**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\_\_\_\_\_\_ Иванова Ивана Ивановича \_\_\_\_\_\_\_\_\_\_\_\_\_

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](#_Toc159485516)

[2. Разработка пользовательского интерфейса 6](#_Toc159485517)

[3. Алгоритм работы приложения(блоксхема) 30](#_Toc159485518)

[4. Проектирование ER-диаграммы 31](#_Toc159485519)

[5. Проектирование и разработка базы данных 32](#_Toc159485520)

[6. Руководство системного программиста 38](#_Toc159485521)

[7. Руководство пользователя 48](#_Toc159485522)

[8. Отладка и тестирование 53](#_Toc159485523)

[Заключение 84](#_Toc159485524)

[Список литературы 85](#_Toc159485525)

1. **Разработка спецификации**
   1. **Наименование модуля**

Конференции по информационной безопасности

* 1. **Цель модуля**

Система предназначена для организаторов конференций по информационной безопасности.

* 1. **Функциональные возможности модуля**

Функциональные возможности модуля:

* Авторизация для:
* Участников
* Организаторов
* Жури
* Модераторов
* Добавление запесей в таблицах:
* Участников
* Организаторов
* Жури
* Модераторов
* Активности
* Удаление запесей в таблицах:
* Участников
* Организаторов
* Жури
* Модераторов
* Активности
* Редактирование запесей в таблицах:
* Участников
* Организаторов
* Жури
* Модераторов
* Активности
* Поиск записей по:
* ФИО
* Название страны
* Фильтрация записей по:
* Полу

Импорт данных Excel в БД Microsoft SQL Management Studio

* 1. **Взаимодействие с другими модулями**

Программный модуль взаимодействует с Excel (импорт данных в БД) и Word (экспорт данных в документы)

* 1. **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.

* 1. **Входные данные**

Таблица №1 «Словарь данных»

|  |  |  |  |
| --- | --- | --- | --- |
| **ключ** | **поле** | **обязательное** | **примечание** |
| **Участники** | | | |
| Первичный | IDParticipants | Да | Идентификатор |
|  | FIO | Да | ФИО участника |
|  | Email | Да | Почта участника |
|  | DateOfBirth | Да | Дата рождения |
| Внешний | IDCountry | Да | Внешний ключ к таблице Стран |
|  | Phone | Да | Телефон участника |
|  | Pasword | Да | Пароль |
|  | Photo | Да | Фото участника |
| Внешний | IDPol | Да | Внешний ключ к таблице Пол |
| **Организаторы** | | | |
| Первичный | IDOrganizers | Да | Идентификатор |
|  | FIO | Да | ФИО организатора |
|  | Email | Да | Почта организатора |
|  | DateOfBirth | Да | Дата рождения |
| Внешний | IDCountry | Да | Внешний ключ к таблице Стран |
|  | Phone | Да | Телефон организатора |
|  | Pasword | Да | Пароль |
|  | Photo | Да | Фото организатора |
| Внешний | IDPol | Да | Внешний ключ к таблице Пол |
| **Жури** | | | |
| Первичный | IDOrganizers | Да | Идентификатор |
|  | FIO | Да | ФИО жури |
|  | Email | Да | Почта жури |
|  | DateOfBirth | Да | Дата рождения |
| Внешний | IDCountry | Да | Внешний ключ к таблице Стран |
|  | Phone | Да | Телефон жури |
|  | Pasword | Да | Пароль |
|  | Photo | Да | Фото жури |
| Внешний | IDPol | Да | Внешний ключ к таблице Пол |
| Внешний | IDDirection | Да | Внешний ключ к таблице Направление |
| **Модераторы** | | | |
| Первичный | IDOrganizers | Да | Идентификатор |
|  | FIO | Да | ФИО жури |
|  | Email | Да | Почта жури |
|  | DateOfBirth | Да | Дата рождения |
| Внешний | IDCountry | Да | Внешний ключ к таблице Стран |
|  | Phone | Да | Телефон жури |
|  | Pasword | Да | Пароль |
|  | Photo | Да | Фото жури |
| Внешний | IDPol | Да | Внешний ключ к таблице Пол |
| Внешний | IDDirection | Да | Внешний ключ к таблице Направление |
|  | NameEvent | Да | Название события |
| **Событие** | | | |
| Первичный | IDEvent | Да | Идентификатор |
|  | NameEvent | Да | Название события |
|  | Datee | Да | Дата проведения события |
|  | Days | Да | Дни |
| Внешний | IDCity | Да | Внешний ключ к таблице Города |
| **Активности** | | | |
| Первичный | IDActivites | Да | Идентификатор |
| Внешний | IDEvent | Да | Внешний ключ к таблице События |
|  | TimeStart | Да | Дата начала активности |
| Внешний | IDOrganizers | Да | Внешний ключ к таблице Организаторы |
| Внешний | IDModerators | Да | Внешний ключ к таблице Модераторы |
| Внешний | IDJury | Да | Внешний ключ к таблице Жури |
| Внешний | IDParticipants | Да | Внешний ключ к таблице Участника |

* 1. **Выходные данные**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ключ** | **поле** | **обязательное** | **примечание** |
| **Участники** | | | |
| Первичный | IDParticipants | Да | Идентификатор |
|  | FIO | Да | ФИО участника |
|  | Email | Да | Почта участника |
|  | DateOfBirth | Да | Дата рождения |
| Внешний | IDCountry | Да | Внешний ключ к таблице Стран |
|  | Phone | Да | Телефон участника |
|  | Pasword | Да | Пароль |
|  | Photo | Да | Фото участника |
| Внешний | IDPol | Да | Внешний ключ к таблице Пол |
| **Организаторы** | | | |
| Первичный | IDOrganizers | Да | Идентификатор |
|  | FIO | Да | ФИО организатора |
|  | Email | Да | Почта организатора |
|  | DateOfBirth | Да | Дата рождения |
| Внешний | IDCountry | Да | Внешний ключ к таблице Стран |
|  | Phone | Да | Телефон организатора |
|  | Pasword | Да | Пароль |
|  | Photo | Да | Фото организатора |
| Внешний | IDPol | Да | Внешний ключ к таблице Пол |
| **Жюри** | | | |
| Первичный | IDOrganizers | Да | Идентификатор |
|  | FIO | Да | ФИО жури |
|  | Email | Да | Почта жури |
|  | DateOfBirth | Да | Дата рождения |
| Внешний | IDCountry | Да | Внешний ключ к таблице Стран |
|  | Phone | Да | Телефон жури |
|  | Pasword | Да | Пароль |
|  | Photo | Да | Фото жури |
| Внешний | IDPol | Да | Внешний ключ к таблице Пол |
| Внешний | IDDirection | Да | Внешний ключ к таблице Направление |
| **Модераторы** | | | |
| Первичный | IDOrganizers | Да | Идентификатор |
|  | FIO | Да | ФИО жури |
|  | Email | Да | Почта жури |
|  | DateOfBirth | Да | Дата рождения |
| Внешний | IDCountry | Да | Внешний ключ к таблице Стран |
|  | Phone | Да | Телефон жури |
|  | Pasword | Да | Пароль |
|  | Photo | Да | Фото жури |
| Внешний | IDPol | Да | Внешний ключ к таблице Пол |
| Внешний | IDDirection | Да | Внешний ключ к таблице Направление |
|  | NameEvent | Да | Название события |
| **Активности** | | | |
| Первичный | IDActivites | Да | Идентификатор |
| Внешний | IDEvent | Да | Внешний ключ к таблице События |
|  | TimeStart | Да | Дата начала активности |
| Внешний | IDOrganizers | Да | Внешний ключ к таблице Организаторы |
| Внешний | IDModerators | Да | Внешний ключ к таблице Модераторы |
| Внешний | IDJury | Да | Внешний ключ к таблице Жури |
| Внешний | IDParticipants | Да | Внешний ключ к таблице Участника |

* 1. **Навигация внутри модуля**

При запуске приложения пользователь заходит в окно авторизации где дальше он может перейти на страницу Меню после внесения данных или на страницу Регистрации. На странице регистрации пользователь вводит свои данные после чего переход на страницу Меню. На странице Меню пользователь может перейти на страницы: Акаунт, События, Активности (если пользователем является Модератор или Организатор, то в Меню добавляется страница Просмотра пользователей). Перейдя на страницу Акаунт пользователей может просмотреть свои данные, изменить пароль и выйти из акаунта. На станице События пользователь может посмотреть информацию о событии и возможность вернуться на страницу меню. На странице Активности можно узнать информацию о результатах события и возможность вернуться на страницу меню. На странице Просмотра пользователя можно узнать информацию о всех пользователях и также можно вернуться на страницу меню.

* 1. **Требования к макету**

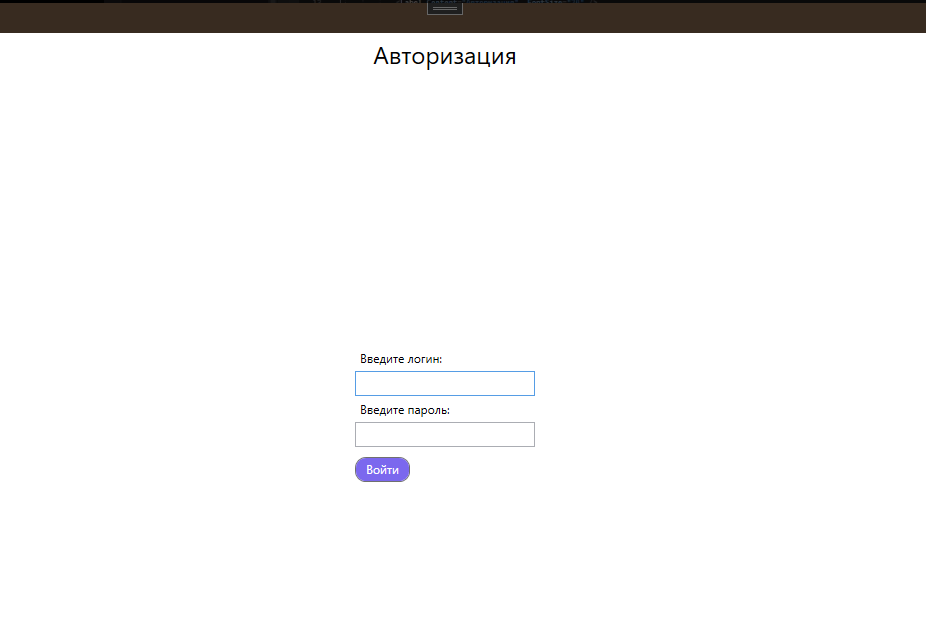
Все компоненты системы должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

− 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

1. **Разработка пользовательского интерфейса**

**Страница «Авторизации»:**

****

Рис№1 «Авторизация»

<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>

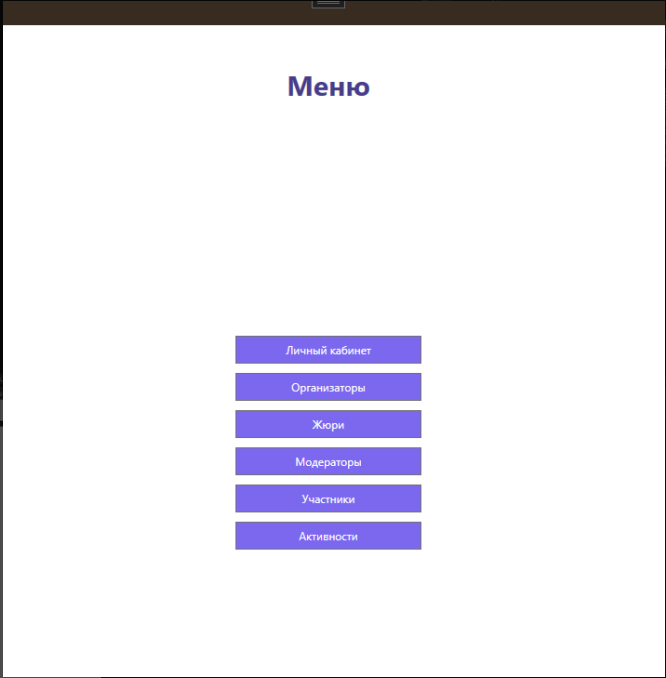
</Button.Resources>

</Button>

</StackPanel>

</StackPanel>

**Страница «Главного меню»:**

****

Рис№2 «Главное меню»

<Grid.RowDefinitions>

<RowDefinition Height="1"/>

<RowDefinition Height="5"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<Label Grid.ColumnSpan="2" Content="Меню" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Center" Name="stplButtons" >

<Button Name="btnPersonalAccount" Width="200" Height="30" Content="Личный кабинет" Click="btnPersonalAccount\_Click" Margin="10"/>

<Button Name="btnOrganizers" Width="200" Height="30" Content="Организаторы" Click="btnOrganizers\_Click"/>

<Button Name="btnJury" Width="200" Height="30" Content="Жюри" Click="btnJury\_Click" Margin="10"/>

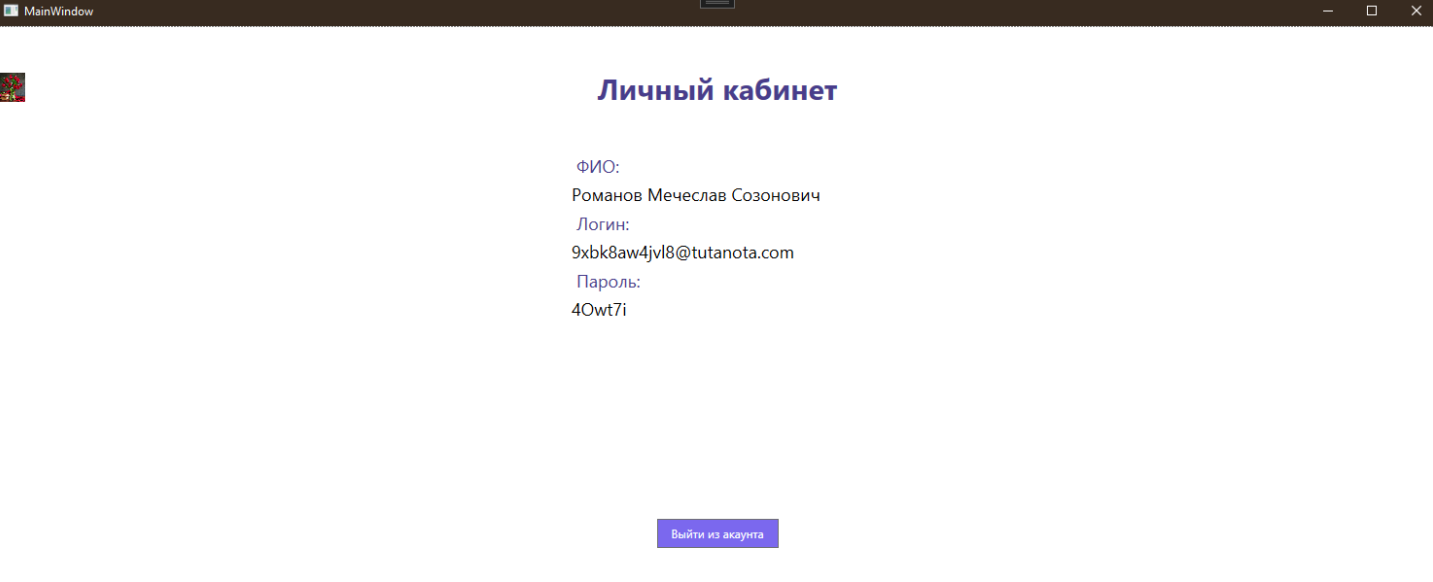
<Button Name="btnModerators" Width="200" Height="30" Content="Модераторы" Click="btnModerators\_Click"/>

<Button Name="btnParticipants" Width="200" Height="30" Content="Участники" Click="btnParticipants\_Click" Margin="10"/>

<Button Name="btnActivites" Width="200" Height="30" Content="Активности" Click="btnActivites\_Click" />

</StackPanel>

**Страница «Личный кабинет»:**



Рис№3 «Главное меню»

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="3\*"/>

<RowDefinition Height="2\*"/>

</Grid.RowDefinitions>

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

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/foto1.jpg"/>

</Button.Background>

</Button>

</StackPanel>

<Label HorizontalAlignment="Center" Content="Личный кабинет" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" HorizontalAlignment="Center" Grid.RowSpan="2">

<Label Content="ФИО:" Style="{StaticResource lblText}"/>

<TextBlock Name="txtFIO" Text="{Binding FIO}" Width="300" Height="25" FontSize="18" Style="{StaticResource ddlStyle}" MouseDown="TextBlock\_MouseDown"/>

<TextBox Name="txtFIO1" Text="{Binding FIO}" Width="300" Height="25" FontSize="18" Margin="0,-25,0,0" KeyUp="TextBox\_KeyUp" Visibility="Hidden"/>

<Label Content="Логин:" Style="{StaticResource lblText}"/>

<TextBlock Name="txtLogin" Text="{Binding Email}" Width="300" Height="25" FontSize="18" MouseDown="TextBlock\_MouseDown"/>

<TextBox Name="txtLogin1" Text="{Binding Email}" Width="300" Height="25" FontSize="18" Margin="0,-25,0,0" KeyUp="TextBox\_KeyUp" Visibility="Hidden"/>

<Label Content="Пароль:" Style="{StaticResource lblText}"/>

<TextBlock Name="txtPasword" Text="{Binding Pasword}" Width="300" Height="25" FontSize="18" MouseDown="TextBlock\_MouseDown"/>

<TextBox Name="txtPasword1" Text="{Binding Pasword}" Width="300" Height="25" FontSize="18" Margin="0,-25,0,0" KeyUp="TextBox\_KeyUp" Visibility="Hidden"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center" Grid.Row="2">

<Button Content="Выйти из акаунта" Click="Button\_Click" Style="{StaticResource btn1}"/>

</StackPanel>

**Страница «Организаторы»:**



Рис№4 «Организаторы»

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

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

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/foto1.jpg"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Список Организаторов" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" VerticalAlignment="Top" >

<Label Content="Поиск:" Style="{StaticResource lblText1}"/>

<TextBox Name="txtSearch" Width="125" Height="20" TextChanged="txtSearch\_TextChanged" Margin="10"/>

<TextBlock Text="Фильтр по полу"

FontSize="20"

Height="25"

Width="150"

Margin="5"

Foreground ="DarkSlateBlue"/>

<ComboBox Name="CmbFiltr"

SelectionChanged="CmbFiltr\_SelectionChanged" Width="125" Height="20">

<ComboBoxItem Content="Мужской"/>

<ComboBoxItem Content="Женский"/>

</ComboBox>

</StackPanel>

<DataGrid Name="dtgOrganizers" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" IsReadOnly="True" Margin="0,4,0,84" Grid.RowSpan="2" >

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto1.jpg"/>

</MenuItem.Icon>

</MenuItem>

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

<MenuItem.Icon>

<Image Source="/Photo/foto2.jpg"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Редактировать" Click="Menu\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto3.jpg"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Обновить" Click="Menu1\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto3.jpg"/>

</MenuItem.Icon>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

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

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Width="150" Stretch="UniformToFill" HorizontalAlignment="Left" Margin="4">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>

\Photo\foto1.jpg

</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

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

<DataGridTextColumn Header="Логин пользователя" Width="\*" Binding="{Binding Email}"/>

<DataGridTextColumn Header="Дата рождения" Width="\*" Binding ="{Binding DateOfBirth}"/>

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

<DataGridTextColumn Header="Номер телефона" Width="\*" Binding ="{Binding Phone}"/>

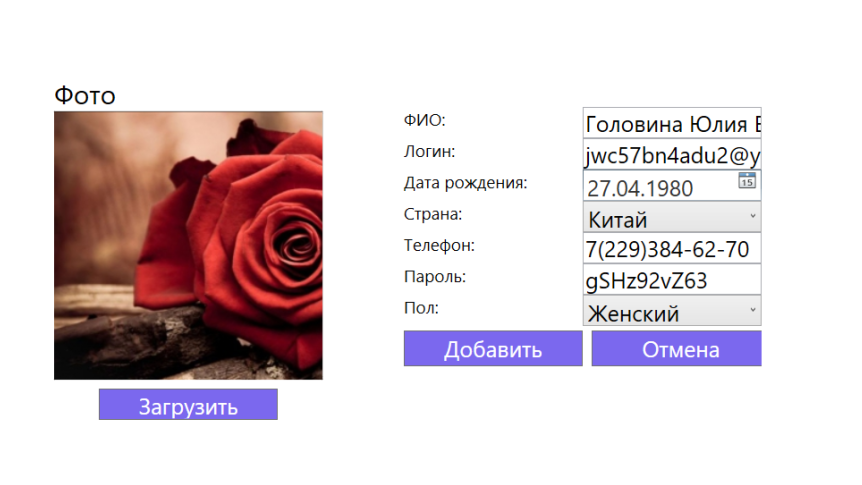
<DataGridTextColumn Header="Пароль" Width="\*" Binding ="{Binding Pasword}"/>

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

</DataGrid.Columns>

</DataGrid>

**Добавление «Организаторы»:**

****

Рис№5 «Добавление «Организаторов»»

<Grid.RowDefinitions>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left" Margin="324,150,0,-150">

<StackPanel Orientation="Vertical" HorizontalAlignment="Center" Margin="15 15 0 0">

<TextBlock Text="Фото" FontSize="30"/>

<StackPanel Name="FillNull" HorizontalAlignment="Left" VerticalAlignment="Center">

<Rectangle Width="300" Height="300" Fill="White" Stroke="Black" StrokeThickness="3"/>

<Path Data="M0 0L300 300M300 0L0 300" Stroke="Black" Margin="0 -300 0 0" StrokeThickness="1"/>

</StackPanel>

<Image Name="imageOrganizers" Width="300" Height="300" Margin="0 -300 0 0" Source="{Binding Photo}" Stretch="UniformToFill" HorizontalAlignment="Left"/>

<Button Content="Загрузить" Width="200" Height="35" FontSize="25" Click="PhotoLoad\_Click" HorizontalAlignment="Center" Margin="0 10 0 10"/>

</StackPanel>

</StackPanel>

<StackPanel Orientation="Vertical"

HorizontalAlignment="Left" Width="400" Height="700" Margin="730,200,0,260">

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

<TextBlock Text="ФИО:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtNazvanieMedicament" Text="{Binding FIO}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Логин:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtEmail" Text="{Binding Email}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Дата рождения:" Width="200" Height="35" FontSize="19"/>

<DatePicker Name="pickerdate" Width="200" Height="35" FontSize="25" SelectedDate="{Binding DateOfBirth}" />

</StackPanel>

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

<TextBlock Text="Страна:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBCountry" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDCountry}" />

</StackPanel>

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

<TextBlock Text="Телефон:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtPhone" Text="{Binding Phone}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Пароль:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtPasword" Text="{Binding Pasword}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Пол:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBPol" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDPol}"/>

</StackPanel>

<StackPanel Height="50" Orientation="Horizontal" Grid.Row="2">

<Button Content="Добавить" Name="BtnAddOrEdit" Click="BtnAddOrEdit\_Click" Width="200" Height="40" FontSize="25" Margin="0 0 10 0" />

<Button Content="Отмена" Click="BtnCancel\_Click" Width="200" Height="40" FontSize="25" />

</StackPanel>

</StackPanel>

**Страница «Жюри»:**



Рис№6 «Жюри»

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

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

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/foto1.jpg"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Список Жюри" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" VerticalAlignment="Top" >

<Label Content="Поиск:" Style="{StaticResource lblText1}"/>

<TextBox Name="txtSearch" Width="125" Height="20" TextChanged="txtSearch\_TextChanged" Margin="10"/>

<TextBlock Text="Фильтр по полу"

FontSize="20"

Height="25"

Width="150"

Margin="5"

Foreground ="DarkSlateBlue"/>

<ComboBox Name="CmbFiltr"

SelectionChanged="CmbFiltr\_SelectionChanged" Width="125" Height="20">

<ComboBoxItem Content="Мужской"/>

<ComboBoxItem Content="Женский"/>

</ComboBox>

</StackPanel>

<DataGrid Name="dtgJury" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" IsReadOnly="True" Margin="0,4,0,84" Grid.RowSpan="2" >

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto1.jpg"/>

</MenuItem.Icon>

</MenuItem>

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

<MenuItem.Icon>

<Image Source="/Photo/foto2.jpg"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Редактировать" Click="Menu\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto3.jpg"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Обновить" Click="Menu1\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto3.jpg"/>

</MenuItem.Icon>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

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

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Width="150" Stretch="UniformToFill" HorizontalAlignment="Left" Margin="4">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>

\Photo\foto1.jpg

</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

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

<DataGridTextColumn Header="Логин пользователя" Width="\*" Binding="{Binding Email}"/>

<DataGridTextColumn Header="Дата рождения" Width="\*" Binding ="{Binding DateOfBirth}"/>

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

<DataGridTextColumn Header="Номер телефона" Width="\*" Binding ="{Binding Phone}"/>

<DataGridTextColumn Header="Направление" Width="\*" Binding ="{Binding Direction.NameDirection}"/>

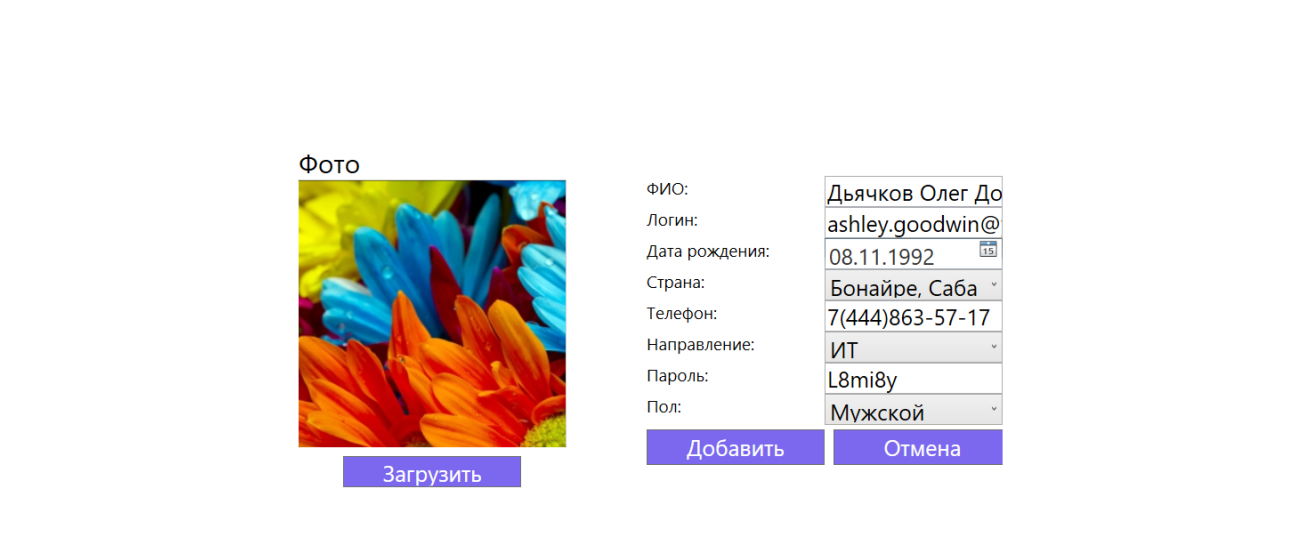
<DataGridTextColumn Header="Пароль" Width="\*" Binding ="{Binding Pasword}"/>

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

</DataGrid.Columns>

</DataGrid>

**Добавление «Жюри»:**



Рис№7 «Добавление «Жюри»»

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left" Margin="324,150,0,-150">

<StackPanel Orientation="Vertical" HorizontalAlignment="Center" Margin="15 15 0 0">

<TextBlock Text="Фото" FontSize="30"/>

<StackPanel x:Name="FillNull" HorizontalAlignment="Left" VerticalAlignment="Center">

<Rectangle Width="300" Height="300" Fill="White" Stroke="Black" StrokeThickness="3"/>

<Path Data="M0 0L300 300M300 0L0 300" Stroke="Black" Margin="0 -300 0 0" StrokeThickness="1"/>

</StackPanel>

<Image x:Name="imageJury" Width="300" Height="300" Margin="0 -300 0 0" Source="{Binding Photo}" Stretch="UniformToFill" HorizontalAlignment="Left"/>

<Button Content="Загрузить" Width="200" Height="35" FontSize="25" Click="PhotoLoad\_Click" HorizontalAlignment="Center" Margin="0 10 0 10"/>

</StackPanel>

</StackPanel>

<StackPanel Orientation="Vertical"

HorizontalAlignment="Left" Width="400" Margin="730,200,0,0">

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

<TextBlock Text="ФИО:" Width="200" Height="35" FontSize="19"/>

<TextBox x:Name="TxtNazvanieMedicament" Text="{Binding FIO}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Логин:" Width="200" Height="35" FontSize="19"/>

<TextBox x:Name="TxtEmail" Text="{Binding Email}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Дата рождения:" Width="200" Height="35" FontSize="19"/>

<DatePicker x:Name="pickerdate" IsDropDownOpen="False" SelectedDateFormat="Short" Width="200" Height="35" FontSize="25" SelectedDate="{Binding DateOfBirth}">

</DatePicker>

</StackPanel>

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

<TextBlock Text="Страна:" Width="200" Height="35" FontSize="19"/>

<ComboBox x:Name="CMBCountry" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDCountry}" />

</StackPanel>

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

<TextBlock Text="Телефон:" Width="200" Height="35" FontSize="19"/>

<TextBox x:Name="TxtPhone" Text="{Binding Phone}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Направление:" Width="200" Height="35" FontSize="19"/>

<ComboBox x:Name="CMBDirection" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDDirection}"/>

</StackPanel>

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

<TextBlock Text="Пароль:" Width="200" Height="35" FontSize="19"/>

<TextBox x:Name="TxtPasword" Text="{Binding Pasword}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Пол:" Width="200" Height="35" FontSize="19"/>

<ComboBox x:Name="CMBPol" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDPol}"/>

</StackPanel>

<StackPanel Height="50" Orientation="Horizontal" Grid.Row="2">

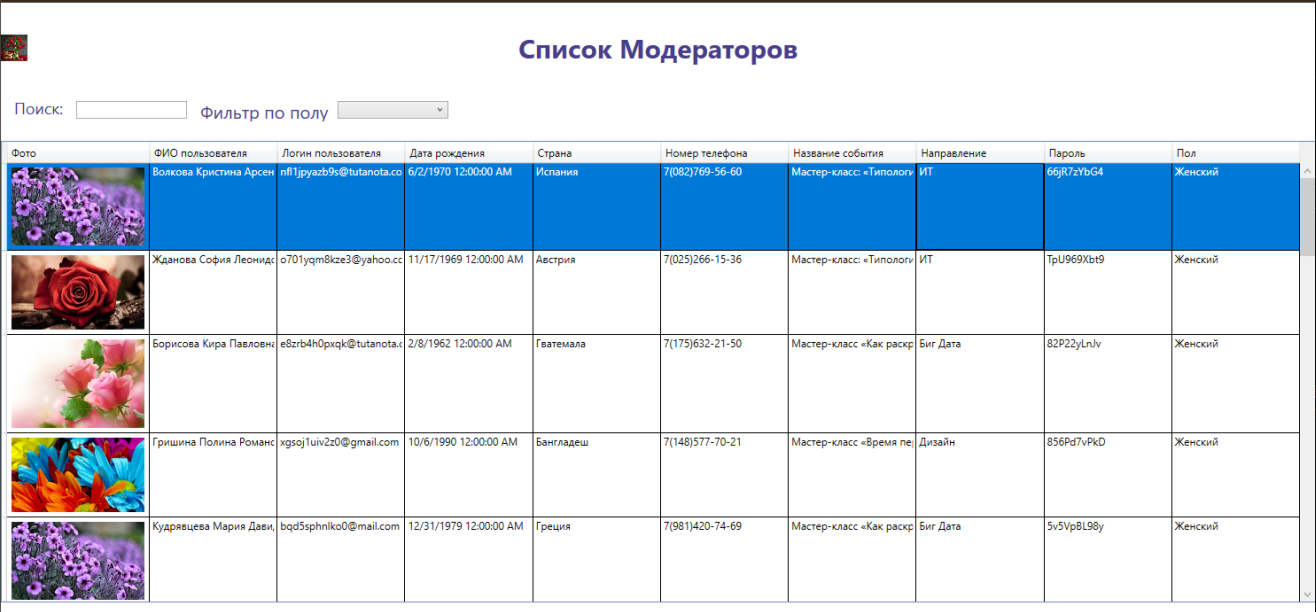
<Button Content="Добавить" x:Name="BtnAddOrEdit" Click="BtnAddOrEdit\_Click" Width="200" Height="40" FontSize="25" Margin="0 0 10 0" />

<Button Content="Отмена" Click="BtnCancel\_Click" Width="200" Height="40" FontSize="25" />

</StackPanel>

</StackPanel>

**Страница «Модераторы»:**

****

Рис№8 «Модераторы»

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

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

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/foto1.jpg"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Список Модераторов" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" VerticalAlignment="Top" >

<Label Content="Поиск:" Style="{StaticResource lblText1}"/>

<TextBox Name="txtSearch" Width="125" Height="20" TextChanged="txtSearch\_TextChanged" Margin="10"/>

<TextBlock Text="Фильтр по полу"

FontSize="20"

Height="25"

Width="150"

Margin="5"

Foreground ="DarkSlateBlue"/>

<ComboBox Name="CmbFiltr"

SelectionChanged="CmbFiltr\_SelectionChanged" Width="125" Height="20">

<ComboBoxItem Content="Мужской"/>

<ComboBoxItem Content="Женский"/>

</ComboBox>

</StackPanel>

<DataGrid Name="dtgModerators" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" IsReadOnly="True" Margin="0,4,0,84" Grid.RowSpan="2" >

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto1.jpg"/>

</MenuItem.Icon>

</MenuItem>

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

<MenuItem.Icon>

<Image Source="/Photo/foto2.jpg"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Редактировать" Click="Menu\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto3.jpg"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Обновить" Click="Menu1\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto3.jpg"/>

</MenuItem.Icon>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

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

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Width="150" Stretch="UniformToFill" HorizontalAlignment="Left" Margin="4">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>

\Photo\foto1.jpg

</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

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

<DataGridTextColumn Header="Логин пользователя" Width="\*" Binding="{Binding Email}"/>

<DataGridTextColumn Header="Дата рождения" Width="\*" Binding ="{Binding DateOfBirth}"/>

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

<DataGridTextColumn Header="Номер телефона" Width="\*" Binding ="{Binding Phone}"/>

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

<DataGridTextColumn Header="Направление" Width="\*" Binding ="{Binding Direction.NameDirection}"/>

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

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

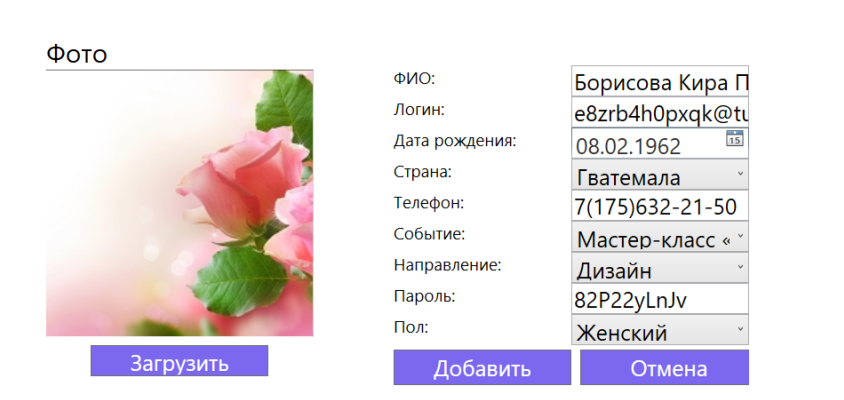
</DataGrid.Columns>

</DataGrid>

</StackPanel>

</StackPanel>

**Добавление «Модераторы»:**



Рис№9 «Добавление «Модераторы»»

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left" Margin="324,150,0,-150">

<StackPanel Orientation="Vertical" HorizontalAlignment="Center" Margin="15 15 0 0">

<TextBlock Text="Фото" FontSize="30"/>

<StackPanel Name="FillNull" HorizontalAlignment="Left" VerticalAlignment="Center">

<Rectangle Width="300" Height="300" Fill="White" Stroke="Black" StrokeThickness="3"/>

<Path Data="M0 0L300 300M300 0L0 300" Stroke="Black" Margin="0 -300 0 0" StrokeThickness="1"/>

</StackPanel>

<Image Name="imageModerators" Width="300" Height="300" Margin="0 -300 0 0" Source="{Binding Photo}" Stretch="UniformToFill" HorizontalAlignment="Left"/>

<Button Content="Загрузить" Width="200" Height="35" FontSize="25" Click="PhotoLoad\_Click" HorizontalAlignment="Center" Margin="0 10 0 10"/>

</StackPanel>

</StackPanel>

<StackPanel Orientation="Vertical"

HorizontalAlignment="Left" Width="400" Margin="730,200,0,0">

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

<TextBlock Text="ФИО:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtNazvanieMedicament" Text="{Binding FIO}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Логин:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtEmail" Text="{Binding Email}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Дата рождения:" Width="200" Height="35" FontSize="19"/>

<DatePicker Name="pickerdate" Width="200" Height="35" FontSize="25" SelectedDate="{Binding DateOfBirth}"/>

</StackPanel>

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

<TextBlock Text="Страна:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBCountry" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDCountry}" />

</StackPanel>

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

<TextBlock Text="Телефон:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtPhone" Text="{Binding Phone}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Событие:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBEvent" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDEvent}"/>

</StackPanel>

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

<TextBlock Text="Направление:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBDirection" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDDirection}"/>

</StackPanel>

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

<TextBlock Text="Пароль:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtPasword" Text="{Binding Pasword}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Пол:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBPol" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDPol}"/>

</StackPanel>

<StackPanel Height="50" Orientation="Horizontal" Grid.Row="2">

<Button Content="Добавить" Name="BtnAddOrEdit" Click="BtnAddOrEdit\_Click" Width="200" Height="40" FontSize="25" Margin="0 0 10 0" />

<Button Content="Отмена" Click="BtnCancel\_Click" Width="200" Height="40" FontSize="25" />

</StackPanel>

</StackPanel>

**Страница «Участники»:**



Рис№10 «Участники»

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

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

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/foto1.jpg"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Список Участники" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" VerticalAlignment="Top" >

<Label Content="Поиск:" Style="{StaticResource lblText1}"/>

<TextBox Name="txtSearch" Width="125" Height="20" TextChanged="txtSearch\_TextChanged" Margin="10"/>

<TextBlock Text="Фильтр по полу"

FontSize="20"

Height="25"

Width="150"

Margin="5"

Foreground ="DarkSlateBlue"/>

<ComboBox Name="CmbFiltr"

SelectionChanged="CmbFiltr\_SelectionChanged" Width="125" Height="20">

<ComboBoxItem Content="Мужской"/>

<ComboBoxItem Content="Женский"/>

</ComboBox>

</StackPanel>

<DataGrid Name="dtgParticipants" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" IsReadOnly="True" Margin="0,4,0,84" Grid.RowSpan="2" >

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto1.jpg"/>

</MenuItem.Icon>

</MenuItem>

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

<MenuItem.Icon>

<Image Source="/Photo/foto2.jpg"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Редактировать" Click="Menu\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto3.jpg"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Обновить" Click="Menu1\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto3.jpg"/>

</MenuItem.Icon>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

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

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Width="150" Stretch="UniformToFill" HorizontalAlignment="Left" Margin="4">

<Image.Source>

<Binding Path="Photo">

<Binding.TargetNullValue>

<ImageSource>

\Photo\foto1.jpg

</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

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

<DataGridTextColumn Header="Логин пользователя" Width="\*" Binding="{Binding Email}"/>

<DataGridTextColumn Header="Дата рождения" Width="\*" Binding ="{Binding DateOfBirth}"/>

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

<DataGridTextColumn Header="Номер телефона" Width="\*" Binding ="{Binding Phone}"/>

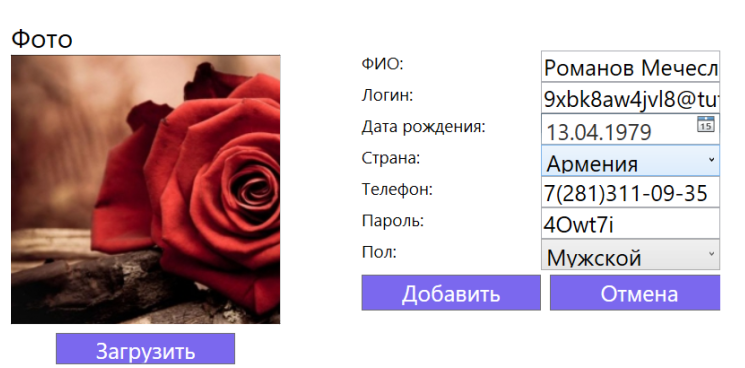
<DataGridTextColumn Header="Пароль" Width="\*" Binding ="{Binding Pasword}"/>

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

</DataGrid.Columns>

</DataGrid>

**Добавление «Участники»:**



Рис№11 «Добавление «Участники»»

<Grid.RowDefinitions>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Left" Margin="324,150,0,-150">

<StackPanel Orientation="Vertical" HorizontalAlignment="Center" Margin="15 15 0 0">

<TextBlock Text="Фото" FontSize="30"/>

<StackPanel Name="FillNull" HorizontalAlignment="Left" VerticalAlignment="Center">

<Rectangle Width="300" Height="300" Fill="White" Stroke="Black" StrokeThickness="3"/>

<Path Data="M0 0L300 300M300 0L0 300" Stroke="Black" Margin="0 -300 0 0" StrokeThickness="1"/>

</StackPanel>

<Image Name="imageParticipants" Width="300" Height="300" Margin="0 -300 0 0" Source="{Binding Photo}" Stretch="UniformToFill" HorizontalAlignment="Left"/>

<Button Content="Загрузить" Width="200" Height="35" FontSize="25" Click="PhotoLoad\_Click" HorizontalAlignment="Center" Margin="0 10 0 10"/>

</StackPanel>

</StackPanel>

<StackPanel Orientation="Vertical"

HorizontalAlignment="Left" Width="400" Height="700" Margin="730,200,0,260">

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

<TextBlock Text="ФИО:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtNazvanieMedicament" Text="{Binding FIO}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Логин:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtEmail" Text="{Binding Email}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Дата рождения:" Width="200" Height="35" FontSize="19"/>

<DatePicker Name="pickerdate" Width="200" Height="35" FontSize="25" SelectedDate="{Binding DateOfBirth}" />

</StackPanel>

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

<TextBlock Text="Страна:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBCountry" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDCountry}" />

</StackPanel>

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

<TextBlock Text="Телефон:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtPhone" Text="{Binding Phone}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Пароль:" Width="200" Height="35" FontSize="19"/>

<TextBox Name="TxtPasword" Text="{Binding Pasword}" Width="200" Height="35" FontSize="25" />

</StackPanel>

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

<TextBlock Text="Пол:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBPol" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDPol}"/>

</StackPanel>

<StackPanel Height="50" Orientation="Horizontal" Grid.Row="2">

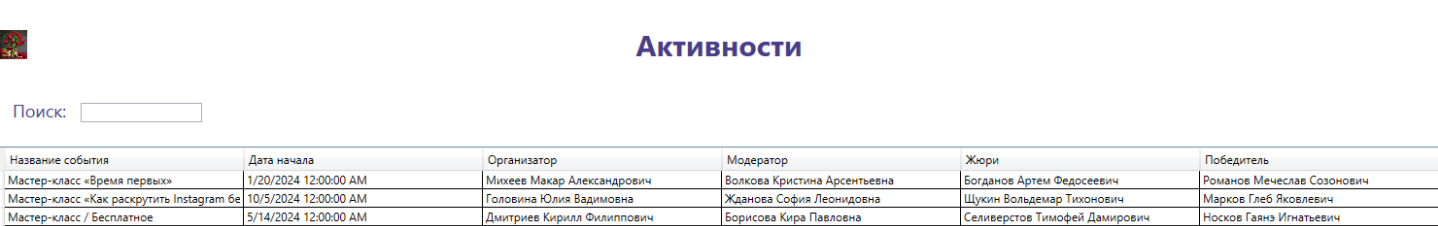
<Button Content="Добавить" Name="BtnAddOrEdit" Click="BtnAddOrEdit\_Click" Width="200" Height="40" FontSize="25" Margin="0 0 10 0" />

<Button Content="Отмена" Click="BtnCancel\_Click" Width="200" Height="40" FontSize="25" />

</StackPanel>

</StackPanel>

**Страница «Активности»:**



Рис№12 «Участники»

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="0.5\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

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

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/foto1.jpg"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Активности" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" VerticalAlignment="Top" >

<Label Content="Поиск:" Style="{StaticResource lblText1}"/>

<TextBox Name="txtSearch" Width="125" Height="20" TextChanged="txtSearch\_TextChanged" Margin="10"/>

</StackPanel>

<DataGrid Name="dtgActivites" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" IsReadOnly="True" Margin="0,4,0,84" Grid.RowSpan="2">

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto1.jpg"/>

</MenuItem.Icon>

</MenuItem>

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

<MenuItem.Icon>

<Image Source="/Photo/foto2.jpg"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Редактировать" Click="Menu\_Click">

<MenuItem.Icon>

<Image Source="/Photo/foto3.jpg"/>

</MenuItem.Icon>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

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

<DataGridTextColumn Header="Дата начала" Width="\*" Binding ="{Binding TimeStart}"/>

<DataGridTextColumn Header="Организатор" Width="\*" Binding ="{Binding Organizers.FIO}"/>

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

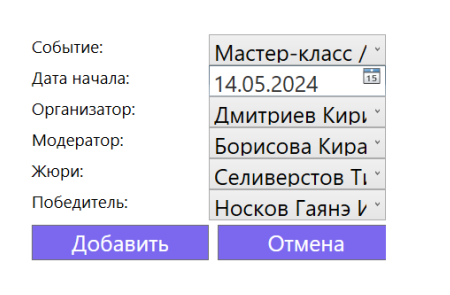
<DataGridTextColumn Header="Жюри" Width="\*" Binding ="{Binding Jury.FIO}"/>

<DataGridTextColumn Header="Победитель" Width="\*" Binding ="{Binding Participants.FIO}"/>

</DataGrid.Columns>

</DataGrid>

**Добавление «Активности»:**



Рис№13 «Добавление «Активности»»

<StackPanel Orientation="Vertical"

HorizontalAlignment="Left" Width="400" Margin="500,200,0,0">

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

<TextBlock Text="Событие:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBEvent" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDEvent}"/>

</StackPanel>

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

<TextBlock Text="Дата начала:" Width="200" Height="35" FontSize="19"/>

<DatePicker Name="pickerdate" Width="200" Height="35" FontSize="25" SelectedDate="{Binding TimeStart}"/>

</StackPanel>

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

<TextBlock Text="Организатор:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBOrganizers" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDOrganizers}"/>

</StackPanel>

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

<TextBlock Text="Модератор:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBModerators" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDModerators}"/>

</StackPanel>

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

<TextBlock Text="Жюри:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBJury" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDJury}"/>

</StackPanel>

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

<TextBlock Text="Победитель:" Width="200" Height="35" FontSize="19"/>

<ComboBox Name="CMBParticipants" Width="200" Height="35" FontSize="25" SelectedValue="{Binding IDParticipants}"/>

</StackPanel>

<StackPanel Height="50" Orientation="Horizontal" Grid.Row="2">

<Button Content="Добавить" Name="BtnAddOrEdit" Click="BtnAddOrEdit\_Click" Width="200" Height="40" FontSize="25" Margin="0 0 10 0" />

<Button Content="Отмена" Click="BtnCancel\_Click" Width="200" Height="40" FontSize="25" />

</StackPanel>

</StackPanel>

**Стиль:**

<Style TargetType="Button">

<Setter Property="Background" Value="MediumSlateBlue"/>

<Setter Property="Foreground" Value="White"/>

</Style>

<Style TargetType="Label" x:Key="lblTitle">

<Setter Property="FontSize" Value="30"/>

<Setter Property="Foreground" Value="DarkSlateBlue"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="VerticalAlignment" Value="Center"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

</Style>

<Style TargetType="Button" x:Key="btnback">

<Setter Property="Content" Value="Вернуться назад"/>

</Style>

<Style TargetType="Label" x:Key="lblText">

<Setter Property="FontSize" Value="18"/>

<Setter Property="Foreground" Value="DarkSlateBlue"/>

<Setter Property="Margin" Value="0,0,150,0"/>

</Style>

<Style x:Key="ddlStyle" TargetType="TextBlock">

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="IsEnabled" Value="True" />

</Trigger>

</Style.Triggers>

</Style>

<Style TargetType="StackPanel" x:Key="stc">

<Setter Property="Margin" Value="0,0,0,40"/>

<Setter Property="VerticalAlignment" Value="Bottom"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

</Style>

<Style TargetType="Label" x:Key="lblText1">

<Setter Property="FontSize" Value="18"/>

<Setter Property="Foreground" Value="DarkSlateBlue"/>

<Setter Property="Margin" Value="10,0,0,0"/>

</Style>

<Style TargetType="Label" x:Key="lblText2">

<Setter Property="FontSize" Value="16"/>

<Setter Property="Foreground" Value="DarkSlateBlue"/>

</Style>

<Style TargetType="Button" x:Key="btn1">

<Setter Property="Background" Value="MediumSlateBlue"/>

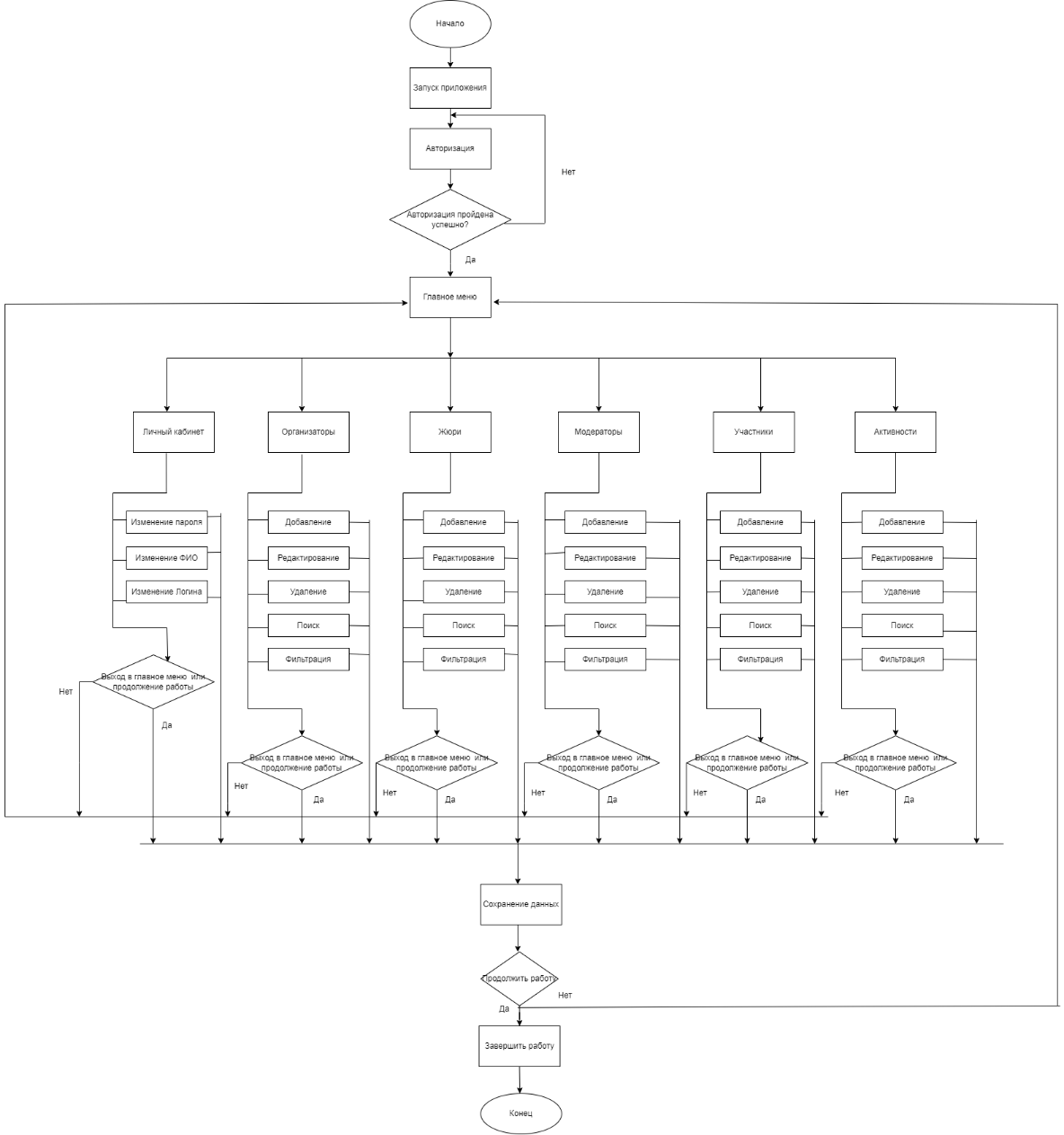
<Setter Property="Foreground" Value="White"/>

<Setter Property="Width" Value="125"/>

<Setter Property="Height" Value="30"/>

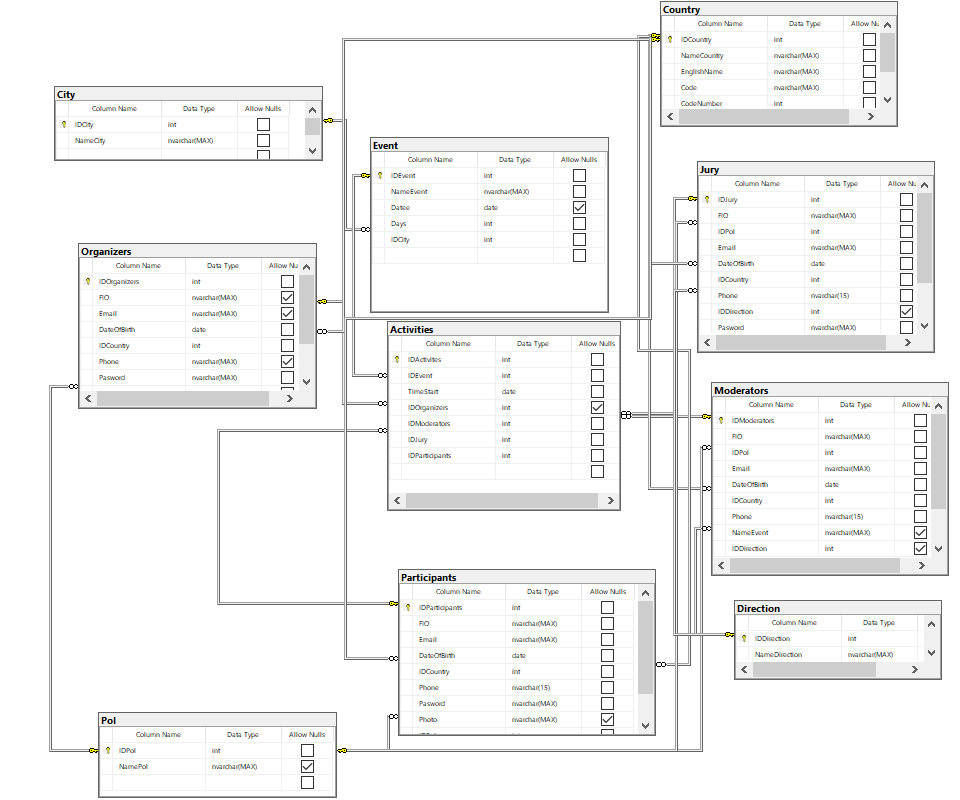
</Style>

1. **Алгоритм работы приложения(блоксхема)**



Рис№14 «Блок-схема»

1. **Проектирование ER-диаграммы**

****

Рис№15 «ER-диаграмма»

1. **Проектирование и разработка базы данных**

Шаг№1

Создаем базу данных «Conference»



Рис.16 «Создание Базы Данных»

Шаг№2

Создаем таблицы «Activities», «City», «Country», «Direction», «Event», «Jury», «Moderators», «Organizers», «Participants», «Pol».

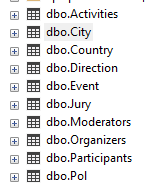


Рис.17 «Создание таблиц»

Шаг№3

Создаем диаграмму(см пункт 4)

Шаг№4

Заполняем таблицы данными через функцию «Задачи» - «Импорт данных»

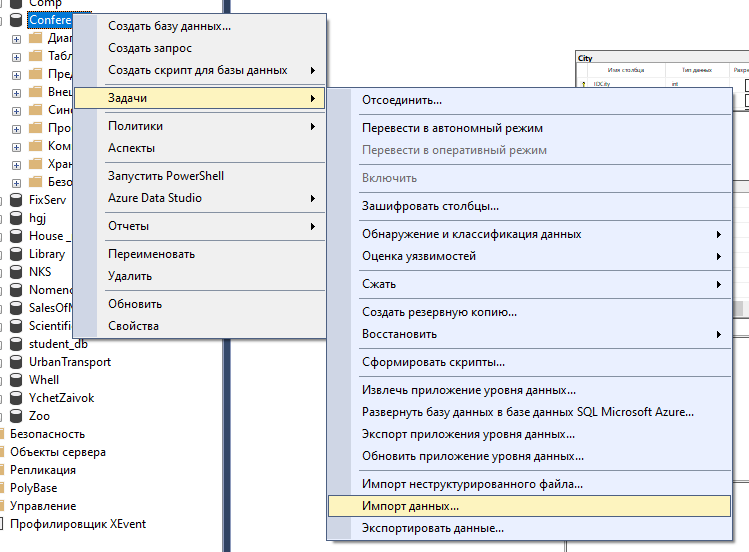


Рис.18 «Заполнение таблицы»

Шаг№5

Открывается мастер импорта нажимаем «Next»

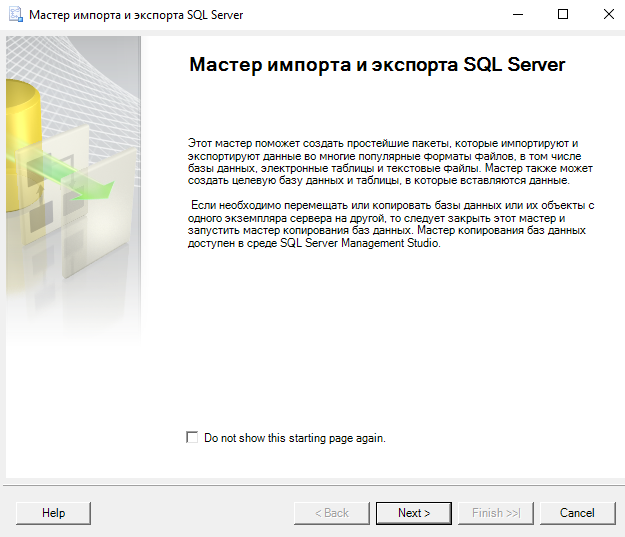


Рис.19 «Мастер импорта»

Шаг№6

Выбираем источник данных

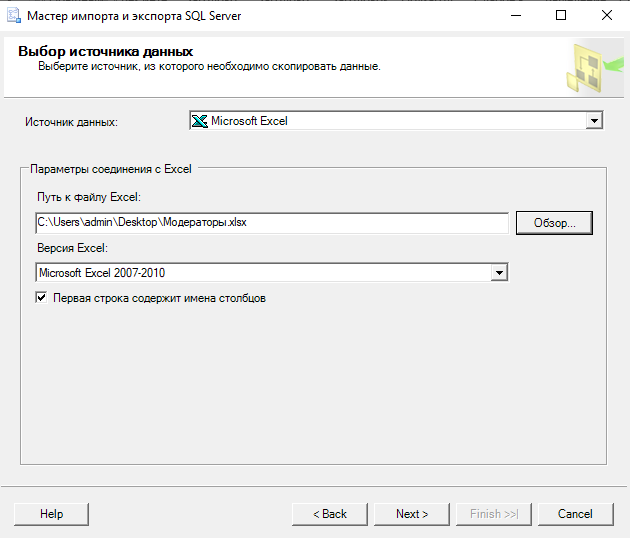


Рис.20 «Источник данных»

Шаг№7

Выбираем назначение

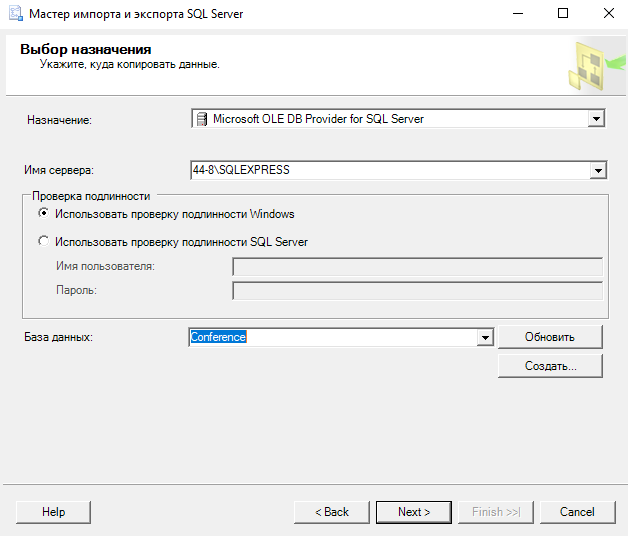


Рис.21 «Назначение»

Шаг№8

Выбор копирования данных

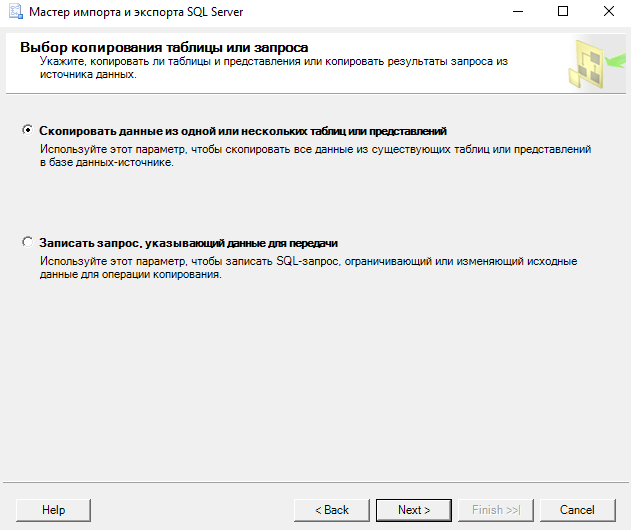


Рис.22 «Копирование данных»

Шаг№9

Выбор исходных таблиц и представлений

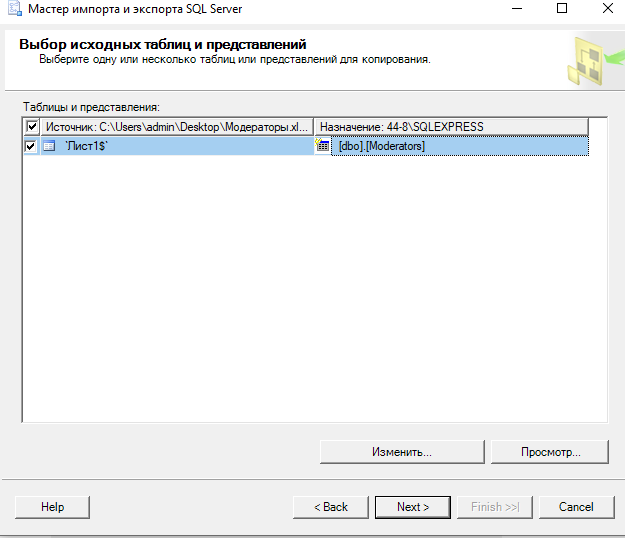


Рис.23 «Исходные таблицы»

Шаг№10

Просматриваем на корректность данных в меню «Просмотр»

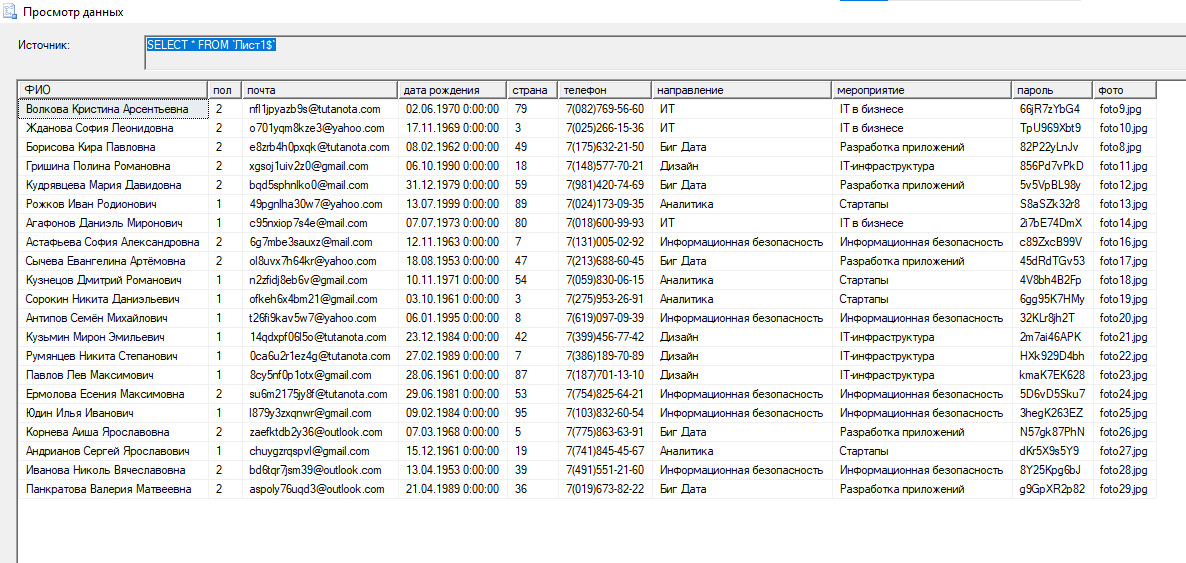


Рис.24 «Проверка данных»

Шаг№11

Просматриваем на тип данных в меню «Изменить»

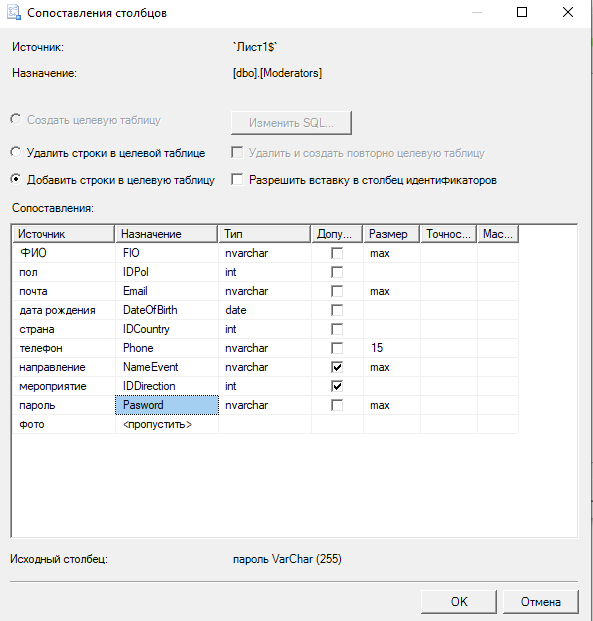


Рис.25 «Проверка типа дынных»

Шаг№12

Сохранение и запуск пакетов

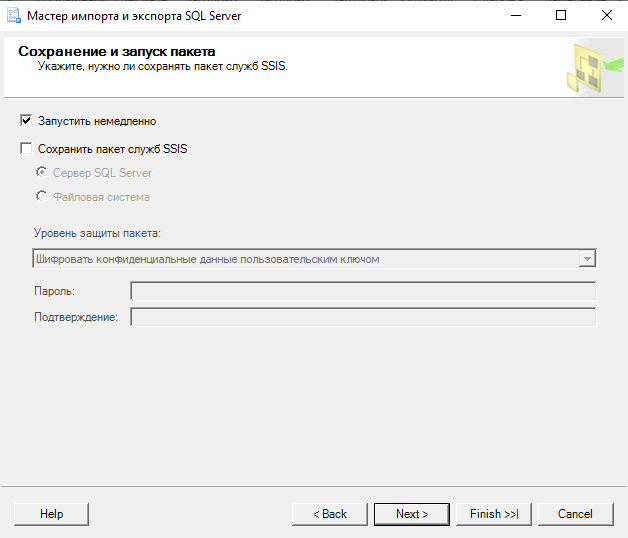


Рис.26 «Сохранение данных»

Шаг№13

Нажимаем кнопку «Finish», и ждем переноса данных.

Шаг№14

Получаем заполненную таблицу

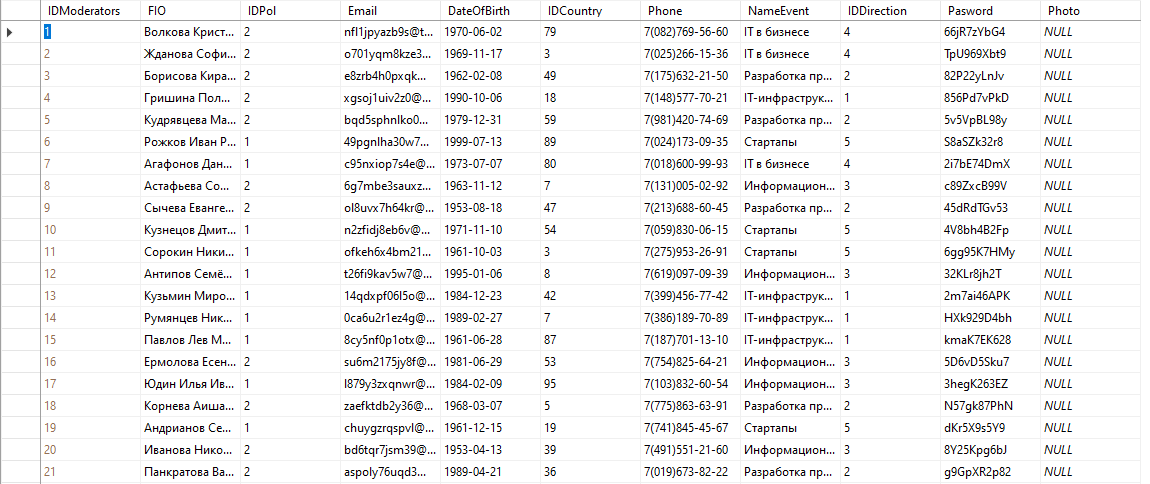


Рис.27 «Заполненная таблица»

**Запросы**

Запросы№1

select \* from Country

WHERE NameCountry = 'Ангилья'



Рис.28 «Запрос на вывод определенной страны»

Запросы№2

SELECT \* FROM City WHERE NameCity LIKE 'А%'

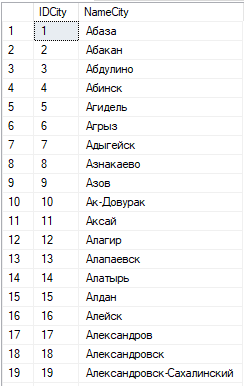


Рис.29 «Запрос на вывод по первой буквы города»

Запрос№3

select \* from Country

where NameCountry IN ('Армения', 'Аргентина')

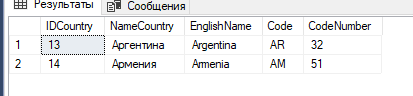


Рис.30 «Запрос на вывод нескольких стран»

Запрос№4

select \* from Participants

ORDER BY Fio

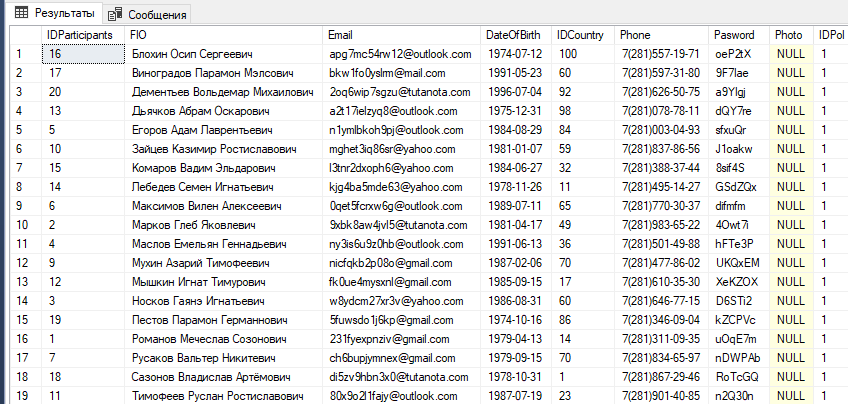


Рис.31 «Запрос на сортировку по FIO»

Запрос№5

select \* from City

JOIN Event ON City.IDCity = Event.IDCity

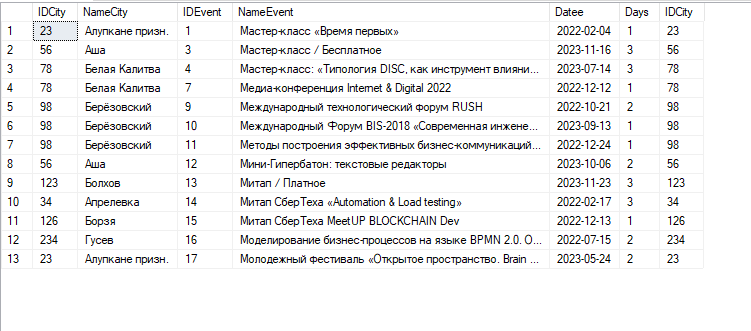
where Event.IDCity >10

Рис.32 «Запрос на сравнение ID и вывод ID больше 10»

1. **Руководство системного программиста**

**6.1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ**

**6.1.1 Назначение и функции программы**

**Смотреть пункт 1.2 и 1.3**

**6.1.2 Сведения о технических и программных средствах, обеспечивающие выполнение программы**

**6.1.2.1 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.

**6.1.2.2 Требования к техническим средствам**

Таблица №2 «Минимальные требования»

|  |  |
| --- | --- |
| Процессор | Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz 2.70 GHz |
| 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 |

**6.2 СТРУКТУРА ПРОГРАММЫ**

**6.2.1** **Структура программы и её составные части**

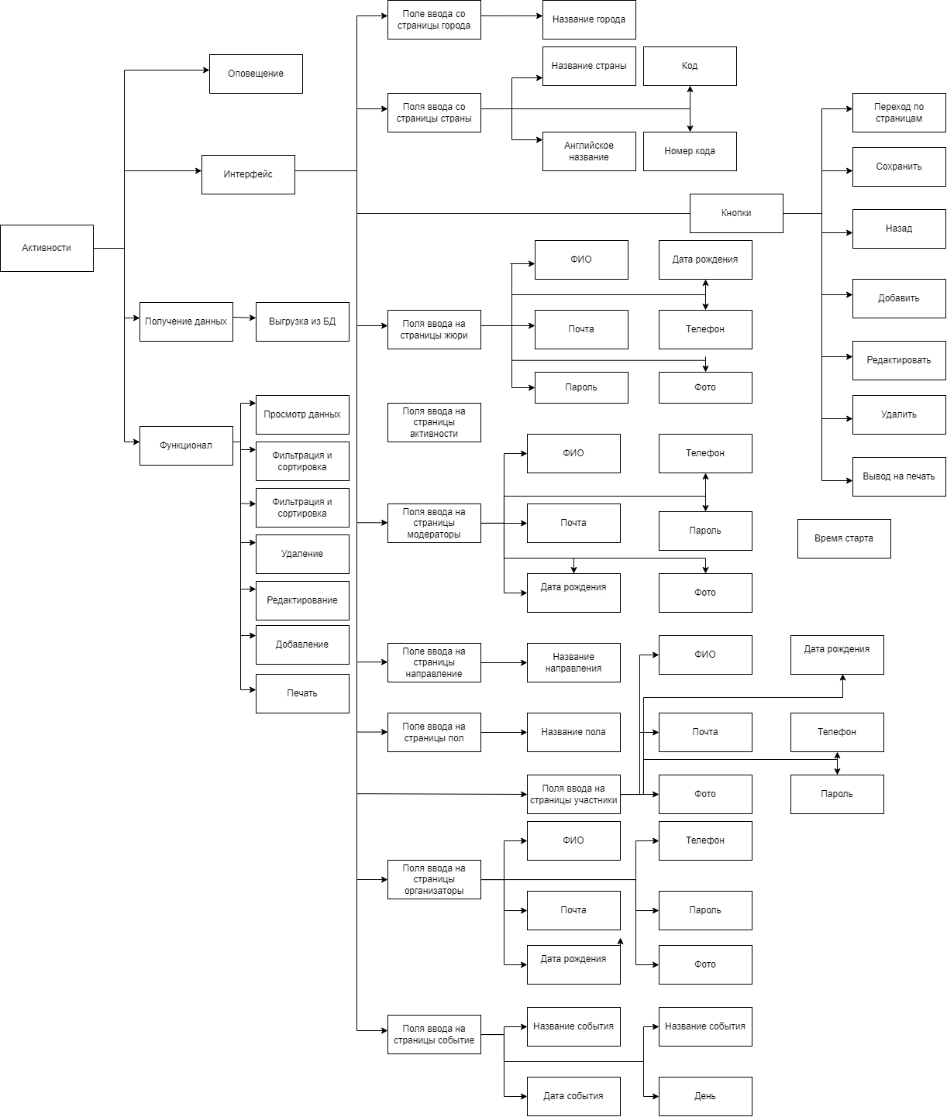


Рис.33 «Структура программы»

**6.2.2** **Связи между составными частями программы**

При запуске приложения пользователь заходит в окно авторизации где дальше он может перейти на страницу Меню после внесения данных или на страницу Регистрации. На странице регистрации пользователь вводит свои данные после чего переход на страницу Меню. На странице Меню пользователь может перейти на страницы: Акаунт, События, Активности (если пользователем является Модератор или Организатор то в Меню добавляется страница Просмотра пользователей). Перейдя на страницу Акаунт пользователей может просмотреть свои данные, изменить пароль и выйти из акаунта. На станице События пользователь может посмотреть информацию о событии и возможность вернуться на страницу меню. На странице Активности можно узнать информацию о результатах события и возможность вернуться на страницу меню. На странице Просмотра пользователя можно узнать информацию о всех пользователях и также можно вернуться на страницу меню.

**6.2.3 Связи программы с другими программами**

Чтобы сохранять данные из таблиц: Модераторы, Жури, Участники, Организаторы, Активности, можно с помощью документа экспортировать в Microsoft Word. И для сохранения базы данных использую приложение SQL Server Management Studio.

**6.3 НАСТРОЙКА ПРОГРАММЫ И СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ**

**6.3.1** **Настройка работы БД в SQL Server Management Studio**

Найти Management Studio в меню «ПУСК», запустить.

Открыть окно соединения с сервером SQL Server, где выбрать:

— Ядро СУБД в поле «Тип сервера»;

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

— Проверка подлинности Windows в поле «Проверка подлинности».

Нажать «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.

**6.3.2** **Сохранение БД**

1. Создать схему БД в SQL Server Management Studio, состоящую из таблиц.
2. Заполнить таблицы данными.
3. БД можно сохранить двумя способами, можно создать скрипт БД или создать резервную копию БД.

**Создание резервной копии БД:**

1. Слева в списке найти свою БД и нажать по ней ПКМ.

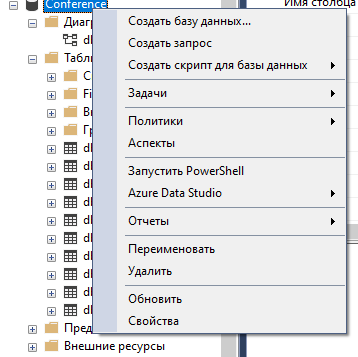


Рис.34 «Контекстное меню»

1. Нажать на пункт «Задачи» и в раскрывающимся списке выбрать пункт «Создать резервную копию».

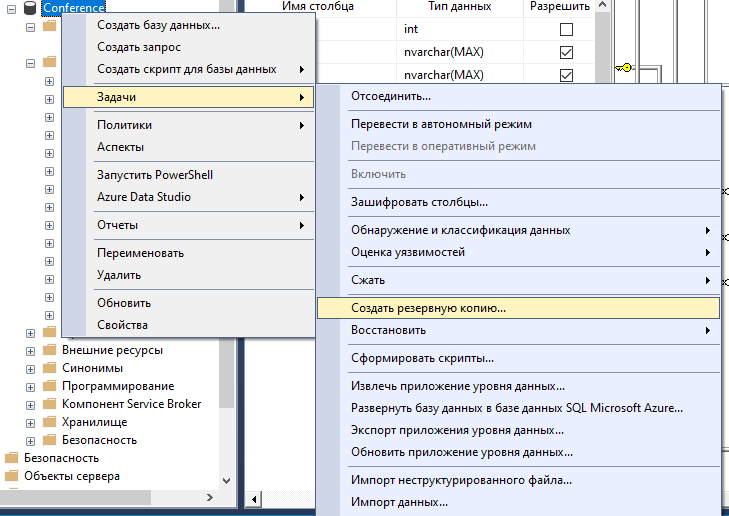


Рис.35 «Создание копии»

1. В открывшемся окне оставить настройки сохранения по стандарту и нажать ОК.

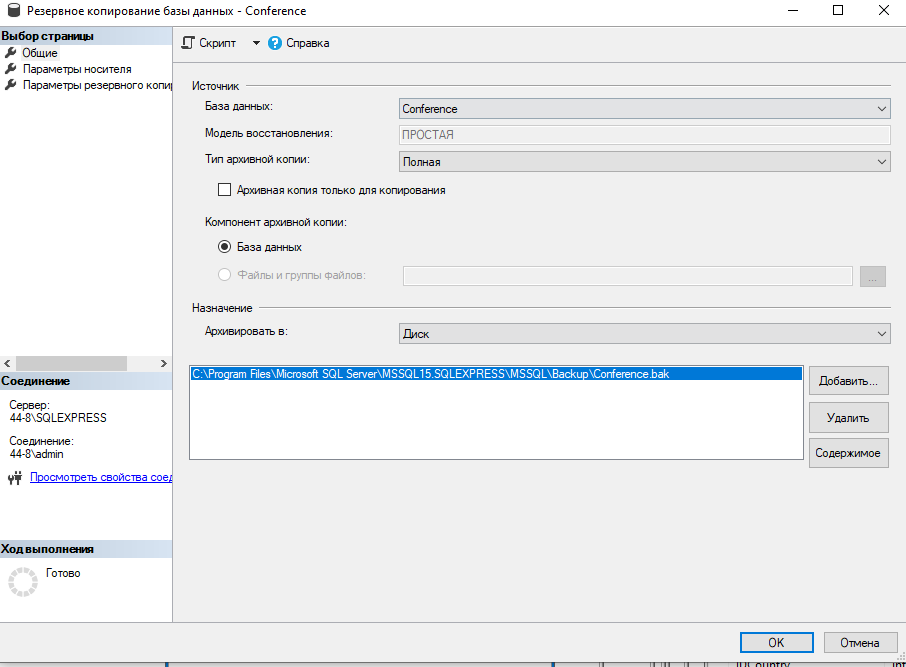


Рис.36 «Настройка сохранения копии»

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



Рис.37 «Резервная копия БД»

**Создание скрипта БД:**

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

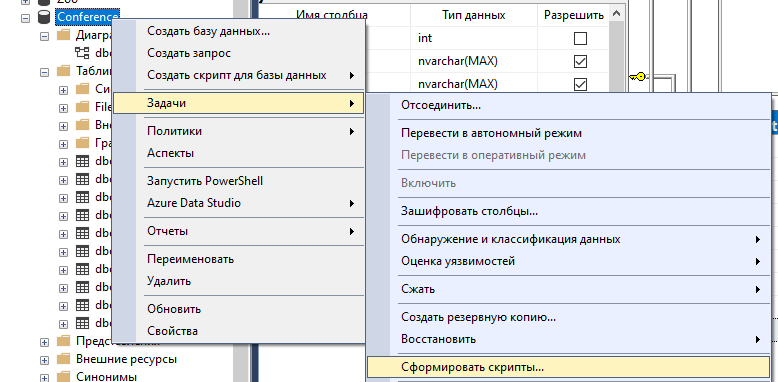


Рис.38 «Создание скрипта»

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

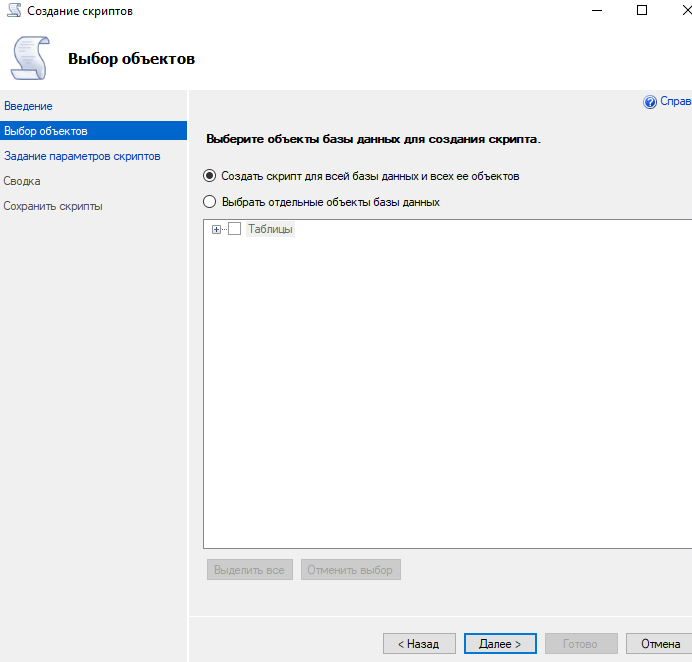


Рис.39 «Выбор объектов»

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

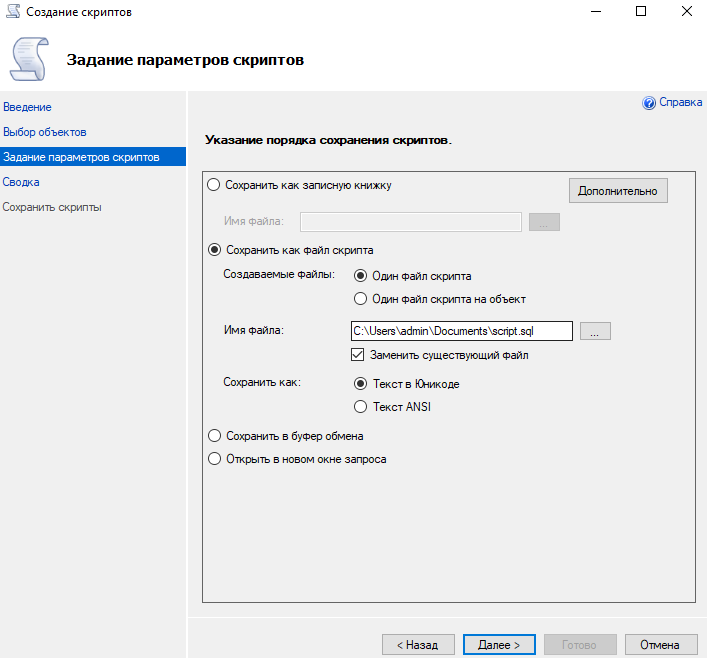


Рис.40 «Задание параметров скрипта»

1. В окне «Сводка» нажать Далее.

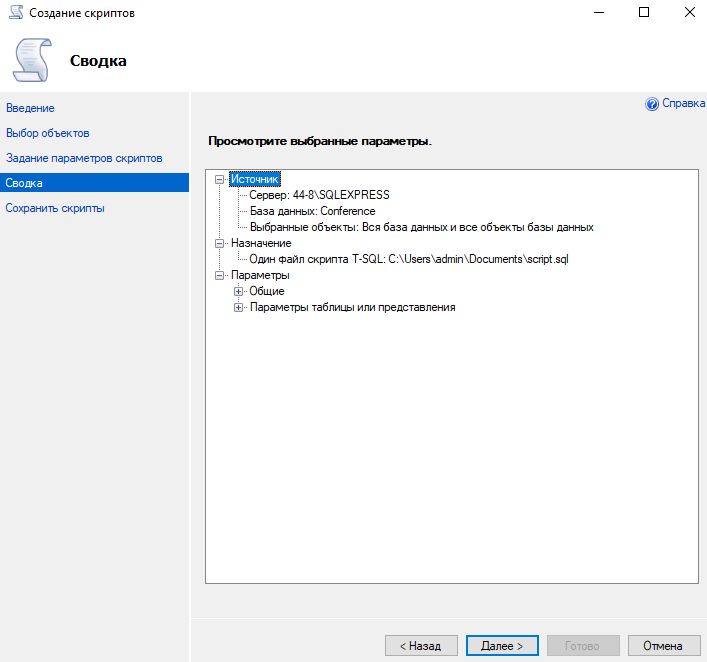


Рис.41 «Сводка»

1. В окне «Сохранить скрипты» подождать создания и сохранения скрипта и нажать Готово.

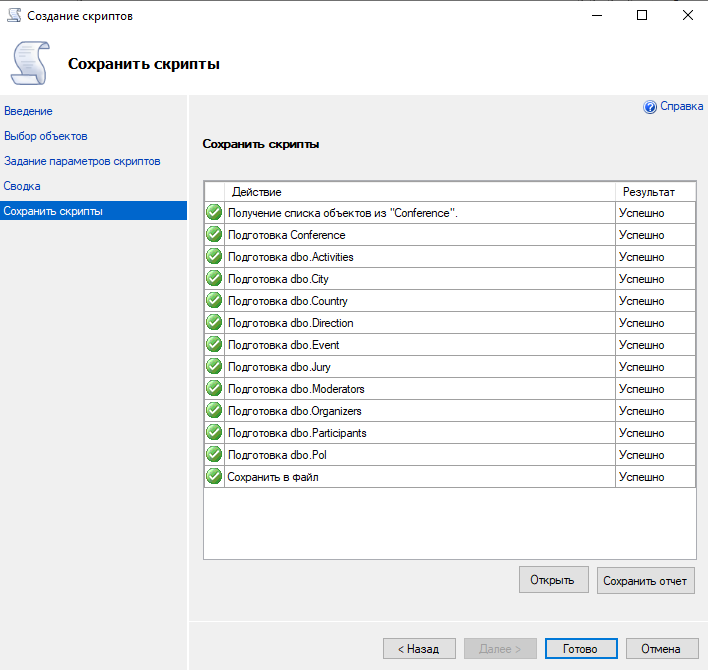


Рис.42 «Сохранение скрипта»

**6.3.4** **Перенос БД и её повторное подключение**

1. На новом ПК зайти в SQL Server Management Studio и создать БД с таким же именем что и в скрипте.



Рис.43 «Созданная БД»

1. Перенести скрипт своей БД в SQL Server Management Studio.

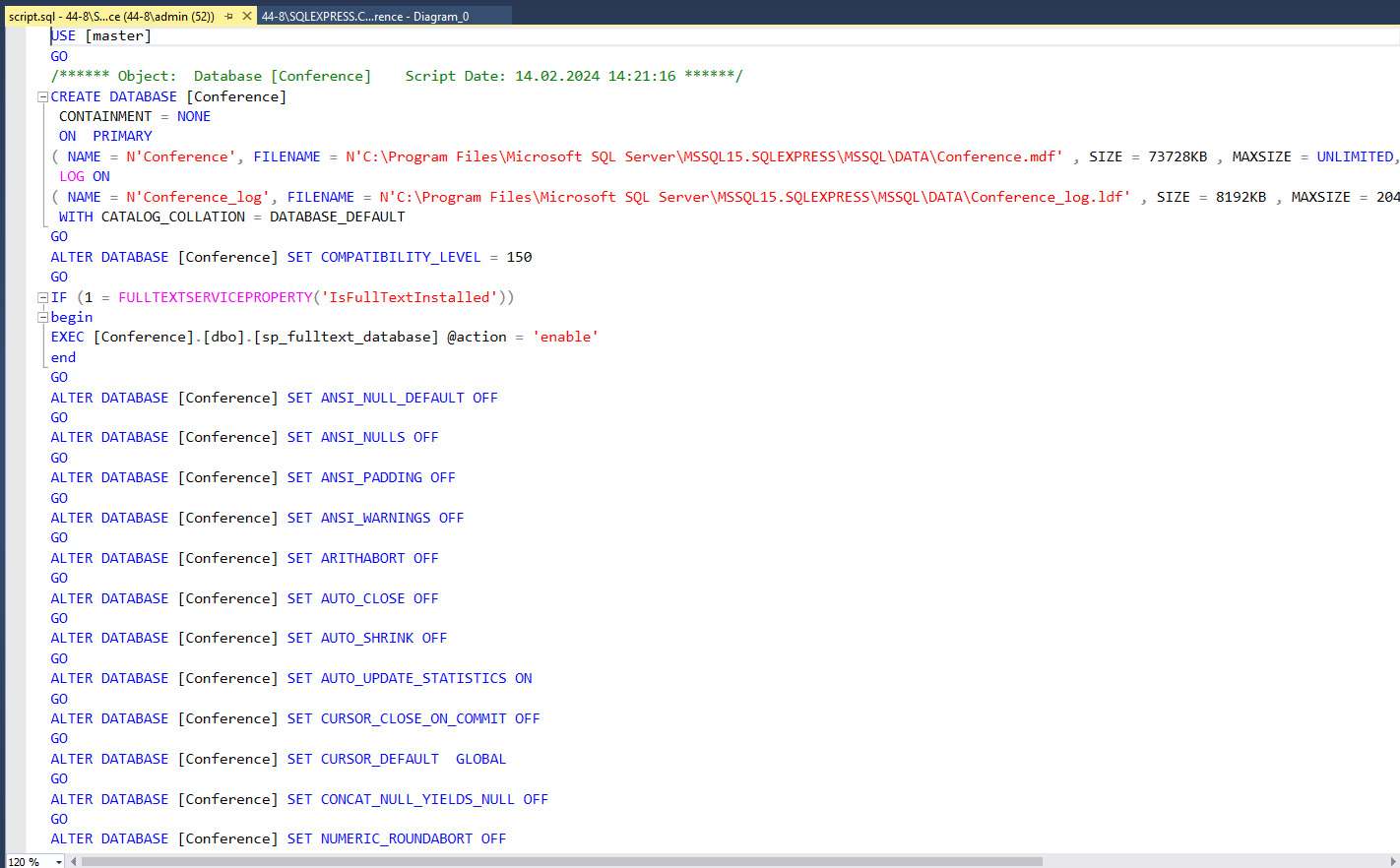


Рис.44 «Скрипт БД»

1. Удалить строчки от CREATE DATABASE и до строчки END и сверху нажать на Выполнить.

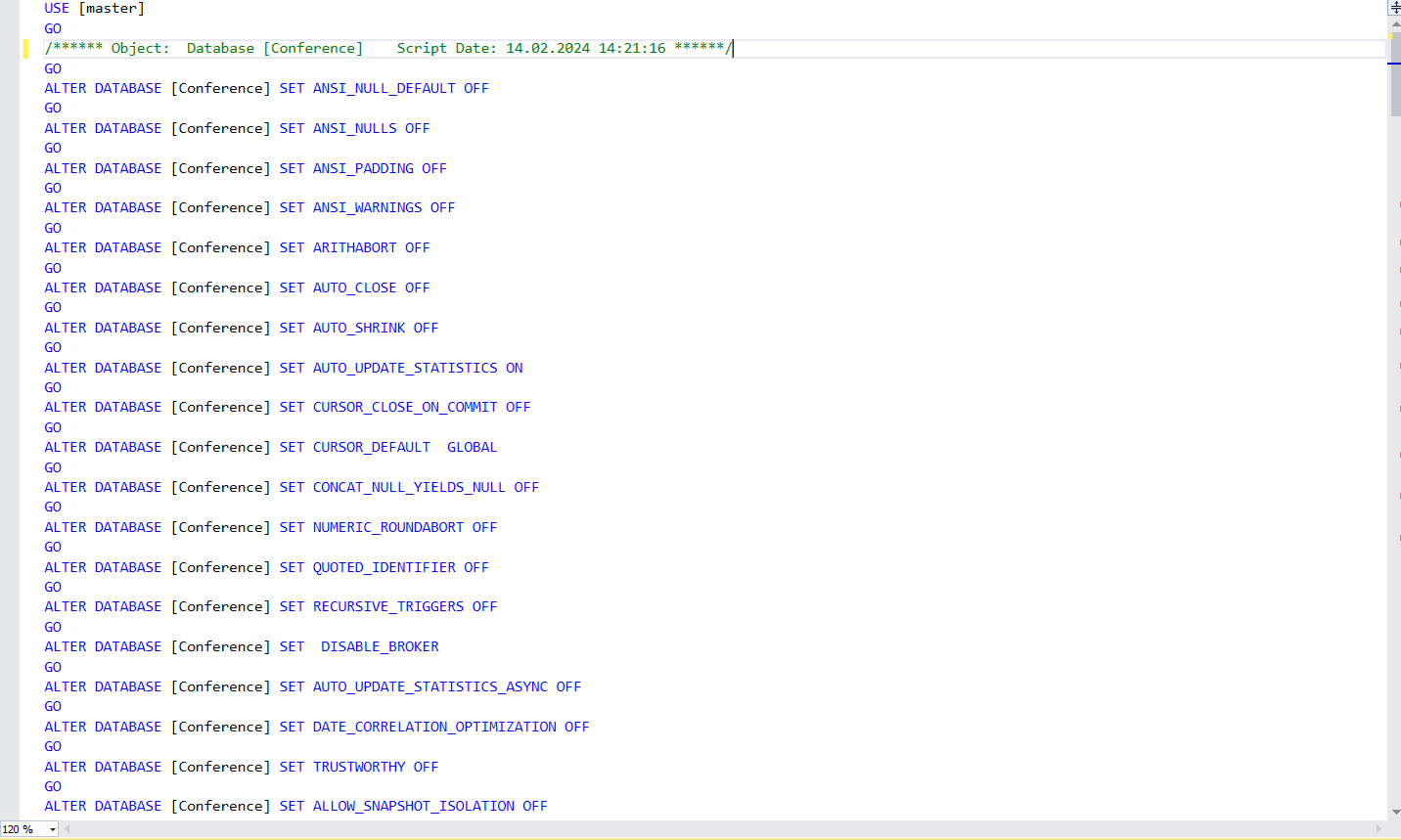


Рис.45 «Удаление строчек»



Рис.46 «Выполнение скрипта»

1. Запустить свою программу и нажать в обозревателе решений на пункт «App.config», чтобы переподключить БД.



Рис.47 «Вкладка переподключения БД»

1. Найти имя своего сервера SQL Server Management Studio и изменить его на имя сервера, который находится на данном ПК.



Рис.48 «Переподключение сервера»

**6.4 ПРОВЕРКА ПРОГРАММЫ И СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ**

**6.4.1** **Способы проверки**

Проверка работоспособности функции «Отображения информации» проводится следующим образом (таблица №2).

Таблица №3 «Проверка отображения информации»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Осуществить открытие exe-файла программы | Успешный запуск программы |
| 2 | **Проверка отображения информации пользователю ПО**  После запуска программы нажать кнопку «Войти»  После входа в программу появляется окно с информацией | Отображение информации в виде таблицы с данными |

Проверка работоспособности функции «Добавление данных» проводится следующим образом (таблица №3).

Таблица №4 «Проверка добавления данных»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Нажать на кнопку «Добавить» | Открытие окна с добавлением данных |
| 2 | **Проверка добавления информации**  После нажатия на кнопку и открытия окна ввести информацию в пустые строки ввода  Нажать 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 №4).

Таблица №6 «Проверка удаления данных»

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

Проверка работоспособности функции «Редактирование данных» проводится следующим образом (таблица №5).

Таблица №5 «Проверка редактирования данных»

|  |  |  |
| --- | --- | --- |
| **№** | **Операции по проверке** | **Результат** |
| 1 | **Подготовка к проверке**  Нажать на кнопку «Редактировать» рядом с записью, которую необходимо отредактировать | Открытие окна редактирование записей |
| 2 | **Проверка удалении информации**  После нажатия на кнопку и открытия окна редактирования записей  Изменить информацию и нажать «Сохранить» | Успешное изменение информации, сохранение и отображение информации в таблице с данными |

**6.4.2** **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.

**6.5 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ**

Дополнительные возможности – отсутствуют.

1. **Руководство пользователя**

**7.1.Назначение программы:**

Система предназначена для организаторов конференций по информационной безопасности

**7.2.Условия выполнения программы:**

Таблица №7 «Минимальные требования»

|  |  |
| --- | --- |
| Процессор | Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz 2.70 GHz |
| 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 2016 (и выше) и Microsoft SQL Management Studio.

**7.3.Выполнение программы:**

При открытии программы (исполняемый «.exe» файл) появится страница авторизации, после авторизации происходит переход в главное меню. Далее по нажатию кнопки можно перейти на различные страницы, такие как «Личный кабинет», «Организаторы», «Жюри», «Модераторы», «Участники», «Активности».

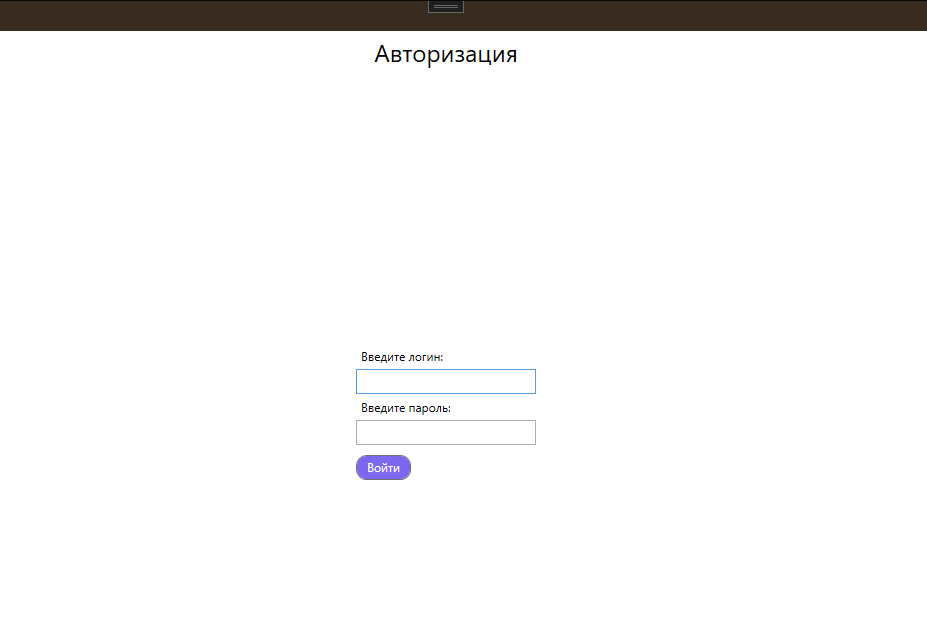


Рис.49 «Страница авторизации»

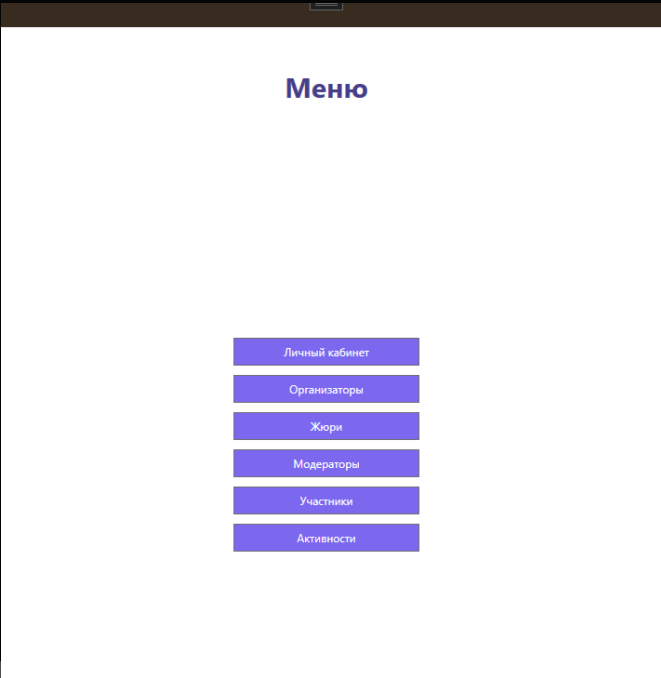


Рис.50 «Главное меню»

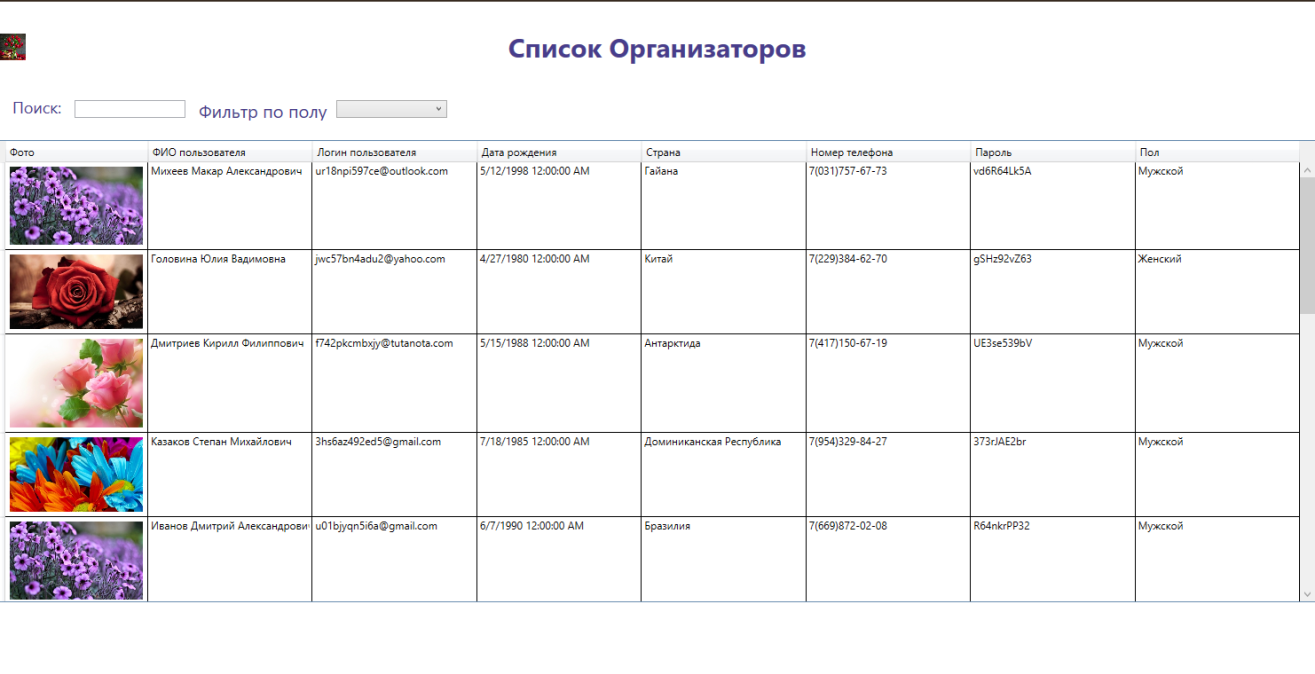


Рис.52 «Страница организаторов»

**Добавление Организатора:**

Для добавления необходимо нажать ПКМ по записи и нажать на кнопку «Добавить» и заполнить следующие поля для таблиц: Организатора – ФИО, Логин, дата рождения, страна, телефон, пароль, пол, фото. Добавление для страниц «Жюри», «Модераторы», «Участники», «Активности», выполняется аналогично.

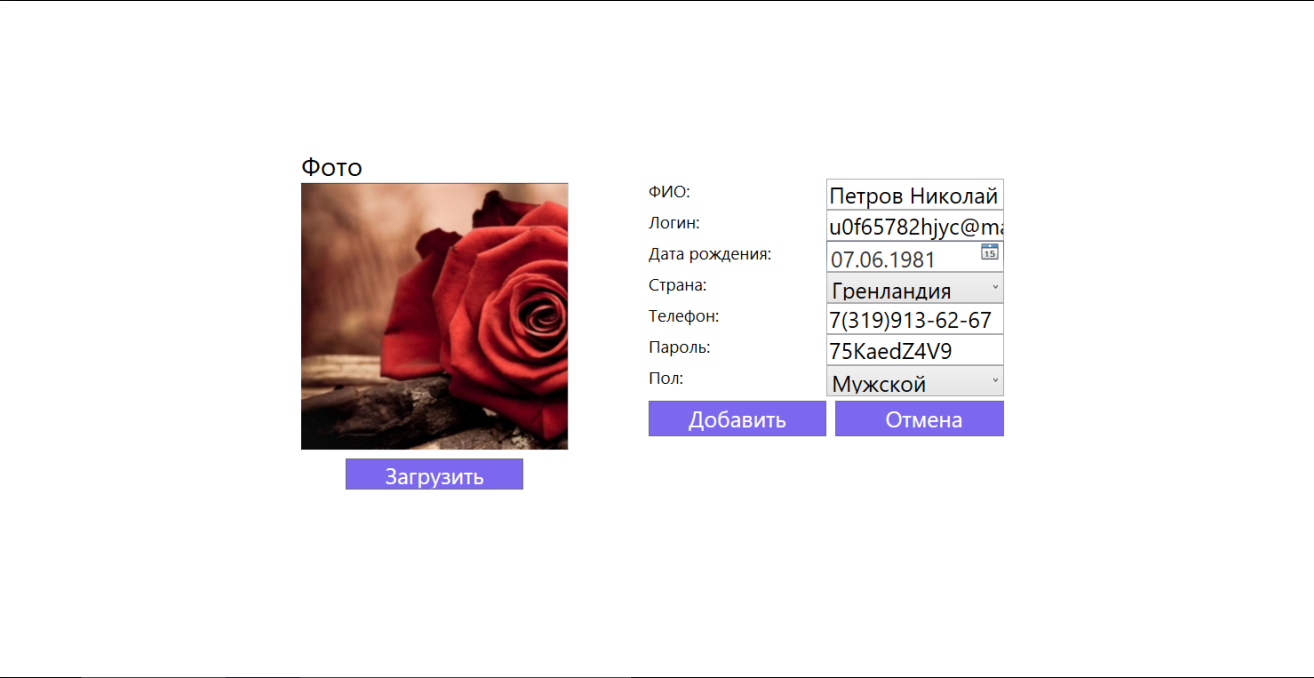


Рис.53 «Добавление новых организаторов»

**Удаление:**

Для удаления записей нужно выбрать запись или несколько записей из списка/таблицы, затем нажать ПКМ и выбрать кнопку с надписью: «Удалить» и подтвердить удаление записи или записей. Удаление для страниц «Жюри», «Модераторы», «Участники», «Активности», выполняется аналогично.

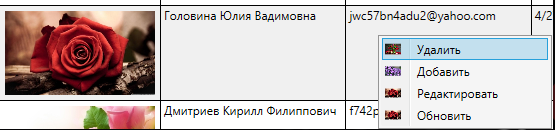


Рис.54 «Удаление записей»

**Фильтры по полу**

Данный фильтр находятся в левой части интерфейса приложения на страницах «Жюри», «Модераторы», «Участники», «Активности», чтобы его применить необходимо нажать на любой из фильтров и в раскрывающимся списке выбрать один из вариантов. Фильтрация для страниц «Жюри», «Модераторы», «Участники», «Активности», выполняется аналогично.

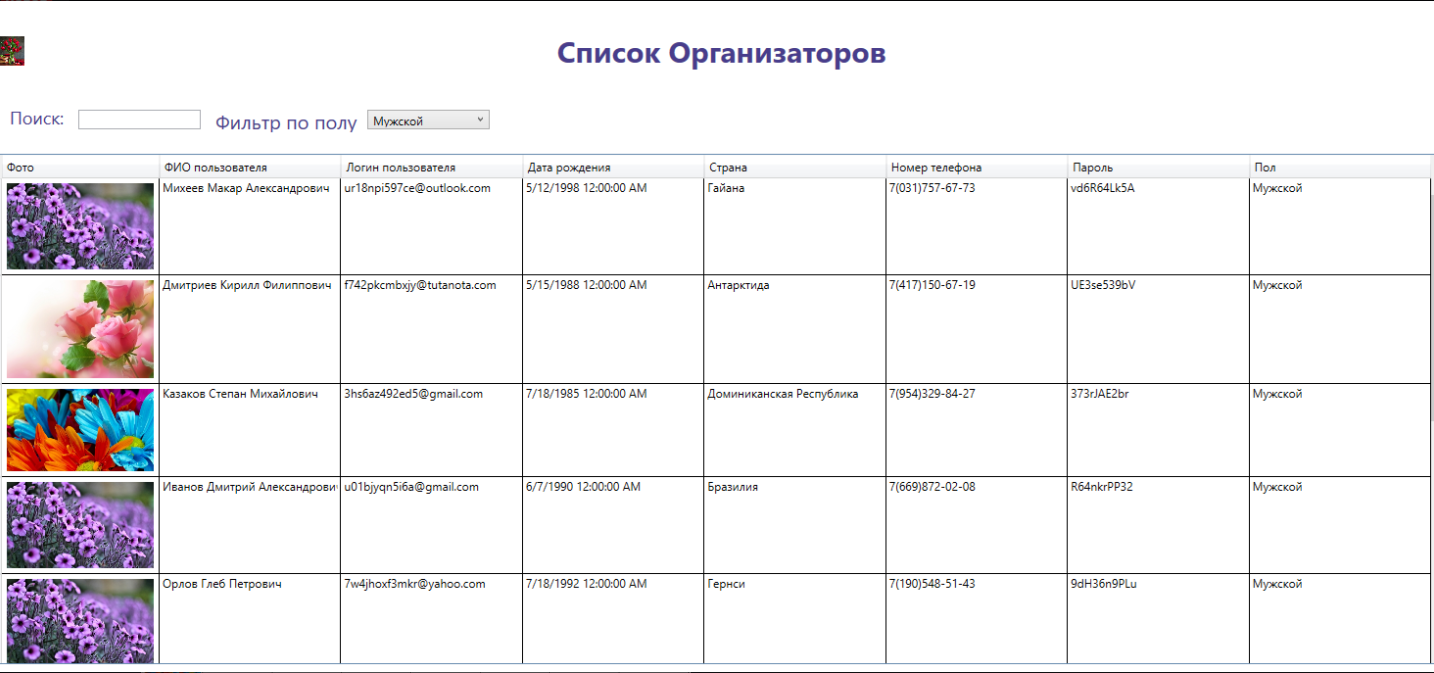


Рис.55 «Фильтр записей»

**Поиск по ФИО**

Данный поиск находится в левой части интерфейса приложения на страницах «Жюри», «Модераторы», «Участники», «Активности», чтобы его применить необходимо нажать на строку поиска и ввести ФИО. Поиск для страниц «Жюри», «Модераторы», «Участники», «Активности», выполняется аналогично.

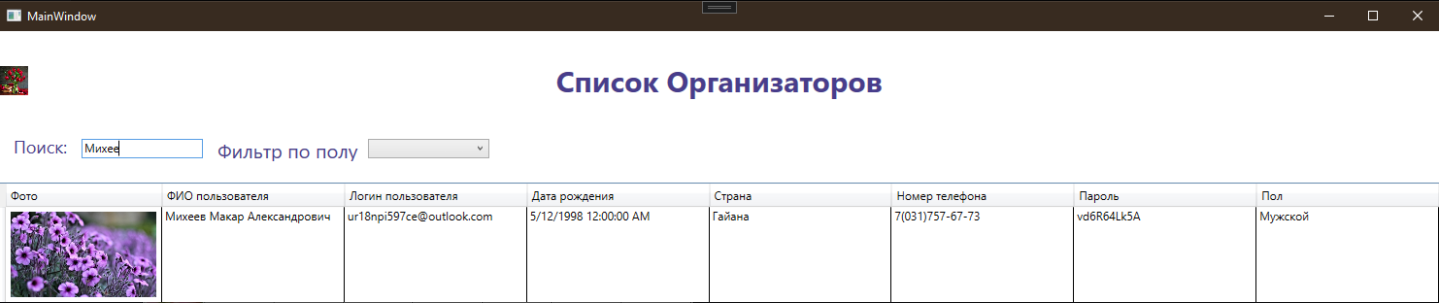


Рис.56 «Поиск по ФИО»

**Поиск по стране**

Данный поиск находится в левой части интерфейса приложения на страницах «Жюри», «Модераторы», «Участники», «Активности», чтобы его применить необходимо нажать на строку поиска и ввести страну. Поиск для страниц «Жюри», «Модераторы», «Участники», «Активности», выполняется аналогично.

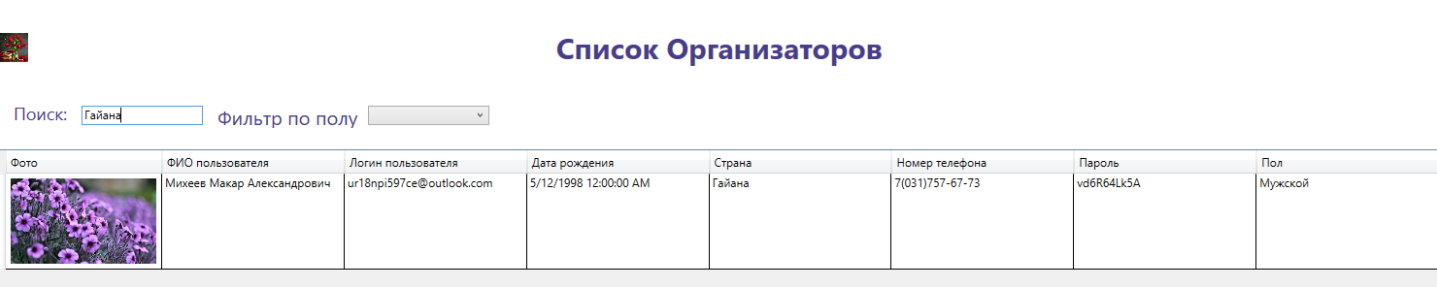


Рис.57 «Поиск по стране»

**Поиск по логину**

Данный поиск находится в левой части интерфейса приложения на страницах «Жюри», «Модераторы», «Участники», «Активности», чтобы его применить необходимо нажать на строку поиска и ввести логин. Поиск для страниц «Жюри», «Модераторы», «Участники», «Активности», выполняется аналогично.

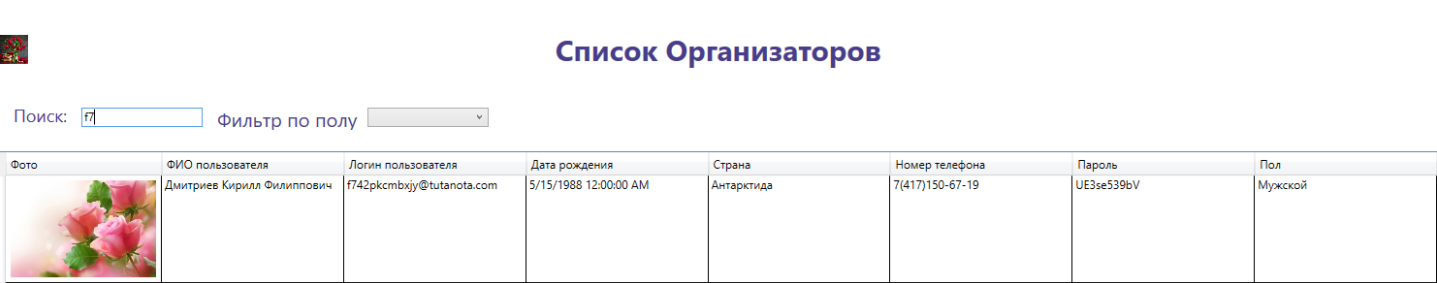


Рис.58 «Поиск по логину»

**Редактирование Организатора:**

Для редактирования необходимо нажать ПКМ по записи и нажать на кнопку «Редактировать» и заполнить следующие поля для таблиц: Организатора – ФИО, Логин, дата рождения, страна, телефон, пароль, пол, фото. Редактирование для страниц «Жюри», «Модераторы», «Участники», «Активности», выполняется аналогично.

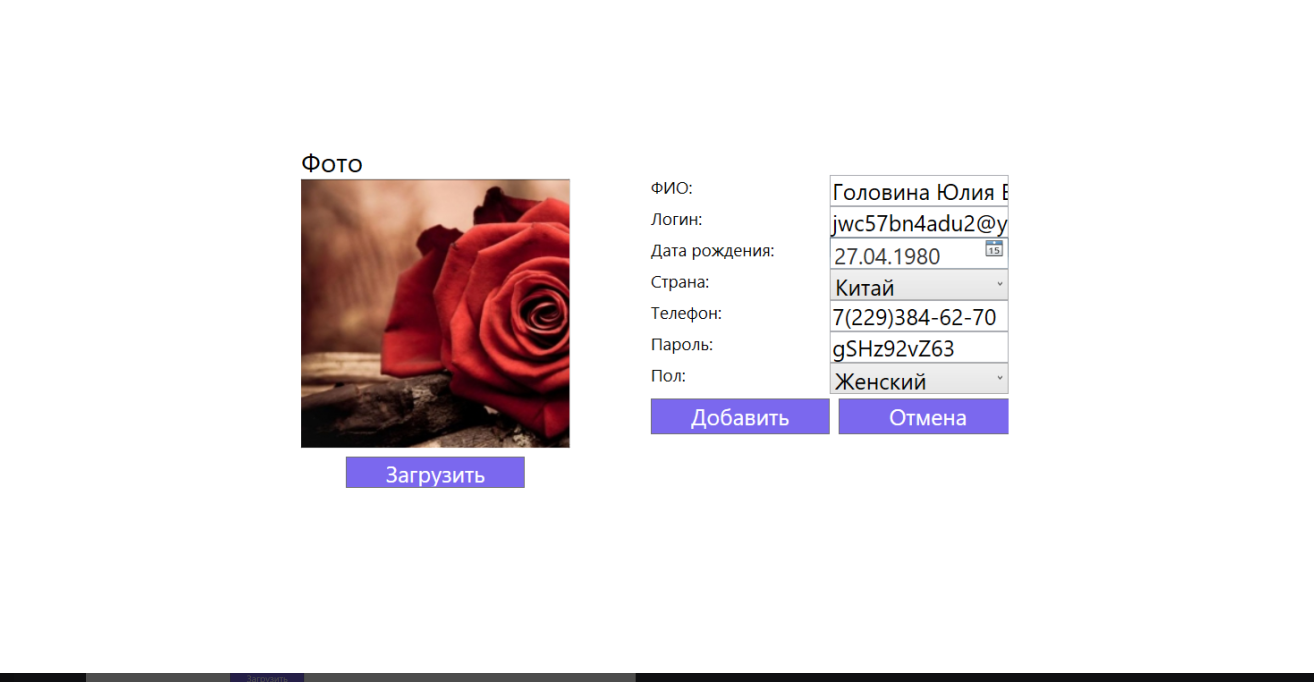


Рис.59 «Редактирование записей»

**Страница Личный кабинет:**

На странице Личный кабинет можно изменить пароль (Нажав ЛКМ), для сохранения пароля нажимаем клавишу Enter, а также выйти из акаунта (Нажав кнопку «Выйти из акаунта»).

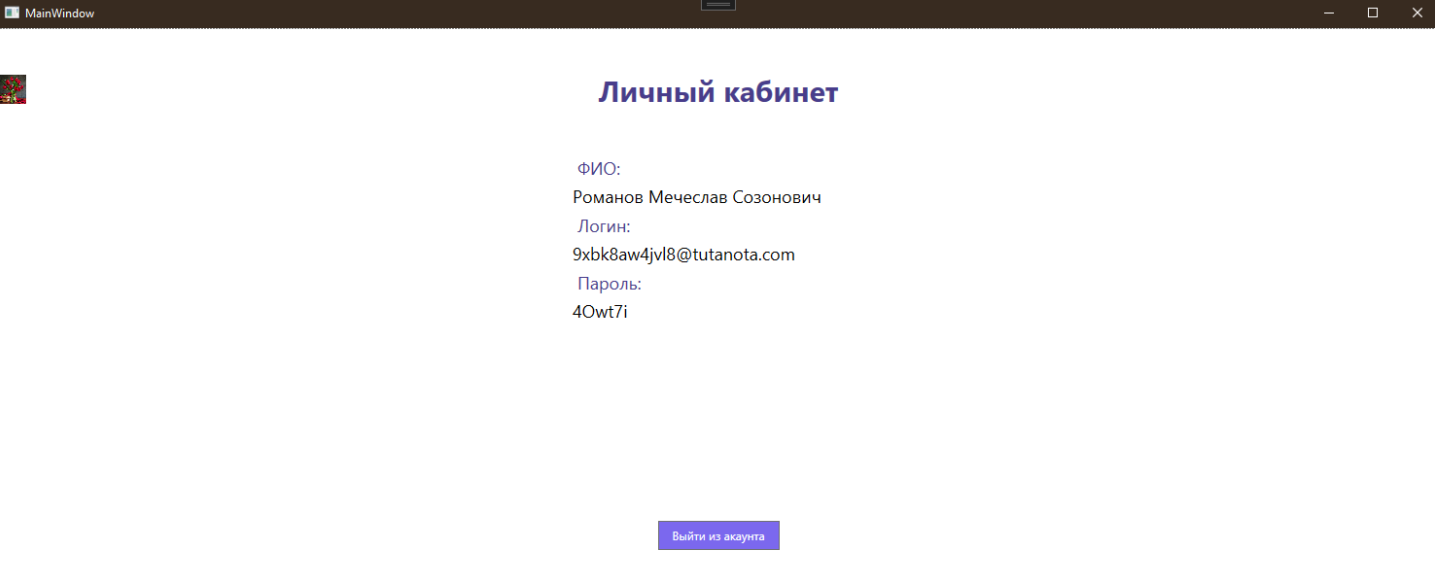


Рис.60 «Страница «Личный кабинет»»

**7.4 Сообщение оператору:**

Попытка добавление записей без данных

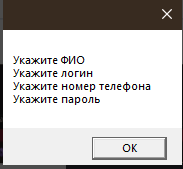


Рис.61 «Попытка добавления данных»

Подтверждение удаления организатора

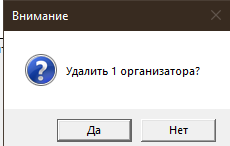


Рис.62 «Подтверждение удаления договора»

Информация после удаления организатора

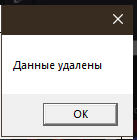


Рис.63 «Информация при удалении договора»

Информация после редактирования организатора

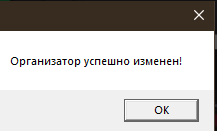


Рис.64 «Информация при редактировании организатора»

Информация после добавления организатора

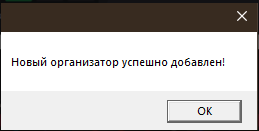


Рис.65 «Информация после добавления организатора»

1. **Отладка и тестирование**

**8.1 Объект испытаний**

**Наименование**

«Конференции по информационной безопасности»

**Область применения**

Система предназначена для организаторов конференций по информационной безопасности.

**Обозначение программы**

«Conference»

**8.2 Цель испытаний**

Цель проведения испытаний - проверка соответствия характеристик разработанной программы (программного продукта) функциональным и отдельным иным видам требований, изложенным в документе Техническое задание.

**8.3 Общие положения**

**Основания для проведения испытаний**

Разработка проводится на основании Договора Между ЛДПК и «Конференции России» № 2 от 10 февраля 2024 г.

**Место и продолжительность испытаний**

Приемосдаточные испытания должны проводиться на объекте заказчика в сроки с 01.12.2023 по 20.01.2024

Приемосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной с заказчиком Программы, и методики испытаний.

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в Протоколе испытаний.

**Перечень документов, предъявляемых на испытания**

Состав программной документации должен включать в себя:

1. Техническое задание;

2. Программу и методику испытаний;

3. Руководство системного программиста;

4. Руководство пользователя;

5. Руководство программиста;

6. Текст программы;

**8.4 Объем испытаний**

**Перечень этапов испытаний**

1. **Ознакомительный –** проверка комплектности программной документации, техническое задание, программа и методика испытаний, руководство программиста, ведомость эксплуатационных документов.
2. **Испытания -** проверка комплектности программной документации, техническое задание, программа и методика испытаний, руководство программиста, руководство оператора, руководство системного программиста, ведомость эксплуатационных документов.

**Количественные характеристики, подлежащие оценке**

В ходе проведения приемосдаточных испытаний оценке подлежат количественные характеристики, такие как:

1. комплектность программной документации;
2. комплектность состава технических и программных средств.

**Качественные характеристики, подлежащие оценке**

В ходе проведения приемосдаточных испытаний оценке подлежат качественные (функциональные) характеристики программы. Проверке подлежит возможность выполнения программой перечисленных ниже функций:

Функциональные возможности модуля:

* Авторизация для:
* Участников
* Организаторов
* Жури
* Модераторов
* Добавление запесей в таблицах:
* Участников
* Организаторов
* Жури
* Модераторов
* Активности
* Удаление запесей в таблицах:
* Участников
* Организаторов
* Жури
* Модераторов
* Активности
* Редактирование запесей в таблицах:
* Участников
* Организаторов
* Жури
* Модераторов
* Активности
* Поиск записей по:
* ФИО
* Название страны
* Фильтрация записей по:
* Полу
* Импорт данных Excel в БД Microsoft SQL Management Studio

**Последовательность проведения и режимы испытаний**

Испытания проводятся в последовательности, указанной в п. «Перечень этапов испытаний».

**Перечень работ, проводимых после завершения испытаний**

В случае успешного проведения испытаний в полном объеме исполнитель совместно с заказчиком на основании Протокола испытаний утверждают Акт приемки-сдачи работ.

Исполнитель передает заказчику программное изделие, программную (эксплуатационную) документацию и т.д.

В случае выявления несоответствия разработанной программы отдельным требованиям технического задания исполнитель проводит корректировку программы и программной документации по результатам испытаний в сроки, согласованные с заказчиком.

По завершении корректировки программы и программной документации исполнитель и заказчик проводят повторные испытания согласно настоящей программы и методик в объеме, требуемом для проверки проведения корректировок.

**Порядок проведения испытаний**

Испытания должны проводиться поэтапно согласно п. «Перечень этапов испытаний» настоящего документа.

**8.5 Средства и порядок испытаний**

**Технические средства, использованные во время испытаний:**

Системные требования для работы программного продукта должны быть следующими:

Таблица №8 «Минимальные требования»

|  |  |
| --- | --- |
| Процессор | Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz 2.70 GHz |
| 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 |

**Программные средства, используемые во время испытаний:**

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

**Условия проведения испытаний**

Испытания должны проводиться в нормальных климатических условиях по ГОСТ 22261-94. Условия проведения испытаний приведены ниже:

* температура окружающего воздуха, °С - 20 ± 5;
* относительная влажность, % - от 30 до 80;
* атмосферное давление, кПа - от 84 до 106;
* частота питающей электросети, Гц - 50 ± 0,5;
* напряжение питающей сети переменного тока, В - 220 ± 4,4.

**Условия начала и завершения отдельных этапов испытаний**

Необходимым и достаточным условием завершения 1 этапа испытаний и начала 2 этапа испытаний является успешное завершение проверок, проводимых на 1 этапе (см. п. «Перечень проверок, проводимых на 1 этапе испытаний»).

Условием завершения 2 этапа испытаний является успешное завершение проверок, проводимых на 2 этапе испытаний.

**Ограничения в условиях проведения испытаний**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

**Требования к техническому обслуживанию**

Требования к техническому обслуживанию не предъявляются.

**Меры, обеспечивающие безопасность и безаварийность проведения испытаний**

При проведении испытаний заказчик должен обеспечить соблюдение требований безопасности.

**Порядок взаимодействия организаций, участвующих в испытаниях**

Исполнитель письменно извещает заказчика о готовности к проведению приемо-сдаточных испытаний не позднее чем за 14 дней до намеченного срока проведения испытаний.

Заказчик Договором назначает срок проведения испытаний и приемочную комиссии, которая должна включать в свой состав представителей заказчика и исполнителя.

Заказчик письменно извещает сторонние организации, которые должны принять участие в приемо-сдаточных испытаниях.

Заказчик совместно с исполнителем проводят все подготовительные мероприятия для проведения испытаний на объекте заказчика, а также проводят испытания в соответствии с настоящей программой и методиками.

Заказчик осуществляет контроль проведения испытаний, а также документирует ход проведения проверок в Протоколе проведения испытаний.

**Требования к персоналу, проводящему испытания**

Персонал, проводящий испытания, должен иметь квалификацию, соответствующую профессиональной деятельности.

**Приложения**

Таблица 9 «Тестовый сценарий 1»

|  |  |
| --- | --- |
| **Тестовый пример #** | ТС\_ПИ\_1. Добавление организатора через страницу добавления |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Добавление |
| **Краткое изложение теста** | Должно производится добавление организаторов через страницу добавления без ошибок |
| **Этапы теста** | Запуск приложения  Пройти авторизацию  В главном меню переходим на страницу «Организаторы»  Нажимаем «ПКМ» и кнопку «Добавить» |
| **Тестовые данные** | ФИО – Сергеев Антон Витальевич, Логин - asd@gmail.com, Дата рождения – 09.11.2000, Страна – Абхазия, Телефон – 89999999999, Пароль - 8976Nnn, Пол – Мужской. |
| **Ожидаемый результат** | Новый организатор успешно добавлен! |
| **Фактический результат** |  |
| **Статус** | Зачёт |
| **Предварительное условие** | Перед проверкой теста нужно запустить программу и нажать на кнопку «Добавление» |
| **Постусловие** | После выполнения теста нажать на кнопку Сохранить. |
| **Примечания/комментарии** | Данный тест пройден успешно. |

Таблица 10 «Тестовый сценарий 2»

|  |  |
| --- | --- |
| **Тестовый пример #** | ТС\_ПИ\_1. Поиск по стране |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Поиск |
| **Краткое изложение теста** | Должен производиться поиск по стране |
| **Этапы теста** | Запуск приложения  Пройти авторизацию  В главном меню переходим на страницу «Организаторы»  В поле вводим необходимую страну |
| **Тестовые данные** | Страна - Абхазия |
| **Ожидаемый результат** | Список данных с указанной страной |
| **Фактический результат** |  |
| **Статус** | Зачёт |
| **Предварительное условие** | Перед проверкой теста нужно запустить программу и заполнить данными поисковую строку |
| **Постусловие** | После выполнения теста нажать на кнопку Сохранить. |
| **Примечания/комментарии** | Данный тест пройден успешно. |

Таблица 11 «Тестовый сценарий 3»

|  |  |
| --- | --- |
| **Тестовый пример #** | ТС\_ПИ\_1. Поиск по стране |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Редактирование |
| **Краткое изложение теста** | Должно производиться редактирование записи |
| **Этапы теста** | Запуск приложения  Пройти авторизацию  В главном меню переходим на страницу «Организаторы»  Выбираем необходимую запись и нажимаем кнопку «Редактировать» |
| **Тестовые данные** | ФИО – Сергеев Антон Витальевич, Логин - asd@gmail.com, Дата рождения – 09.11.2000, Страна – Китай, Телефон – 89999999999, Пароль - 8976Nnn, Пол – Мужской. |
| **Ожидаемый результат** | Организатор успешно изменен! |
| **Фактический результат** |  |
| **Статус** | Зачёт |
| **Предварительное условие** | Перед проверкой теста нужно запустить программу и выбрать данные для редактирования |
| **Постусловие** | После выполнения теста нажать на кнопку Сохранить. |
| **Примечания/комментарии** | Данный тест пройден успешно. |

Отладка — этап разработки компьютерной программы, на котором обнаруживают, локализуют и устраняют ошибки

Таблица 12 «Отладка»

|  |  |  |
| --- | --- | --- |
| 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.**  Отсутствует точка с запятой. |  | Требуется дописать точку с запятой. |
| **Отсутствие ссылок на дополнительные пакеты.**  Отсутствует ссылка на объект данных |  | Добавить ссылку на объект данных |
| **Ошибка с типом данных**  Неверный тип данных у атрибута |  | Исправить тип данных |
| **Ошибка при обновлении базы данных**  При обновлении базы данных пропал Get Context |  | Прописать Get Context |

1. **Текст программы**

**Код CS для страницы «Авторизация»**

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

{

Participants participants = ClassFrame.db.Participants.FirstOrDefault(x => x.Email == login.Text && x.Pasword == password.Password);

Moderators moderators = ClassFrame.db.Moderators.FirstOrDefault(x => x.Email == login.Text && x.Pasword == password.Password);

Jury jury = ClassFrame.db.Jury.FirstOrDefault(x => x.Email == login.Text && x.Pasword == password.Password);

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

if (participants != null)

{

Users \_user = new Users(participants);

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

}

else if (moderators != null)

{

Users \_user = new Users(moderators);

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

}

else if (jury != null)

{

Users \_user = new Users(jury);

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

}

else if(organizers != null)

{

Users \_user = new Users(organizers);

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

}

if (participants == null && organizers == null && jury == null && moderators == null)

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

}

}

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

{

//ClassFrame.frmObj.Navigate(new PageRegistration());

}

}

**Код CS для страницы «Главного меню»**

public Menu()

{

InitializeComponent();

}

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

{

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

}

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

{

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

}

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

{

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

}

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

{

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

}

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

{

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

}

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

{

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

}

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

{

//ClassFrame.frmObj.Navigate(new PageRegistration());

}

}

**Код CS для страницы «Личный кабинет»**

Public partial class PersonalAccount : Page

{

Participants \_currentUser = ClassFrame.db.Participants.FirstOrDefault(x => x.IDParticipants == Users.GetUsers.ID);

Jury \_currentUser1 = ClassFrame.db.Jury.FirstOrDefault(x => x.IDJury == Users.GetUsers.ID);

Organizers \_currentUser2 = ClassFrame.db.Organizers.FirstOrDefault(x => x.IDOrganizers == Users.GetUsers.ID);

Moderators \_currentUser3 = ClassFrame.db.Moderators.FirstOrDefault(x => x.IDModerators == Users.GetUsers.ID);

public PersonalAccount()

{

InitializeComponent();

DataContext = Users.GetUsers;

}

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

{

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

}

private void TextBox\_KeyUp(object sender, KeyEventArgs e)

{

if (e.Key == Key.Enter)

{

TextBox textBox = e.Source as TextBox;

if (textBox.Name == "txtFIO1")

{

Users.GetUsers.FIO = textBox.Text;

txtFIO1.Visibility = Visibility.Hidden;

\_currentUser.FIO = textBox.Text;

}

if (textBox.Name == "txtLogin1")

{

Users.GetUsers.Email = textBox.Text;

txtLogin1.Visibility = Visibility.Hidden;

\_currentUser.Email = textBox.Text;

}

if (textBox.Name == "txtPasword1")

{

Users.GetUsers.Pasword = textBox.Text;

txtPasword1.Visibility = Visibility.Hidden;

\_currentUser.Pasword = textBox.Text;

}

ClassFrame.db.SaveChanges();

}

if (e.Key == Key.Enter)

{

TextBox textBox = e.Source as TextBox;

if (textBox.Name == "txtFIO1")

{

Users.GetUsers.FIO = textBox.Text;

txtFIO1.Visibility = Visibility.Hidden;

\_currentUser2.FIO = textBox.Text;

}

if (textBox.Name == "txtLogin1")

{

Users.GetUsers.Email = textBox.Text;

txtLogin1.Visibility = Visibility.Hidden;

\_currentUser2.Email = textBox.Text;

}

if (textBox.Name == "txtPasword1")

{

Users.GetUsers.Pasword = textBox.Text;

txtPasword1.Visibility = Visibility.Hidden;

\_currentUser2.Pasword = textBox.Text;

}

ClassFrame.db.SaveChanges();

}

else if (e.Key == Key.Enter)

{

TextBox textBox = e.Source as TextBox;

if (textBox.Name == "txtFIO1")

{

Users.GetUsers.FIO = textBox.Text;

txtFIO1.Visibility = Visibility.Hidden;

\_currentUser3.FIO = textBox.Text;

}

if (textBox.Name == "txtLogin1")

{

Users.GetUsers.Email = textBox.Text;

txtLogin1.Visibility = Visibility.Hidden;

\_currentUser3.Email = textBox.Text;

}

if (textBox.Name == "txtPasword1")

{

Users.GetUsers.Pasword = textBox.Text;

txtPasword1.Visibility = Visibility.Hidden;

\_currentUser3.Pasword = textBox.Text;

}

ClassFrame.db.SaveChanges();

}

else if (e.Key == Key.LeftCtrl)

{

TextBox textBox = e.Source as TextBox;

if (textBox.Name == "txtFIO1")

{

textBox.Text = \_currentUser.FIO;

txtFIO1.Visibility = Visibility.Hidden;

}

if (textBox.Name == "txtLogin1")

{

textBox.Text = \_currentUser.Email;

txtLogin1.Visibility = Visibility.Hidden;

}

if (textBox.Name == "txtPasword1")

{

textBox.Text = \_currentUser.Pasword;

txtPasword1.Visibility = Visibility.Hidden;

}

}

else if (e.Key == Key.LeftCtrl)

{

TextBox textBox = e.Source as TextBox;

if (textBox.Name == "txtFIO1")

{

textBox.Text = \_currentUser1.FIO;

txtFIO1.Visibility = Visibility.Hidden;

}

if (textBox.Name == "txtLogin1")

{

textBox.Text = \_currentUser1.Email;

txtLogin1.Visibility = Visibility.Hidden;

}

if (textBox.Name == "txtPasword1")

{

textBox.Text = \_currentUser1.Pasword;

txtPasword1.Visibility = Visibility.Hidden;

}

}

else if (e.Key == Key.LeftCtrl)

{

TextBox textBox = e.Source as TextBox;

if (textBox.Name == "txtFIO1")

{

textBox.Text = \_currentUser2.FIO;

txtFIO1.Visibility = Visibility.Hidden;

}

if (textBox.Name == "txtLogin1")

{

textBox.Text = \_currentUser2.Email;

txtLogin1.Visibility = Visibility.Hidden;

}

if (textBox.Name == "txtPasword1")

{

textBox.Text = \_currentUser2.Pasword;

txtPasword1.Visibility = Visibility.Hidden;

}

}

else if (e.Key == Key.LeftCtrl)

{

TextBox textBox = e.Source as TextBox;

if (textBox.Name == "txtFIO1")

{

textBox.Text = \_currentUser3.FIO;

txtFIO1.Visibility = Visibility.Hidden;

}

if (textBox.Name == "txtLogin1")

{

textBox.Text = \_currentUser3.Email;

txtLogin1.Visibility = Visibility.Hidden;

}

if (textBox.Name == "txtPasword1")

{

textBox.Text = \_currentUser3.Pasword;

txtPasword1.Visibility = Visibility.Hidden;

}

}

}

private void TextBlock\_MouseDown(object sender, MouseButtonEventArgs e)

{

if (e.ClickCount == 2)

{

TextBlock textBlock = sender as TextBlock;

if (textBlock.Name == "txtFIO")

{

txtFIO1.Visibility = Visibility.Visible;

txtFIO1.CaretIndex = txtFIO1.Text.Length;

txtFIO1.Focus();

}

if (textBlock.Name == "txtLogin")

{

txtLogin1.Visibility = Visibility.Visible;

txtLogin1.CaretIndex = txtLogin1.Text.Length;

txtLogin1.Focus();

}

if (textBlock.Name == "txtPasword")

{

txtPasword1.Visibility = Visibility.Visible;

txtPasword1.CaretIndex = txtPasword1.Text.Length;

txtPasword1.Focus();

}

}

}

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

{

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

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

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

}

**Код CS для страницы «Организаторы»**

public partial class OrganizersList : Page

{

public OrganizersList()

{

InitializeComponent();

foreach (var item in dtgOrganizers.Items)

MessageBox.Show(item.ToString());

dtgOrganizers.Items.Clear();

dtgOrganizers.ItemsSource = ConferenceEntities.GetContext().Organizers.ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

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

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (txtSearch.Text.Count() != 0)

dtgOrganizers.ItemsSource = ClassFrame.db.Organizers.Where(x => x.FIO.Contains(txtSearch.Text.ToLower()) || x.Country.NameCountry.Contains(txtSearch.Text.ToLower()) || x.Email.Contains(txtSearch.Text.ToLower())).ToList();

else dtgOrganizers.ItemsSource = ClassFrame.db.Organizers.ToList();

}

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

{

var usersForRemoving = dtgOrganizers.SelectedItems.Cast<Organizers>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} организатора?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

ConferenceEntities.GetContext().Organizers.RemoveRange(usersForRemoving);

ConferenceEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

dtgOrganizers.ItemsSource = ConferenceEntities.GetContext().Organizers.ToList();

}

catch (Exception ex)

{

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

}

}

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

{

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

}

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

{

ClassFrame.frmObj.Navigate(new AddOrganizers((Organizers)dtgOrganizers.SelectedItem));

}

private void Menu1\_Click(object sender, RoutedEventArgs e)

{

dtgOrganizers.ItemsSource = ConferenceEntities.GetContext().Organizers.ToList();

CmbFiltr.SelectedIndex = -1;

}

private void CmbFiltr\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (CmbFiltr.SelectedIndex == 0)

{

dtgOrganizers.ItemsSource = ConferenceEntities.GetContext().Organizers.Where(x => x.IDPol == 1).ToList();

}

else

if (CmbFiltr.SelectedIndex == 1)

{

dtgOrganizers.ItemsSource = ConferenceEntities.GetContext().Organizers.Where(x => x.IDPol == 2).ToList();

}

}

**Код CS для добавления организаторов**

public partial class AddOrganizers : Page

{

private Organizers \_currentOrganizers = new Organizers();

string imgLoc = "пусто";

public AddOrganizers(Organizers selectedOrganizers)

{

InitializeComponent();

CMBPol.ItemsSource = ConferenceEntities.GetContext().Pol.ToList();

CMBPol.SelectedValuePath = "IDPol";

CMBPol.DisplayMemberPath = "NamePol";

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

CMBCountry.SelectedValuePath = "IDCountry";

CMBCountry.DisplayMemberPath = "NameCountry";

if (selectedOrganizers != null)

{

\_currentOrganizers = selectedOrganizers;

}

DataContext = \_currentOrganizers;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentOrganizers.FIO)) error.AppendLine("Укажите ФИО");

if (string.IsNullOrWhiteSpace(\_currentOrganizers.Email)) error.AppendLine("Укажите логин");

if (string.IsNullOrWhiteSpace(\_currentOrganizers.DateOfBirth.ToString())) error.AppendLine("Укажите дату рождения");

if (string.IsNullOrWhiteSpace(\_currentOrganizers.Phone)) error.AppendLine("Укажите номер телефона");

if (string.IsNullOrWhiteSpace(\_currentOrganizers.Pasword)) error.AppendLine("Укажите пароль");

if (string.IsNullOrWhiteSpace(\_currentOrganizers.IDPol.ToString())) error.AppendLine("Укажите пол");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentOrganizers.IDOrganizers == 0)

{

ConferenceEntities.GetContext().Organizers.Add(\_currentOrganizers);

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

string filename = imgLoc.Substring(imgLoc.LastIndexOf('\\') + 1);

\_currentOrganizers.Photo = String.Concat("/Photo/", filename);

}

if (imgLoc == "пусто") \_currentOrganizers.Photo = null;

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new OrganizersList());

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

}

catch (Exception ex)

{

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

}

}

else

{

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

\_currentOrganizers.Photo = imgLoc;

}

if (imgLoc == "пусто") \_currentOrganizers.Photo = null;

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new OrganizersList());

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

}

catch (Exception ex)

{

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

}

}

}

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

{

Classes.ClassFrame.frmObj.Navigate(new OrganizersList());

}

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

{

try

{

OpenFileDialog dlg = new OpenFileDialog

{

Filter = "JPG Files (\*.jpg)|\*.jpg|PNG Files (\*.png)|\*.png|GIF Files (\*.gif)|\*.gif|All Files (\*.\*)|\*.\*",

Title = "Выберите фото/изображение организатора"

};

if (dlg.ShowDialog() == true)

**Код CS для страницы «Жюри»**

public partial class JuryList : Page

{

public JuryList()

{

InitializeComponent();

foreach (var item in dtgJury.Items)

MessageBox.Show(item.ToString());

dtgJury.Items.Clear();

dtgJury.ItemsSource = ConferenceEntities.GetContext().Jury.ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

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

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (txtSearch.Text.Count() != 0)

dtgJury.ItemsSource = ClassFrame.db.Jury.Where(x => x.FIO.Contains(txtSearch.Text.ToLower()) || x.Country.NameCountry.Contains(txtSearch.Text.ToLower()) || x.Email.Contains(txtSearch.Text.ToLower())).ToList();

else dtgJury.ItemsSource = ClassFrame.db.Jury.ToList();

}

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

{

var usersForRemoving = dtgJury.SelectedItems.Cast<Jury>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} жюри?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

ConferenceEntities.GetContext().Jury.RemoveRange(usersForRemoving);

ConferenceEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

dtgJury.ItemsSource = ConferenceEntities.GetContext().Jury.ToList();

}

catch (Exception ex)

{

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

}

}

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

{

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

}

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

{

ClassFrame.frmObj.Navigate(new AddJury((Jury)dtgJury.SelectedItem));

}

private void Menu1\_Click(object sender, RoutedEventArgs e)

{

dtgJury.ItemsSource = ConferenceEntities.GetContext().Jury.ToList();

CmbFiltr.SelectedIndex = -1;

}

private void CmbFiltr\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (CmbFiltr.SelectedIndex == 0)

{

dtgJury.ItemsSource = ConferenceEntities.GetContext().Jury.Where(x => x.IDPol == 1).ToList();

}

else

if (CmbFiltr.SelectedIndex == 1)

{

dtgJury.ItemsSource = ConferenceEntities.GetContext().Jury.Where(x => x.IDPol == 2).ToList();

}

}

}

**Код CS для добавления жюри**

public partial class AddJury : Page

{

private Jury \_currentJury = new Jury();

string imgLoc = "пусто";

public AddJury(Jury selectedJury)

{

InitializeComponent();

CMBPol.ItemsSource = ConferenceEntities.GetContext().Pol.ToList();

CMBPol.SelectedValuePath = "IDPol";

CMBPol.DisplayMemberPath = "NamePol";

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

CMBCountry.SelectedValuePath = "IDCountry";

CMBCountry.DisplayMemberPath = "NameCountry";

CMBDirection.ItemsSource = ConferenceEntities.GetContext().Direction.ToList();

CMBDirection.SelectedValuePath = "IDDirection";

CMBDirection.DisplayMemberPath = "NameDirection";

if (selectedJury != null)

{

\_currentJury = selectedJury;

}

DataContext = \_currentJury;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentJury.FIO)) error.AppendLine("Укажите ФИО");

if (string.IsNullOrWhiteSpace(\_currentJury.Email)) error.AppendLine("Укажите логин");

if (string.IsNullOrWhiteSpace(\_currentJury.DateOfBirth.ToString())) error.AppendLine("Укажите дату рождения");

if (string.IsNullOrWhiteSpace(\_currentJury.Phone)) error.AppendLine("Укажите номер телефона");

if (string.IsNullOrWhiteSpace(\_currentJury.Pasword)) error.AppendLine("Укажите пароль");

if (string.IsNullOrWhiteSpace(\_currentJury.IDPol.ToString())) error.AppendLine("Укажите пол");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentJury.IDJury == 0)

{

ConferenceEntities.GetContext().Jury.Add(\_currentJury);

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

string filename = imgLoc.Substring(imgLoc.LastIndexOf('\\') + 1);

\_currentJury.Photo = String.Concat("/Photo/", filename);

}

if (imgLoc == "пусто") \_currentJury.Photo = null;

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new JuryList());

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

}

catch (Exception ex)

{

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

}

}

else

{

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

\_currentJury.Photo = imgLoc;

}

if (imgLoc == "пусто") \_currentJury.Photo = null;

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new JuryList());

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

}

catch (Exception ex)

{

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

}

}

}

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

{

Classes.ClassFrame.frmObj.Navigate(new JuryList());

}

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

{

try

{

OpenFileDialog dlg = new OpenFileDialog

{

Filter = "JPG Files (\*.jpg)|\*.jpg|PNG Files (\*.png)|\*.png|GIF Files (\*.gif)|\*.gif|All Files (\*.\*)|\*.\*",

Title = "Выберите фото/изображение организатора"

};

if (dlg.ShowDialog() == true)

{

imgLoc = dlg.FileName.ToString();

imageJury.Source = new BitmapImage(new Uri(imgLoc));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

**Код CS для страницы «Модераторы»**

public partial class ModeratorsList : Page

{

public ModeratorsList()

{

InitializeComponent();

foreach (var item in dtgModerators.Items)

MessageBox.Show(item.ToString());

dtgModerators.Items.Clear();

dtgModerators.ItemsSource = ConferenceEntities.GetContext().Moderators.ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

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

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (txtSearch.Text.Count() != 0)

dtgModerators.ItemsSource = ClassFrame.db.Moderators.Where(x => x.FIO.Contains(txtSearch.Text.ToLower()) || x.Country.NameCountry.Contains(txtSearch.Text.ToLower()) || x.Email.Contains(txtSearch.Text.ToLower())).ToList();

else dtgModerators.ItemsSource = ClassFrame.db.Moderators.ToList();

}

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

{

var usersForRemoving = dtgModerators.SelectedItems.Cast<Moderators>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} модератора?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

ConferenceEntities.GetContext().Moderators.RemoveRange(usersForRemoving);

ConferenceEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

dtgModerators.ItemsSource = ConferenceEntities.GetContext().Moderators.ToList();

}

catch (Exception ex)

{

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

}

}

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

{

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

}

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

{

ClassFrame.frmObj.Navigate(new AddModerators((Moderators)dtgModerators.SelectedItem));

}

private void Menu1\_Click(object sender, RoutedEventArgs e)

{

dtgModerators.ItemsSource = ConferenceEntities.GetContext().Moderators.ToList();

CmbFiltr.SelectedIndex = -1;

}

private void CmbFiltr\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (CmbFiltr.SelectedIndex == 0)

{

dtgModerators.ItemsSource = ConferenceEntities.GetContext().Moderators.Where(x => x.IDPol == 1).ToList();

}

else

if (CmbFiltr.SelectedIndex == 1)

{

dtgModerators.ItemsSource = ConferenceEntities.GetContext().Moderators.Where(x => x.IDPol == 2).ToList();

}

}

}

**Код CS для добавления модераторы**

public partial class AddModerators : Page

{

private Moderators \_currentModerators = new Moderators();

string imgLoc = "пусто";

public AddModerators(Moderators selectedModerators)

{

InitializeComponent();

CMBPol.ItemsSource = ConferenceEntities.GetContext().Pol.ToList();

CMBPol.SelectedValuePath = "IDPol";

CMBPol.DisplayMemberPath = "NamePol";

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

CMBCountry.SelectedValuePath = "IDCountry";

CMBCountry.DisplayMemberPath = "NameCountry";

CMBDirection.ItemsSource = ConferenceEntities.GetContext().Direction.ToList();

CMBDirection.SelectedValuePath = "IDDirection";

CMBDirection.DisplayMemberPath = "NameDirection";

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

CMBEvent.SelectedValuePath = "IDEvent";

CMBEvent.DisplayMemberPath = "NameEvent";

if (selectedModerators != null)

{

\_currentModerators = selectedModerators;

}

DataContext = \_currentModerators;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentModerators.FIO)) error.AppendLine("Укажите ФИО");

if (string.IsNullOrWhiteSpace(\_currentModerators.Email)) error.AppendLine("Укажите логин");

if (string.IsNullOrWhiteSpace(\_currentModerators.DateOfBirth.ToString())) error.AppendLine("Укажите дату рождения");

if (string.IsNullOrWhiteSpace(\_currentModerators.Phone)) error.AppendLine("Укажите номер телефона");

if (string.IsNullOrWhiteSpace(\_currentModerators.Pasword)) error.AppendLine("Укажите пароль");

if (string.IsNullOrWhiteSpace(\_currentModerators.IDPol.ToString())) error.AppendLine("Укажите пол");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentModerators.IDModerators == 0)

{

ConferenceEntities.GetContext().Moderators.Add(\_currentModerators);

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

string filename = imgLoc.Substring(imgLoc.LastIndexOf('\\') + 1);

\_currentModerators.Photo = String.Concat("/Photo/", filename);

}

if (imgLoc == "пусто") \_currentModerators.Photo = null;

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new ModeratorsList());

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

}

catch (Exception ex)

{

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

}

}

else

{

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

\_currentModerators.Photo = imgLoc;

}

if (imgLoc == "пусто") \_currentModerators.Photo = null;

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new ModeratorsList());

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

}

catch (Exception ex)

{

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

}

}

}

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

{

Classes.ClassFrame.frmObj.Navigate(new ModeratorsList());

}

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

{

try

{

OpenFileDialog dlg = new OpenFileDialog

{

Filter = "JPG Files (\*.jpg)|\*.jpg|PNG Files (\*.png)|\*.png|GIF Files (\*.gif)|\*.gif|All Files (\*.\*)|\*.\*",

Title = "Выберите фото/изображение организатора"

};

if (dlg.ShowDialog() == true)

{

imgLoc = dlg.FileName.ToString();

imageModerators.Source = new BitmapImage(new Uri(imgLoc));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

**Код CS для страницы «Участники»**

public partial class ParticipantsList : Page

{

public ParticipantsList()

{

InitializeComponent();

foreach (var item in dtgParticipants.Items)

MessageBox.Show(item.ToString());

dtgParticipants.Items.Clear();

dtgParticipants.ItemsSource = ConferenceEntities.GetContext().Participants.ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

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

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (txtSearch.Text.Count() != 0)

dtgParticipants.ItemsSource = ClassFrame.db.Participants.Where(x => x.FIO.Contains(txtSearch.Text.ToLower()) || x.Country.NameCountry.Contains(txtSearch.Text.ToLower()) || x.Email.Contains(txtSearch.Text.ToLower())).ToList();

else dtgParticipants.ItemsSource = ClassFrame.db.Participants.ToList();

}

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

{

var usersForRemoving = dtgParticipants.SelectedItems.Cast<Participants>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} учасника?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

ConferenceEntities.GetContext().Participants.RemoveRange(usersForRemoving);

ConferenceEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

dtgParticipants.ItemsSource = ConferenceEntities.GetContext().Participants.ToList();

}

catch (Exception ex)

{

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

}

}

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

{

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

}

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

{

ClassFrame.frmObj.Navigate(new AddParticipants((Participants)dtgParticipants.SelectedItem));

}

private void Menu1\_Click(object sender, RoutedEventArgs e)

{

dtgParticipants.ItemsSource = ConferenceEntities.GetContext().Participants.ToList();

CmbFiltr.SelectedIndex = -1;

}

private void CmbFiltr\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (CmbFiltr.SelectedIndex == 0)

{

dtgParticipants.ItemsSource = ConferenceEntities.GetContext().Participants.Where(x => x.IDPol == 1).ToList();

}

else

if (CmbFiltr.SelectedIndex == 1)

{

dtgParticipants.ItemsSource = ConferenceEntities.GetContext().Participants.Where(x => x.IDPol == 2).ToList();

}

}

}

**Код CS для добавления участики**

public partial class AddParticipants : Page

{

private Participants \_currentParticipants = new Participants();

string imgLoc = "пусто";

public AddParticipants(Participants selectedParticipants)

{

InitializeComponent();

CMBPol.ItemsSource = ConferenceEntities.GetContext().Pol.ToList();

CMBPol.SelectedValuePath = "IDPol";

CMBPol.DisplayMemberPath = "NamePol";

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

CMBCountry.SelectedValuePath = "IDCountry";

CMBCountry.DisplayMemberPath = "NameCountry";

if (selectedParticipants != null)

{

\_currentParticipants = selectedParticipants;

}

DataContext = \_currentParticipants;

}

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

{

StringBuilder error = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentParticipants.FIO)) error.AppendLine("Укажите ФИО");

if (string.IsNullOrWhiteSpace(\_currentParticipants.Email)) error.AppendLine("Укажите логин");

if (string.IsNullOrWhiteSpace(\_currentParticipants.DateOfBirth.ToString())) error.AppendLine("Укажите дату рождения");

if (string.IsNullOrWhiteSpace(\_currentParticipants.Phone)) error.AppendLine("Укажите номер телефона");

if (string.IsNullOrWhiteSpace(\_currentParticipants.Pasword)) error.AppendLine("Укажите пароль");

if (string.IsNullOrWhiteSpace(\_currentParticipants.IDPol.ToString())) error.AppendLine("Укажите пол");

if (error.Length > 0)

{

MessageBox.Show(error.ToString());

return;

}

if (\_currentParticipants.IDParticipants == 0)

{

ConferenceEntities.GetContext().Participants.Add(\_currentParticipants);

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

string filename = imgLoc.Substring(imgLoc.LastIndexOf('\\') + 1);

\_currentParticipants.Photo = String.Concat("/Photo/", filename);

}

if (imgLoc == "пусто") \_currentParticipants.Photo = null;

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new ParticipantsList());

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

}

catch (Exception ex)

{

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

}

}

else

{

try

{

if (imgLoc != null && imgLoc != "пусто")

{

byte[] img = null;

FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);

BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);

\_currentParticipants.Photo = imgLoc;

}

if (imgLoc == "пусто") \_currentParticipants.Photo = null;

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new ParticipantsList());

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

}

catch (Exception ex)

{

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

}

}

}

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

{

Classes.ClassFrame.frmObj.Navigate(new ParticipantsList());

}

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

{

try

{

OpenFileDialog dlg = new OpenFileDialog

{

Filter = "JPG Files (\*.jpg)|\*.jpg|PNG Files (\*.png)|\*.png|GIF Files (\*.gif)|\*.gif|All Files (\*.\*)|\*.\*",

Title = "Выберите фото/изображение организатора"

};

if (dlg.ShowDialog() == true)

{

imgLoc = dlg.FileName.ToString();

imageParticipants.Source = new BitmapImage(new Uri(imgLoc));

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

**Код CS для страницы «Активности»**

public partial class ActivitesList : Page

{

public ActivitesList()

{

InitializeComponent();

foreach (var item in dtgActivites.Items)

MessageBox.Show(item.ToString());

dtgActivites.Items.Clear();

dtgActivites.ItemsSource = ConferenceEntities.GetContext().Activities.ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

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

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (txtSearch.Text.Count() != 0)

dtgActivites.ItemsSource = ClassFrame.db.Activities.Where(x => x.Jury.FIO.Contains(txtSearch.Text.ToLower()) || x.Participants.FIO.Contains(txtSearch.Text.ToLower()) || x.Moderators.FIO.Contains(txtSearch.Text.ToLower())).ToList();

else dtgActivites.ItemsSource = ClassFrame.db.Activities.ToList();

}

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

{

var usersForRemoving = dtgActivites.SelectedItems.Cast<Activities>().ToList();

if (MessageBox.Show($"Удалить {usersForRemoving.Count()} Активность?",

"Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

try

{

ConferenceEntities.GetContext().Activities.RemoveRange(usersForRemoving);

ConferenceEntities.GetContext().SaveChanges();

MessageBox.Show("Данные удалены");

dtgActivites.ItemsSource = ConferenceEntities.GetContext().Activities.ToList();

}

catch (Exception ex)

{

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

}

}

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

{

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

}

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

{

ClassFrame.frmObj.Navigate(new AddActivites((Activities)dtgActivites.SelectedItem));

}

}

**Код CS для добавления активностей**

public partial class AddActivites : Page

{

private Activities \_currentActivities = new Activities();

public AddActivites(Activities selectedActivities)

{

InitializeComponent();

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

CMBEvent.SelectedValuePath = "IDEvent";

CMBEvent.DisplayMemberPath = "NameEvent";

CMBOrganizers.ItemsSource = ConferenceEntities.GetContext().Organizers.ToList();

CMBOrganizers.SelectedValuePath = "IDOrganizers";

CMBOrganizers.DisplayMemberPath = "FIO";

CMBModerators.ItemsSource = ConferenceEntities.GetContext().Moderators.ToList();

CMBModerators.SelectedValuePath = "IDModerators";

CMBModerators.DisplayMemberPath = "FIO";

CMBJury.ItemsSource = ConferenceEntities.GetContext().Jury.ToList();

CMBJury.SelectedValuePath = "IDJury";

CMBJury.DisplayMemberPath = "FIO";

CMBParticipants.ItemsSource = ConferenceEntities.GetContext().Participants.ToList();

CMBParticipants.SelectedValuePath = "IDParticipants";

CMBParticipants.DisplayMemberPath = "FIO";

if (selectedActivities != null)

{

\_currentActivities = selectedActivities;

}

DataContext = \_currentActivities;

}

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

{

StringBuilder error = new StringBuilder();

if (\_currentActivities.IDActivites == 0)

{

ConferenceEntities.GetContext().Activities.Add(\_currentActivities);

try

{

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new ActivitesList());

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

}

catch (Exception ex)

{

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

}

}

else

{

try

{

ConferenceEntities.GetContext().SaveChanges();

Classes.ClassFrame.frmObj.Navigate(new ActivitesList());

MessageBox.Show("Активность успешно изменена!");

}

catch (Exception ex)

# Заключение

В результате разработанная программа для конференции по информационной безопасности является незаменимым инструментом для эффективного планирования и управления мероприятием. Благодаря программе участники получают доступ к актуальной информации о конференции, а организаторы имеют возможность управлять расписанием, контролировать регистрацию и анализировать статистические данные. В целом, программа способствует успешному проведению конференции и повышению уровня удовлетворенности ее участников.

В процессе разработки данного программного приложения, я углубил свои знания и навыки в области информационных технологий, изучив множество новых п полезных методик и технологий, которые я успешно применил в ходе работы. При написании документации для программного модуля, в которой описывается как пользоваться программным продуктом. Были сформированы требования к составу и параметрам технических средств, к информационной и программной совместимости, к функциональным характеристикам.

Чтобы обеспечить уверенность в качестве приложения были выполнены модульное тестирование и тестовые сценарии. Документирование позволило рассмотреть программный модуль с точки зрения программиста, пользователя и тестировщика

# Список литературы

1.Культин Никита Борисович Основы программирования в Microsoft Visual C# 2020; БХВ-Петербург - М., 2020. - 384 c.

2.Джозеф Албахари C# 5.0. Справочник. Полное описание языка; Диалектика / Вильямс - М., 2023. - 545 c.

3.Тернстрем, Т. Microsoft SQL Server 2019. Разработка баз данных. Учебный курс Microsoft / Т. Тернстрем. - М.: Русская Редакция, 2019. - 781 c.

4.Дейтел П. Как программировать на Visual C# 2022; Питер - М., 2019. - 316 c.

5.Бен-Ган, Ицик Microsoft SQL Server 2021. Основы T-SQL / Ицик Бен-Ган. - М.: Эксмо, 2020. - 722 c.

6. ПОЛИКЛИНИЧЕСКАЯ ТЕРАПИЯ под редакцией проф. И.И. Чукаевой, проф. Б.Я. Барта 2021. - 870 с.

7.METANIT C# ПРОГРАММИРОВАНИЕ.