Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

Колледж информатики и программирования

Специальность 09.02.03 «Программирование в компьютерных системах»

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8**

Выполнил:

обучающийся учебной группы № 220

М.Ю. Иванова

*(И.О. Фамилия)*

Проверил:

руководитель практики от колледжа:

И.В. Сибирев

*(И.О. Фамилия)*

**Москва**

**20­­­­23**

*Цель работы:* используя выданный программный код, переписать его под WPF.

**Ход работы**

1. В начале я разработала логику для окна программы MainWindow.xaml.cs:

using System;

using System.Net.Http;

using System.Net;

using System.Threading.Tasks;

using System.Windows;

using System.ComponentModel;

using System.Windows.Controls;

using System.Windows.Input;

namespace Messanger\_A

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

HttpListener \_HttpListener = new HttpListener()

.Set\_Prefixes\_Add("http://127.0.0.1:8888/connection/", \_IsOpen: false)

.Set\_Start();

private Style TextBlockStyle = (Style)Application.Current.Resources["TextBlockStyle"];

private Style BorderkStyle = (Style)Application.Current.Resources["BorderkStyle"];

private Style Gray = (Style)Application.Current.Resources["Gray"];

public static BackgroundWorker BackgroundWorker = new BackgroundWorker();

public MainWindow()

{

InitializeComponent();

MaxHeight = System.Windows.SystemParameters.PrimaryScreenHeight - 26;

BackgroundWorker.DoWork += GetMessegesWork;

BackgroundWorker.RunWorkerAsync(null);

}

public void GetMessegesWork(object sender, DoWorkEventArgs e)

{

while (true)

{

AddTB(new HttpClient().GetStringAsync("http://127.0.0.2:8889/connection/?qweBFYUWEFGUYSDADSDFW").GetAwaiter().GetResult(), HorizontalAlignment.Left, TextAlignment.Left);

}

}

private void AddTB(string Text, HorizontalAlignment ha, TextAlignment ta)

{

Application.Current.Dispatcher.Invoke(() =>

{

TextBlock tb = new TextBlock

{

Text = Text,

Style = TextBlockStyle,

TextAlignment = ta,

HorizontalAlignment = ha,

Margin = new Thickness(5)

};

Border border = new Border

{

Style = BorderStyle,

Background = Gray,

HorizontalAlignment = ha,

Child = tb

};

MessegesSP.Children.Add(border);

});

}

private void SendClick(object sender, RoutedEventArgs e)

{

SendMessege();

}

private void TBKeyUp(object sender, KeyEventArgs e)

{

if (e.Key == Key.Enter) SendMessege();

}

private void SendMessege()

{

AddTB(SendText.Text, HorizontalAlignment.Right, TextAlignment.Right);

\_HttpListener

.Get\_ContextAsync(a =>

{

a.Response.Set\_Bytes(SendText.Text.Get\_Encoding\_UTF8\_Bytes());

}

);

SendText.Text = string.Empty;

}

}

}

1. Далее я сконструировала разметку окна MainWindow.xaml:

<Window x:Class="Messanger\_A.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Messanger\_A"

mc:Ignorable="d"

Title="Messenger\_A" Height="450" Width="800" Closing="Window\_Closing" Loaded="Window\_Loaded">

<Grid Background="White">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="20"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="20"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="20"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40\*"/>

<RowDefinition Height="20"/>

</Grid.RowDefinitions>

<Grid Margin="5" Grid.Column="1" Grid.Row="1">

<ScrollViewer VerticalScrollBarVisibility="Auto">

<Border Style="{DinamicResourse BorderStyle}">

<StackPanel x:Name="Messenger" Orientation="Vertical" Margin="10"/>

</Border>

</ScrollViewer>

</Grid>

<Grid Margin="5" Grid.Column="1" Grid.Row="2">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="8\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<TextBox x:Name="SendText" Style="{DinamicResourse TextBoxStyle}" KeyUp="TBKeyUp" Grid.Column="0"/>

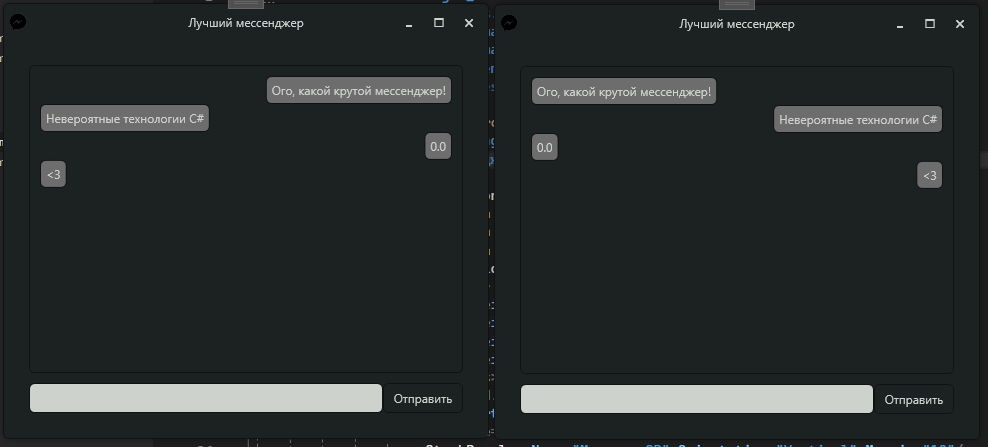
<Button Content="Отправить" Style="{DinamicResourse ButtonStyle}" Click="SendClick" Grid.Column="1"/>

</Grid>

</Grid>

</Window>

1. Как итог проделанной работы получился следующий мессенджер:



*Вывод:* используя выданный программный код, я переписала его под WPF, попутно изучив новые средства и возможности.