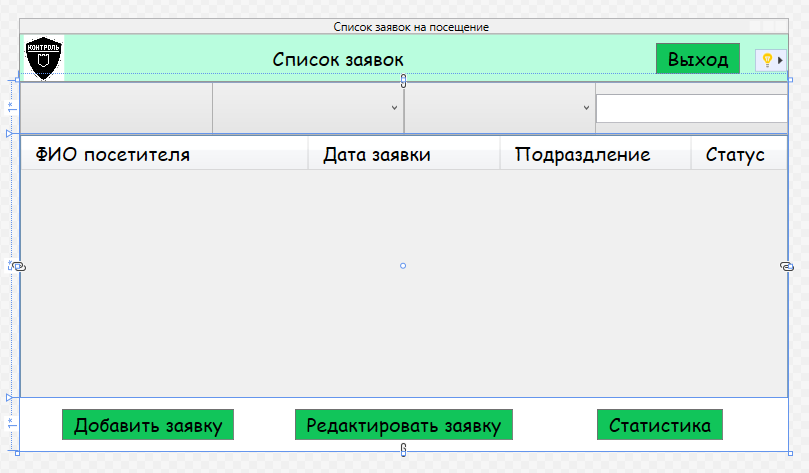
**Реализация окна «Список»**

1. Интерфейсная часть



<Window x:Class="ООО\_Адмиралтейские\_Верфи.MainWindow"

xmlns:local="clr-namespace:ООО\_Адмиралтейские\_Верфи"

mc:Ignorable="d"

Title="Список заявок на посещение" Height="450" Width="800"

Icon="/Resources/Логотип.ico" Loaded="Window\_Loaded" >

<!—Разметка всего окна-->

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid/>

<!--Заголовок-->

<Grid Background="#FFB9FDDE">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="50"/>

<ColumnDefinition Width="3\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Image Source="/Resources/Логотип.png" />

<TextBlock Grid.Column="1" Text="Список заявок" />

<Button Grid.Column="2" Content="Выход" Click="Button\_Click"/>

</Grid>

<!--Функциональная часть-->

<Grid Grid.Row="1">

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<!--Для фильтрации и поиска-->

<Grid Grid.Row="0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<ComboBox Name="cbSort" Grid.Column="0"

SelectionChanged="cbSort\_SelectionChanged" Grid.ColumnSpan="4">

<ComboBoxItem Content="Без сортировки"/>

<ComboBoxItem Content="По возрастанию"/>

<ComboBoxItem Content="По убыванию"/>

</ComboBox>

<ComboBox Name="cbDevision" Grid.Column="1"

SelectionChanged="cbDevision\_SelectionChanged" />

<ComboBox Name="cbStatus" Grid.Column="2"

SelectionChanged="cbStatus\_SelectionChanged" />

<TextBox x:Name="tbSearch" Grid.Column="3" TextChanged="tbSearch\_TextChanged"

FontFamily="Comic Sans MS" FontSize="20" VerticalAlignment="Center"/>

</Grid>

<!--Табличные данные-->

<!--<ListView Grid.Row="1" Name="lvRequest" >

<ListView.View>

<GridView >

<GridViewColumn Header="ФИО посетителя" DisplayMemberBinding="{Binding Visitor.VisitorName}" />

<GridViewColumn Header="Дата подачи заявки" DisplayMemberBinding="{Binding RequestDate, StringFormat=dd:MM:yyyy}"/>

<GridViewColumn Header="Подраздление" DisplayMemberBinding="{Binding Devision.DevisionName}"/>

<GridViewColumn Header="Статус заявки" DisplayMemberBinding="{Binding Status.StatusName}"/>

</GridView>

</ListView.View>

</ListView>-->

<DataGrid x:Name="dgRequest" Grid.Row="1" AutoGenerateColumns="False" FontSize="12"

CanUserAddRows="False">

<DataGrid.Columns>

<DataGridTextColumn Header="ФИО посетителя"

Binding="{Binding Visitor.VisitorName}" Width="3\*"/>

<DataGridTextColumn Header="Дата заявки"

Binding="{Binding RequestDate, StringFormat=dd:MM:yyyy}" Width="2\*"/>

<DataGridTextColumn Header="Подразделение"

Binding="{Binding Devision.DevisionName}" Width="2\*"/>

<DataGridTextColumn Header="Статус" Binding="{Binding Status.StatusName}"

Width="\*"/>

</DataGrid.Columns>

</DataGrid>

<!--Подвал с кнопками-->

<Grid Grid.Row="2">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Button x:Name="butAdd" Grid.Column="0" Content="Добавить заявку"

Click="butAdd\_Click"/>

<Button x:Name="butEdit" Grid.Column="1" Content="Редактировать заявку"

Click="butEdit\_Click"/>

<Button x:Name="butStatictic" Grid.Column="2" Content="Статистика"

Click="butStatictic\_Click"/>

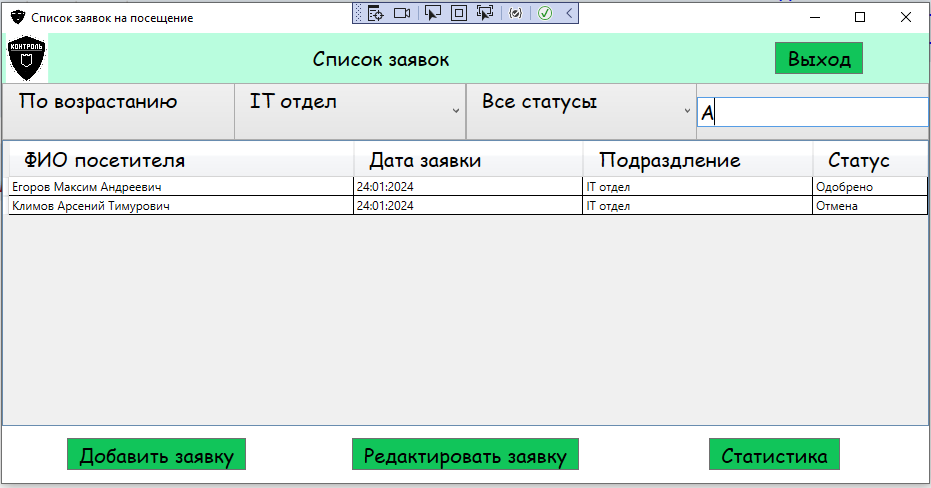
</Grid>

</Grid>

</Grid>

</Window>

1. Логическая часть



using ООО\_Адмиралтейские\_Верфи.Model;

namespace ООО\_Адмиралтейские\_Верфи

{

/// Логика взаимодействия для MainWindow.xaml

public partial class MainWindow : Window

{

List<Model.Request> requests;

public MainWindow()

{

InitializeComponent();

try

{

App.DB = new Model.DBShipyard(); //Создание подключения к БД

//MessageBox.Show("Удалось подключиться к БД");

}

catch

{

//MessageBox.Show("Не удалось подключиться к БД");

App.Current.Shutdown();

}

}

/// Завершить работу

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

{

this.Close();

}

/// Отображать заявки с учетом фильтрации и поиска

private void ShowRequest()

{

//Все данные из БД

try

{

//Получить все данные из БД

requests = App.DB.Request.ToList();

//lvRequest.ItemsSource = requests;

}

catch

{

MessageBox.Show("Не удалось получить данные о заявках на посещение");

App.Current.Shutdown();

}

//Сортировка

if (cbSort.SelectedIndex>0)

{

switch(cbSort.SelectedIndex)

{

case 1:

requests = requests.OrderBy(x => x.RequestDate).ToList();

break;

case 2:

requests = requests.OrderByDescending(x => x.RequestDate).ToList();

break;

}

}

//Фильтрация по подразделениям

if (cbDevision.SelectedIndex > 0)

{

requests = requests.Where(x =>

x.RequestDevision == (int)cbDevision.SelectedValue).ToList();

}

//Фильтрация по статусам

if (cbStatus.SelectedIndex > 0)

{

requests = requests.Where(x => x.RequestStatus == (int)cbStatus.SelectedValue).ToList();

}

//Поиск по ФИО

string search = tbSearch.Text;

if (!String.IsNullOrEmpty(search))

{

requests = requests.Where(x => x.Visitor.VisitorName.Contains(search) == true).ToList();

}

//Отобразить результаты фильтрации и поиска

dgRequest.ItemsSource = requests;

}

/// При загрузке окна

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

cbSort.SelectedIndex = 0;

//Все подразделения

List<Model.Devision> devisions = App.DB.Devision.ToList();

Model.Devision devision = new Devision();

devision.DevisionId = 0;

devision.DevisionName = "Все подразделения";

devisions.Insert(0, devision);

cbDevision.ItemsSource = devisions;

cbDevision.DisplayMemberPath = "DevisionName";

cbDevision.SelectedValuePath = "DevisionId";

cbDevision.SelectedIndex = 0;

//Все статусы

List<Model.Status> statuses = App.DB.Status.ToList();

Model.Status status = new Model.Status();

status.StatusId = 0;

status.StatusName = "Все статусы";

statuses.Insert(0, status);

cbStatus.ItemsSource = statuses;

cbStatus.DisplayMemberPath = "StatusName";

cbStatus.SelectedValuePath = "StatusId";

cbStatus.SelectedIndex = 0;

ShowRequest();

}

/// Сортировка

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

{

ShowRequest();

}

//Фильтрация по подразделениям

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

{

ShowRequest();

}

//Фильтрация по статусу

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

{

ShowRequest();

}

//Поиск по ФИО

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

{

ShowRequest();

}

/// Добавление новой заявки

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

{ }

/// Редактирование имеющейся заявки

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

{ }

/// Статистика по заявкам

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

{ }

}

}