UDC 004.42

DEVELOPING AN AUTHORISATION INTERFACE USING WPF

V.I. Pylypenko, Senior Lecturer, Department of ICT
Kyiv National University of Technologies and Design
M. S. Petrchuk, student
Kyiv National University of Technologies and Design

Keywords: WPF, C#, XAML, authorization, interface.

In solving user interface design problems – WPF is one of the key technological tools that allows for effective code management and optimization of interface display. WPF (Windows Presentation Foundation) is a graphical subsystem for creating user interfaces, which was introduced as part of the .NET Framework 3.0. It is an analogue of WinForms, but provides enhanced capabilities for developing modern and visually attractive desktop applications [1-2]. Using WPF to create a graphical authorization interface brings numerous advantages. One of them is the use of the XAML markup language, which allows you to define the interface structure and styles. The implementation of the authorization interface using the XAML markup language is presented in Listing 1.

Listing 1. Implementing the user authorization interface

```
<Window x: Class="UserLoginApp.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="WPFLogin" Height="440" Width="700" ResizeMode="NoResize"
WindowStartupLocation="CenterScreen">
  <Grid>
    <Grid HorizontalAlignment="Left">
      <Grid.Background>
        <LinearGradientBrush EndPoint="2,1" StartPoint="-2,1"
Opacity="0.8">
          <GradientStop Color="Black" Offset="1"/> <GradientStop
Color="White"/> </LinearGradientBrush></Grid.Background>
      <StackPanel Width="246" Height="158">
        <TextBlock Text="Раді вас бачити!" FontSize="20"
FontWeight="Bold" HorizontalAlignment="Center" FontFamily="Book
Antiqua"/>
      <WrapPanel Height="40"><Image Height="31" Width="30"
Source="/userICON.png"/>
          <TextBox x: Name="UsernameTextBox" FontSize="18"
Width="211" Height="30" BorderThickness="0,0,0,1" BorderBrush="Black">
            <TextBox.Background>
    <SolidColorBrush Color="White" Opacity="0"/></TextBox.Background>
       </TextBox> </WrapPanel> <WrapPanel Height="40" Width="246">
       <Image Height="30" Width="31" Source="/lockICON.png"/>
```

```
            <PasswordBox x:Name="PasswordBox" Width="211" FontSize="15"
Height="32" BorderThickness="0,0,0,1" BorderBrush="Black">
        <PasswordBox.Background> <SolidColorBrush Color="White"
Opacity="0"/>
             </PasswordBox.Background> </PasswordBox> </WrapPanel>
           <Button Content="Авторизуватися" Width="242" Height="40"Click =
"LoginButton_Click" FontSize="15" FontFamily="Book Antiqua" FontWeight
= "Normal">
       <Button.Background><SolidColorBrush Color="#FFDDDDDD"
Opacity="0"/>
         </Button.Background><Button.BorderBrush><SolidColorBrush
Color="#FF707070" Opacity="0"/> </Button.BorderBrush>
</Button></StackPanel> </Grid><Grid.Background>
<ImageBrushImageSource="/educate.jpg"/></Grid.Background>
   </Grid>
</Window>
```

The general view of the implemented authorization window using the XAML and WPF markup language is presented in Fig. 1.

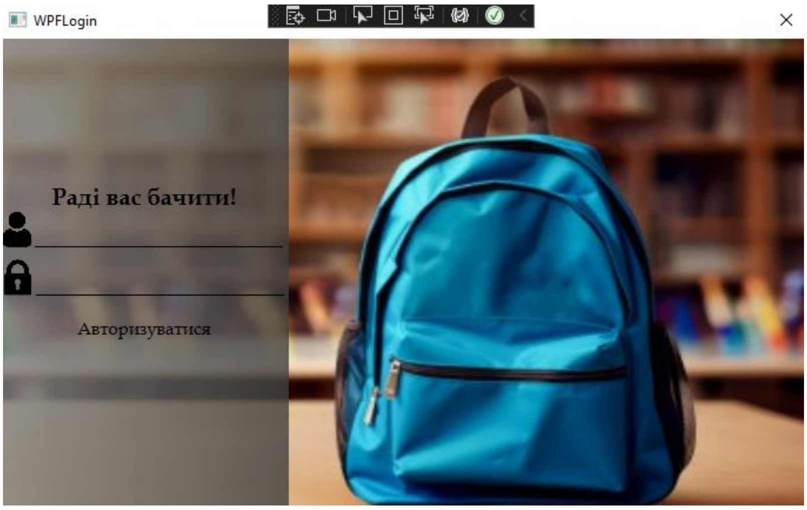

Figure 1 – User authorization window

Using WPF and XAML simplifies interface creation because designers can work with XAML without having to dive into code. This provides data binding capabilities and allows for easy integration of business logic with the user interface.

References

1. Yuen, S. (2020). Mastering Windows Presentation Foundation: Build responsive UIs for desktop applications with WPF. Packt Publishing Ltd.

2. Troelsen, A., Japikse, P., Troelsen, A., & Japikse, P. (2021). WPF Notifications, Validations, Commands, and MVVM. Pro C# 9 with. NET 5: Foundational Principles and Practices in Programming, 1143-1177.