible = true;

// Add a new document.

wrdDoc = wrdApp.Documents.Add(ref oMissing, ref oMissing,

ref oMissing, ref oMissing);

wrdDoc.Select();

wrdSelection = wrdApp.Selection;

wrdMailMerge = wrdDoc.MailMerge;

// Create a MailMerge Data file.

//CreateMailMergeDataFile();

// Create a string and insert it into the document.

StrToAdd = $"Конкурс {salesForRemoving.Event.NameEvent} ";

wrdSelection.ParagraphFormat.Alignment =

Word.WdParagraphAlignment.wdAlignParagraphCenter;

wrdSelection.TypeText(StrToAdd);

InsertLines(4);

// Insert merge data.

wrdSelection.ParagraphFormat.Alignment =

Word.WdParagraphAlignment.wdAlignParagraphLeft;

wrdMergeFields = wrdMailMerge.Fields;

wrdMergeFields.Add(wrdSelection.Range, "Фамилия");

wrdSelection.TypeText(" ");

wrdMergeFields.Add(wrdSelection.Range, "Имя");

wrdSelection.TypeParagraph();

wrdMergeFields.Add(wrdSelection.Range, $"{salesForRemoving.ActivName}");

wrdSelection.TypeParagraph();

InsertLines(2);

// Right justify the line and insert a date field

// with the current date.

wrdSelection.ParagraphFormat.Alignment =

Word.WdParagraphAlignment.wdAlignParagraphRight;

Object objDate = "dddd, MMMM dd, yyyy";

wrdSelection.InsertDateTime(ref objDate, ref oFalse, ref oMissing,

ref oMissing, ref oMissing);

InsertLines(2);

// Create a string and insert it into the document.

StrToAdd = $"Поздравляем {salesForRemoving.Person.Fio} с заслуженной победой в конкурсе! " +

$"Желаем не останавливаться на достигнутом, всегда двигаться вперед, покорять любые вершины. " +

$"Пусть на пути к успеху Вам всегда светит счастливая звезда!";

wrdSelection.TypeText(StrToAdd);

wrdSelection.TypeParagraph();

wrdSelection.TypeText("Подпись: ");

}

Word.Application wrdApp;

Word.\_Document wrdDoc;

Object oMissing = System.Reflection.Missing.Value;

Object oFalse = false;

private void InsertLines(int LineNum)

{

int iCount;

// Insert "LineNum" blank lines.

for (iCount = 1; iCount <= LineNum; iCount++)

{

wrdApp.Selection.TypeParagraph();

}

}

private void FillRow(Word.\_Document oDoc, int Row, string Text1, string Text2, string Text3, string Text4)

{

// Insert the data into the specific cell.

oDoc.Tables[1].Cell(Row, 1).Range.InsertAfter(Text1);

oDoc.Tables[1].Cell(Row, 2).Range.InsertAfter(Text2);

oDoc.Tables[1].Cell(Row, 3).Range.InsertAfter(Text3);

oDoc.Tables[1].Cell(Row, 4).Range.InsertAfter(Text4);

}

private void MenuExcel\_Click(object sender, RoutedEventArgs e)

{

Excel.Application excelApp = new Excel.Application();

Excel.Workbook wb = excelApp.Workbooks.Open($"{Directory.GetCurrentDirectory()}\\Shablon.xlsx");

Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets[1];

ws.Cells[1, 3] = "Конкурсы";

int indexRows = 2;

//ячейка

ws.Cells[1][indexRows] = "Событие";

ws.Cells[2][indexRows] = "Победитель";

//список пользователей из таблицы после фильтрации и поиска

var printItems = dtgActiv.Items;

//цикл по данным из списка для печати

foreach (Activity item in printItems)

{

ws.Cells[1][indexRows + 1] = item.Event.NameEvent;

ws.Cells[2][indexRows + 1] = item.Person.Fio;

indexRows++;

}

excelApp.Visible = true;

}

}

}

**PageJuri.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageJuri.xaml

/// </summary>

public partial class PageJuri : Page

{

public PageJuri()

{

InitializeComponent();

dtgJuri.ItemsSource = HolidayEntities.GetContext().Juri.ToList();

}

private void MenuAdd\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddJuri(null));

}

private void MenuEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddJuri((Juri)dtgJuri.SelectedItem));

}

private void MenuUpdate\_Click(object sender, RoutedEventArgs e)

{

dtgJuri.ItemsSource = HolidayEntities.GetContext().Juri.ToList();

}

private void MenuDel\_Click(object sender, RoutedEventArgs e)

{

var salesForRemoving = dtgJuri.SelectedItems.Cast<Juri>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {salesForRemoving.Count()} записи?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

HolidayEntities.GetContext().Juri.RemoveRange(salesForRemoving);

HolidayEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

dtgJuri.ItemsSource = HolidayEntities.GetContext().Juri.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void MenuGlav\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMainMenu());

}

}

}

**PageModer.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageModer.xaml

/// </summary>

public partial class PageModer : Page

{

public PageModer()

{

InitializeComponent();

dtgModerator.ItemsSource = HolidayEntities.GetContext().Moderator.ToList();

}

private void MenuAdd\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddModer(null));

}

private void MenuEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddModer((Moderator)dtgModerator.SelectedItem));

}

private void MenuUpdate\_Click(object sender, RoutedEventArgs e)

{

dtgModerator.ItemsSource = HolidayEntities.GetContext().Moderator.ToList();

}

private void MenuDel\_Click(object sender, RoutedEventArgs e)

{

var salesForRemoving = dtgModerator.SelectedItems.Cast<Moderator>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {salesForRemoving.Count()} записи?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

HolidayEntities.GetContext().Moderator.RemoveRange(salesForRemoving);

HolidayEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

dtgModerator.ItemsSource = HolidayEntities.GetContext().Moderator.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void MenuGlav\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMainMenu());

}

}

}

**PageOrganiz.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageOrganiz.xaml

/// </summary>

public partial class PageOrganiz : Page

{

public PageOrganiz()

{

InitializeComponent();

dtgOrganiz.ItemsSource = HolidayEntities.GetContext().Organizer.ToList();

}

private void MenuAdd\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddOrgniz(null));

}

private void MenuEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddOrgniz((Organizer)dtgOrganiz.SelectedItem));

}

private void MenuUpdate\_Click(object sender, RoutedEventArgs e)

{

dtgOrganiz.ItemsSource = HolidayEntities.GetContext().Person.ToList();

}

private void MenuDel\_Click(object sender, RoutedEventArgs e)

{

var salesForRemoving = dtgOrganiz.SelectedItems.Cast<Organizer>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {salesForRemoving.Count()} записи?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

HolidayEntities.GetContext().Organizer.RemoveRange(salesForRemoving);

HolidayEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

dtgOrganiz.ItemsSource = HolidayEntities.GetContext().Organizer.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void MenuGlav\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMainMenu());

}

}

}

**PagePerson.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Remoting.Metadata.W3cXsd2001;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PagePerson.xaml

/// </summary>

public partial class PagePerson : Page

{

public PagePerson()

{

InitializeComponent();

dtgPerson.ItemsSource = HolidayEntities.GetContext().Person.ToList();

}

private void MenuAdd\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddPerson(null));

}

private void MenuEdit\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageAddPerson((Person)dtgPerson.SelectedItem));

}

private void MenuUpdate\_Click(object sender, RoutedEventArgs e)

{

dtgPerson.ItemsSource = HolidayEntities.GetContext().Person.ToList();

}

private void MenuDel\_Click(object sender, RoutedEventArgs e)

{

var salesForRemoving = dtgPerson.SelectedItems.Cast<Person>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {salesForRemoving.Count()} записи?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

HolidayEntities.GetContext().Person.RemoveRange(salesForRemoving);

HolidayEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены!");

dtgPerson.ItemsSource = HolidayEntities.GetContext().Person.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void MenuGlav\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMainMenu());

}

}

}

**PageMainMenu.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageMainMenu.xaml

/// </summary>

public partial class PageMainMenu : Page

{

public PageMainMenu()

{

InitializeComponent();

}

private void btnToPerson\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PagePerson());

}

private void btnToJuri\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageJuri());

}

private void btnToOrginiz\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageOrganiz());

}

private void btnToModer\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageModer());

}

private void btnToActiv\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageActiv());

}

}

}

**PageAddActiv.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageAddActiv.xaml

/// </summary>

public partial class PageAddActiv : Page

{

private Activity \_currentItem = new Activity();

public PageAddActiv(Activity selectedItem)

{

InitializeComponent();

if (selectedItem != null)

{

\_currentItem = selectedItem;

Titletxt.Text = "Изменение событьия";

BtnAdd.Content = "Изменить";

}

DataContext = \_currentItem;

CMBEvent.ItemsSource = HolidayEntities.GetContext().Event.ToList();

CMBEvent.SelectedValuePath = "IdEvent";

CMBEvent.DisplayMemberPath = "NameEvent";

CMBJuri.ItemsSource = HolidayEntities.GetContext().Juri.ToList();

CMBJuri.SelectedValuePath = "IdJuri";

CMBJuri.DisplayMemberPath = "Email";

CMBModer.ItemsSource = HolidayEntities.GetContext().Moderator.ToList();

CMBModer.SelectedValuePath = "IdModerator";

CMBModer.DisplayMemberPath = "FIO";

CMBPerson.ItemsSource = HolidayEntities.GetContext().Person.ToList();

CMBPerson.SelectedValuePath = "IdPerson";

CMBPerson.DisplayMemberPath = "Fio";

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdModerator))) error.AppendLine("Укажите модератора");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdJuri))) error.AppendLine("Укажите жюри");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Days))) error.AppendLine("Укажите дни");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.ActivName))) error.AppendLine("Укажите доп. событие");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Time))) error.AppendLine("Укажите время");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdPerson))) error.AppendLine("Укажите победителя");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdEvent))) error.AppendLine("Укажите событьие");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentItem.IdActivity == 0)

{

HolidayEntities.GetContext().Activity.Add(\_currentItem);

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PageActiv());

MessageBox.Show("Новое событие успешно добавлен!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

else

{

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PageActiv());

MessageBox.Show("Событие успешно изменён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageActiv());

}

}

}

**PageAddJuri.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageAddJuri.xaml

/// </summary>

public partial class PageAddJuri : Page

{

private Juri \_currentItem = new Juri();

public PageAddJuri(Juri selectedItem)

{

InitializeComponent();

if (selectedItem != null)

{

\_currentItem = selectedItem;

Titletxt.Text = "Изменение жюри";

BtnAdd.Content = "Изменить";

}

DataContext = \_currentItem;

CMBCountry.ItemsSource = HolidayEntities.GetContext().Country.ToList();

CMBCountry.SelectedValuePath = "IdCountry";

CMBCountry.DisplayMemberPath = "NameCountry";

CMBDiric.ItemsSource = HolidayEntities.GetContext().Dirictory.ToList();

CMBDiric.SelectedValuePath = "IdDirictory";

CMBDiric.DisplayMemberPath = "NameDir";

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdCountry))) error.AppendLine("Укажите страну");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdDirictory))) error.AppendLine("Укажите направление");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Email))) error.AppendLine("Укажите почту");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.DateBirthday))) error.AppendLine("Укажите день рождения");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Password))) error.AppendLine("Укажите пароль");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Phone))) error.AppendLine("Укажите телефон");

if (HolidayEntities.GetContext().Juri.Where(x => x.Country.NameCountry.ToString() == CMBDiric.Text && x.Dirictory.NameDir == CMBDiric.Text && x.Email == TxtEmail.Text && x.DateBirthday.ToString() == datePicker.Text && x.Password == TxtPassword.Text && x.Phone.ToString() == Txtphone.Text).Count() > 0)

{

MessageBox.Show("Такой жюри уже есть");

return;

}

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentItem.IdJuri == 0)

{

HolidayEntities.GetContext().Juri.Add(\_currentItem);

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PageJuri());

MessageBox.Show("Новый жюри успешно добавлен!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

else

{

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PageJuri());

MessageBox.Show("Жюри успешно изменён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageJuri());

}

}

}

**PageAddModer.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageAddModer.xaml

/// </summary>

public partial class PageAddModer : Page

{

private Moderator \_currentItem = new Moderator();

public PageAddModer(Moderator selectedItem)

{

InitializeComponent();

if (selectedItem != null)

{

\_currentItem = selectedItem;

Titletxt.Text = "Изменение модератора";

BtnAdd.Content = "Изменить";

}

DataContext = \_currentItem;

CMBCountry.ItemsSource = HolidayEntities.GetContext().Country.ToList();

CMBCountry.SelectedValuePath = "IdCountry";

CMBCountry.DisplayMemberPath = "NameCountry";

CMBDirictory.ItemsSource = HolidayEntities.GetContext().Dirictory.ToList();

CMBDirictory.SelectedValuePath = "IdDirictory";

CMBDirictory.DisplayMemberPath = "NameDir";

CMBEventModer.ItemsSource = HolidayEntities.GetContext().Eventmoder.ToList();

CMBEventModer.SelectedValuePath = "IdEventmoder";

CMBEventModer.DisplayMemberPath = "NameMod";

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdCountry))) error.AppendLine("Укажите страну");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.FIO))) error.AppendLine("Укажите ФИО");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Email))) error.AppendLine("Укажите почту");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.DateBirthday))) error.AppendLine("Укажите день рождения");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Password))) error.AppendLine("Укажите пароль");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Phone))) error.AppendLine("Укажите телефон");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Pol))) error.AppendLine("Укажите пол");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdDirictory))) error.AppendLine("Укажите направление");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdEventmoder))) error.AppendLine("Укажите событие модератора");

if (HolidayEntities.GetContext().Organizer.Where(x => x.Country.NameCountry.ToString() == CMBCountry.Text && x.Fio == Txtfio.Text && x.Email == TxtEmail.Text && x.DateBirthday.ToString() == datePicker.Text && x.Password == TxtPassword.Text && x.Pol.ToString() == TxtPol.Text).Count() > 0)

{

MessageBox.Show("Такой модератор уже есть");

return;

}

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentItem.IdModerator == 0)

{

HolidayEntities.GetContext().Moderator.Add(\_currentItem);

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PageModer());

MessageBox.Show("Новый модератор успешно добавлен!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

else

{

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PageModer());

MessageBox.Show("Модератор успешно изменён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageModer());

}

}

}

**PageAddOrgniz.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageAddOrgniz.xaml

/// </summary>

public partial class PageAddOrgniz : Page

{

private Organizer \_currentItem = new Organizer();

public PageAddOrgniz(Organizer selectedItem)

{

InitializeComponent();

if (selectedItem != null)

{

\_currentItem = selectedItem;

Titletxt.Text = "Изменение организатора";

BtnAdd.Content = "Изменить";

}

DataContext = \_currentItem;

CMBCountry.ItemsSource = HolidayEntities.GetContext().Country.ToList();

CMBCountry.SelectedValuePath = "IdCountry";

CMBCountry.DisplayMemberPath = "NameCountry";

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdCountry))) error.AppendLine("Укажите страну");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Fio))) error.AppendLine("Укажите ФИО");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Email))) error.AppendLine("Укажите почту");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.DateBirthday))) error.AppendLine("Укажите день рождения");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Password))) error.AppendLine("Укажите пароль");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Phone))) error.AppendLine("Укажите телефон");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Pol))) error.AppendLine("Укажите пол");

if (HolidayEntities.GetContext().Organizer.Where(x => x.Country.NameCountry.ToString() == CMBCountry.Text && x.Fio == Txtfio.Text && x.Email == TxtEmail.Text && x.DateBirthday.ToString() == datePicker.Text && x.Password == TxtPassword.Text && x.Pol.ToString() == TxtPol.Text).Count() > 0)

{

MessageBox.Show("Такой организатор уже есть");

return;

}

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentItem.IdOrganizer == 0)

{

HolidayEntities.GetContext().Organizer.Add(\_currentItem);

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PageOrganiz());

MessageBox.Show("Новый оргназитор успешно добавлен!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

else

{

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PageOrganiz());

MessageBox.Show("Организатор успешно изменён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageOrganiz());

}

}

}

**PageAddPerson.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.Remoting.Metadata.W3cXsd2001;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Wpf\_Holiday.Classes;

namespace Wpf\_Holiday.Pages

{

/// <summary>

/// Логика взаимодействия для PageAddPerson.xaml

/// </summary>

public partial class PageAddPerson : Page

{

private Person \_currentItem = new Person();

public PageAddPerson(Person selectedItem)

{

InitializeComponent();

if (selectedItem != null)

{

\_currentItem = selectedItem;

Titletxt.Text = "Изменение участника";

BtnAdd.Content = "Изменить";

}

DataContext = \_currentItem;

CMBCountry.ItemsSource = HolidayEntities.GetContext().Country.ToList();

CMBCountry.SelectedValuePath = "IdCountry";

CMBCountry.DisplayMemberPath = "NameCountry";

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.IdCountry))) error.AppendLine("Укажите страну");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Fio))) error.AppendLine("Укажите ФИО");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Email))) error.AppendLine("Укажите почту");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.DateBirthday))) error.AppendLine("Укажите день рождения");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Password))) error.AppendLine("Укажите пароль");

if (string.IsNullOrWhiteSpace(Convert.ToString(\_currentItem.Pol))) error.AppendLine("Укажите пол");

if (HolidayEntities.GetContext().Person.Where(x => x.Country.NameCountry.ToString() == CMBCountry.Text && x.Fio == Txtfio.Text && x.Email == TxtEmail.Text && x.DateBirthday.ToString() == datePicker.Text && x.Password == TxtPassword.Text && x.Pol.ToString() == TxtPol.Text).Count() > 0)

{

MessageBox.Show("Такой участник уже есть");

return;

}

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentItem.IdPerson == 0)

{

HolidayEntities.GetContext().Person.Add(\_currentItem);

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PagePerson());

MessageBox.Show("Новый участник успешно добавлен!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

else

{

try

{

HolidayEntities.GetContext().SaveChanges();

ClassFrame.frmObj.Navigate(new PagePerson());

MessageBox.Show("Участник успешно изменён!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void BtnCancel\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PagePerson());

}

}

}

# Заключение

Учебную практику по ПМ.02 «Осуществление интеграции программных модулей» проходил в учебном учреждении Ликино-Дулевский политехнический колледж – филиал ГГТУ.

Во время учебной практики выполнялись следующие виды работ:

1. Разработка технического задания и спецификации;
2. Проектирование предметной области;
3. Разработке и интеграция программных модулей;
4. Проведение тестирования и отладки программных модулей;
5. Разработка технической документации.

В соответствии с постановкой задачи был разработано приложение со следующей структурой данных:

* Участник
* Модератор
* Организатор
* Жюри

Спроектирована и разработана база данных, хранение данных организованно в локальной базе данных.

Для разработки приложения использовались следующие программные средства: ОС Windows 10 и выше, Microsoft Excel 2016, Microsoft Word 2016 (и выше) и Microsoft SQL Management Studio.

В соответствии с постановкой задачи разработан алгоритм работы приложения. Спроектированы экранные формы приложения. На этапе проектирования разработаны ER-диаграммы.

Произведена отладка, тестирование и оптимизация приложения с использованием встроенных средств отладки приложений в среде программирования.

На основании созданного приложения и в соответствие с ГОСТами разработана техническая документация: техническое задание, руководство пользователя, алгоритм программы, руководство системного программиста.

# Список литературы

1. Овечкин Г.В. Компьютерное моделирование: учебник для учреждений СПО /Г.В. Овечкин, П.В. Овечкин - М.: Издательский центр «Академия»,2020.- 205 с. -(Профессиональное образование)

2. Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2023. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/514591

3. Древс, Ю. Г. Имитационное моделирование : учебное пособие для среднего профессионального образования / Ю. Г. Древс, В. В. Золотарёв. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 142 с. — (Профессиональное образование). — ISBN 978-5-534-11951-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/517790

4. Огнева, М. В. Программирование на языке С++: практический курс : учебное пособие для среднего профессионального образования / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2023. — 335 с. — (Профессиональное образование). — ISBN 978-5-534-05780-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/515206

5.Подбельский, В. В. Программирование. Базовый курс С# : учебник для среднего профессионального образования / В. В. Подбельский. — Москва : Издательство Юрайт, 2023. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/517893

6. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — Москва : Издательство Юрайт, 2023. — 219 с. — (Профессиональное образование). — ISBN 978-5-9916-9984-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/513113

7.Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для среднего профессионального образования / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 147 с. — (Профессиональное образование). — ISBN 978-5-534-09823-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/515393

Интернет-ресурсы

1. ЭБС «Университетская библиотека online» - http://biblioclub.ru/

2. ЭБС «ЮРАЙТ» - https://urait.ru/