# Book Action Page

## XAML

<Page x:Class="BookLove.Views.Pages.BookPage.BookActionPage"

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

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

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

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

xmlns:local="clr-namespace:BookLove.Views.Pages.BookPage"

mc:Ignorable="d" FontFamily="Century Gothic"

FontSize="16"

d:DesignHeight="700" d:DesignWidth="1280"

Title="Добавление / Редактирование">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center">

<Border BorderThickness="2">

<Image Width="250"

Height="250"

Name="BookPicture"

Source="{Binding Book.GetPicture}"/>

</Border>

<Button Style="{StaticResource nowidhtButtonStyle}"

Content="Выбрать"

Margin="0,5,0,0"

Click="SelectedPicture\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

<Grid Grid.Column="1" Margin="16,0,-16,0">

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Width="300">

<TextBlock Text="Наименование: "/>

<TextBox Text="{Binding Book.title}"/>

<TextBlock Text="Имя Автора: "/>

<TextBox Text="{Binding Author.firstName}"/>

<TextBlock Text="Фамилия Автора: "/>

<TextBox Text="{Binding Author.lastName}"/>

<TextBlock Text="Отчество Автора: "/>

<TextBox Text="{Binding Author.secondName}"/>

<TextBlock Text="Издатель: "/>

<ComboBox SelectedItem="{Binding Book.Publisher}"

ItemsSource="{Binding Publishers}"

DisplayMemberPath="titile"/>

<TextBlock Text="Жанр: "/>

<ComboBox SelectedItem="{Binding Book.Genre}"

ItemsSource="{Binding Genres}"

DisplayMemberPath="title"/>

<TextBlock Text="Количество: "/>

<TextBox Text="{Binding Book.count}"/>

<TextBlock Text="Цена: "/>

<TextBox Text="{Binding Book.price}"

Name="txbTotalPrice"/>

<Button Style="{StaticResource nowidhtButtonStyle}"

Margin="0,10,0,0"

Content="Сохранить"

Click="BookSave\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

</Grid>

</Page>

## C#

using BookLove.Context;

using BookLove.Model;

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace BookLove.Views.Pages.BookPage

{

/// <summary>

/// Interaction logic for BookActionPage.xaml

/// </summary>

public partial class BookActionPage : Page

{

public List<Publisher> Publishers { get; set; }

public List<Genre> Genres { get; set; }

public Author Author { get; set; }

public Book Book { get; set; }

public BookActionPage(Book book, Author author)

{

InitializeComponent();

Publishers = AppData.db.Publisher.ToList();

Genres = AppData.db.Genre.ToList();

Author = author;

Book = book;

this.DataContext = this;

}

OpenFileDialog file = new OpenFileDialog();

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

{

file.Filter = "Image (\*.jpg;\*.jpeg;\*png;)|\*.jpg;\*.jpeg;\*png;";

BookPicture.Source = file.ShowDialog() == true ? BookPicture.Source = new BitmapImage(new Uri(file.FileName)) : null;

}

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

{

if(Book.id == 0 && Author.id == 0)

{

AppData.db.Author.Add(Author);

AppData.db.Book.Add(Book);

}

File.Copy(file.FileName, $"pics\\{System.IO.Path.GetFileName(file.FileName).Trim()}", true);

Book.GetPicture = "\\pics\\" + System.IO.Path.GetFileName(file.FileName);

AppData.db.SaveChanges();

MessageBox.Show("Данные успешно сохранены в Базе Данных.", "Внимание, операция прошла успешно!", MessageBoxButton.OK, MessageBoxImage.Information);

GC.Collect();

NavigationService.GoBack();

}

}

}

# Book ViewData Page

## XAML

<Page x:Class="BookLove.Views.Pages.BookPage.BookViewPage"

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

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

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

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

xmlns:local="clr-namespace:BookLove.Views.Pages.BookPage"

mc:Ignorable="d" FontFamily="Century Gothic"

FontSize="18" Loaded="Page\_Loaded"

d:DesignHeight="700" d:DesignWidth="1280"

Title="Книги">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="8\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid Grid.Row="0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

</Grid>

<Grid Grid.Row="0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Horizontal"

Grid.Column="0"

HorizontalAlignment="Center"

VerticalAlignment="Center">

<TextBlock Text="Экспорт: "

VerticalAlignment="Center"

FontSize="24"

FontWeight="Bold"/>

<Button Style="{StaticResource historyPageButtonStyle}"

Content=".PDF"

Click="btnPDFExport\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<Button Style="{StaticResource historyPageButtonStyle}"

Content=".CSV"

Click="btnCSVExport\_Click"

Margin="10,0,0,0">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

</StackPanel>

<StackPanel Grid.Column="1"

VerticalAlignment="Center"

HorizontalAlignment="Center"

Orientation="Horizontal">

<StackPanel>

<TextBlock Text="Поиск: "/>

<TextBox Name="txbSearch"

TextChanged="txbSearch\_TextChanged"

Width="250"/>

</StackPanel>

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

<TextBlock Text="Фильтрация по Жанру"/>

<ComboBox Width="250"

Name="cmbFilterGender"

SelectionChanged="cmbFilterGender\_SelectionChanged">

<ComboBoxItem>Все типы</ComboBoxItem>

<ComboBoxItem>Роман</ComboBoxItem>

<ComboBoxItem>Саморазвитие</ComboBoxItem>

<ComboBoxItem>Детектив</ComboBoxItem>

<ComboBoxItem>Дорама</ComboBoxItem>

</ComboBox>

</StackPanel>

</StackPanel>

</Grid>

<Grid Grid.Row="1">

<ListView Name="BookListView">

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<Image Source="{Binding GetPicture}"

Width="300"

Height="300"/>

</Grid>

<Grid Grid.Column="1"

Margin="10,40,0,0">

<StackPanel>

<Label Content="{Binding title}" ContentStringFormat="Наименование: {0}"/>

<Label Content="{Binding Author.GetFullName}" ContentStringFormat="Автор: {0}"/>

<Label Content="{Binding Genre.title}" ContentStringFormat="Жанр: {0}"/>

<Label Content="{Binding Publisher.titile}" ContentStringFormat="Издатель: {0}"/>

<Label Content="{Binding count}" ContentStringFormat="Количество: {0}"/>

<Label Content="{Binding price}" ContentStringFormat="Цена: {0}"/>

</StackPanel>

</Grid>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

<Grid Grid.Row="2">

<StackPanel Orientation="Horizontal"

VerticalAlignment="Center"

HorizontalAlignment="Center">

<Button Style="{StaticResource functionButtonStyle}"

Content="Добавить"

Click="CreateBookPage\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<Button Style="{StaticResource functionButtonStyle}"

Content="Редактировать"

Margin="10,0" Click="EditBookPage\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<Button Style="{StaticResource functionButtonStyle}"

Content="Удалить"

Click="RemoveSelectedItem\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

</Grid>

</Page>

## C#

using BookLove.Context;

using BookLove.Model;

using System;

using System.Diagnostics;

using System.IO;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

using Word = Microsoft.Office.Interop.Word;

namespace BookLove.Views.Pages.BookPage

{

/// <summary>

/// Interaction logic for BookViewPage.xaml

/// </summary>

public partial class BookViewPage : Page

{

public BookViewPage()

{

InitializeComponent();

}

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

{

NavigationService.Navigate(new BookActionPage(new Book(), new Author()));

}

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

{

var selectedBook = BookListView.SelectedItem as Book;

var selectedAuthor = selectedBook.Author;

if (selectedBook != null)

NavigationService.Navigate(new BookActionPage(selectedBook, selectedAuthor));

else

MessageBox.Show("Чтобы редактировать запись, вам необходимо его выбрать.", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Exclamation);

}

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

{

BookListView.ItemsSource = AppData.db.Book.ToList();

}

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

{

var selectedBook = BookListView.SelectedItem as Book;

if (selectedBook != null)

{

if (MessageBox.Show("Вы действительно хотите удалить данные? Данные будут удалены навсегда.", "Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

AppData.db.Book.Remove(selectedBook);

AppData.db.SaveChanges();

Page\_Loaded(null, null);

GC.Collect();

}

else

MessageBox.Show("Чтобы удалить запись, вам необходимо его выбрать.", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Exclamation);

}

}

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

{

var word = new Word.Application();

try

{

var document = word.Documents.Add();

var paragraph = word.ActiveDocument.Paragraphs.Add();

var tableRange = paragraph.Range;

var books = AppData.db.Book.ToList();

var table = document.Tables.Add(tableRange, books.Count, 6);

table.Borders.Enable = 1;

table.Cell(1, 0).Range.Text = "Наименование";

table.Cell(1, 1).Range.Text = "Автор";

table.Cell(1, 2).Range.Text = "Жанр";

table.Cell(1, 3).Range.Text = "Издатель";

table.Cell(1, 4).Range.Text = "Количество";

table.Cell(1, 5).Range.Text = "Цена";

int i = 2;

foreach (var item in books)

{

table.Cell(i, 0).Range.Text = item.title;

table.Cell(i, 1).Range.Text = item.Author.GetFullName;

table.Cell(i, 2).Range.Text = item.Genre.title;

table.Cell(i, 3).Range.Text = item.Publisher.titile;

table.Cell(i, 4).Range.Text = item.count.ToString();

table.Cell(i, 5).Range.Text = item.price.ToString();

i++;

}

document.SaveAs2(Environment.CurrentDirectory + @"books.pdf", Word.WdSaveFormat.wdFormatPDF);

document.Close(Word.WdSaveOptions.wdDoNotSaveChanges);

word.Quit(Word.WdSaveOptions.wdDoNotSaveChanges);

MessageBox.Show("Сохранение прошло успешно!", "Сохранено!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source + " выдал исключение!", MessageBoxButton.OK, MessageBoxImage.Error);

word.Quit(Word.WdSaveOptions.wdDoNotSaveChanges);

}

if (MessageBox.Show("Документ PDF успешно сформирован. Хотите открыть его?", "Операция прошла успешно!", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)

{

Process.Start(Environment.CurrentDirectory + @"books.pdf");

}

}

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

{

using (FileStream stream = new FileStream(Environment.CurrentDirectory + @"books.csv", FileMode.Create))

{

using (StreamWriter writer = new StreamWriter(stream))

{

var books = AppData.db.Book.ToList();

writer.WriteLine("Наименование;Автор;Жанр;Издатель;Цена;Количество");

foreach (var item in books)

{

writer.WriteLine($"{item.title};{item.Author.GetFullName};{item.Genre.title};{item.Publisher.titile};{item.count}");

}

}

}

MessageBox.Show($"Сохранение прошло успешно! Откройте документ тут: {Environment.CurrentDirectory}", "Сохранено!", MessageBoxButton.OK, MessageBoxImage.Information); ;

}

private void Update(string filter = "", string search = "")

{

var books = AppData.db.Book.ToList();

if (!string.IsNullOrEmpty(filter) && !string.IsNullOrEmpty(filter))

{

if (filter == "Роман" || filter == "Саморазвитие" || filter == "Детектив" || filter == "Дорама")

{

books = AppData.db.Book.Where(item => item.Genre.title == filter).ToList();

}

}

if (!string.IsNullOrEmpty(search) && !string.IsNullOrEmpty(search))

books = books.Where(item => item.title.ToLower().Contains(txbSearch.Text.ToLower()) || item.Author.GetFullName.ToLower().Contains(search.ToLower())).ToList();

BookListView.ItemsSource = books;

}

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

{

Update((cmbFilterGender.SelectedItem as ComboBoxItem).Content.ToString(), txbSearch.Text);

}

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

{

Update(cmbFilterGender.Text, txbSearch.Text);

}

}

}

# Create client page

## XAML

<Page x:Class="BookLove.Views.Pages.ClientPage.CreateClientPage"

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

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

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

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

xmlns:local="clr-namespace:BookLove.Views.Pages.ClientPage"

mc:Ignorable="d" FontSize="16" FontFamily="Century Gothic"

d:DesignHeight="700" d:DesignWidth="1280"

Title="Добавить клиента">

<Grid>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Width="300">

<TextBlock Text="Наименование: "/>

<TextBox Text="{Binding Client.title}"/>

<TextBlock Text="Город: "/>

<ComboBox SelectedItem="{Binding Client.City}"

DisplayMemberPath="title"

ItemsSource="{Binding Cities}"/>

<TextBlock Text="Адрес: "/>

<TextBox Text="{Binding Client.street}" Height="80"

TextWrapping="Wrap"/>

<Button Style="{StaticResource nowidhtButtonStyle}"

Margin="0,10,0,0"

Content="Создать"

Click="SaveClient\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<Button Style="{StaticResource nowidhtButtonStyle}"

Content="Клиент существует"

Margin="0,10,0,0"

Click="CreateSkipClient\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

</Page>

## C#

using BookLove.Context;

using BookLove.Model;

using BookLove.Views.Pages.OrderPage;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace BookLove.Views.Pages.ClientPage

{

/// <summary>

/// Interaction logic for CreateClientPage.xaml

/// </summary>

public partial class CreateClientPage : Page

{

public Client Client { get; set; }

public List<City> Cities{ get; set; }

public CreateClientPage(Client client)

{

InitializeComponent();

Cities = AppData.db.City.ToList();

Client = client;

this.DataContext = this;

}

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

{

if(Client.id == 0)

{

AppData.db.Client.Add(Client);

}

AppData.db.SaveChanges();

MessageBox.Show("Клиент успешно зарегистрирован!", "Успешно!", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.Navigate(new OrderCreatePage());

}

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

{

NavigationService.Navigate(new OrderCreatePage());

}

}

}

# History page

## XAML

<Page x:Class="BookLove.Views.Pages.OrderPage.HistoryOrderPage"

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

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

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

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

xmlns:local="clr-namespace:BookLove.Views.Pages.OrderPage"

mc:Ignorable="d" FontFamily="Century Gothic"

FontSize="18"

d:DesignHeight="700" d:DesignWidth="1280"

Title="HistoryOrderPage">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="8\*"/>

</Grid.RowDefinitions>

<Grid Grid.Row="0">

<StackPanel Orientation="Horizontal"

VerticalAlignment="Center"

HorizontalAlignment="Center">

<TextBlock Text="Выберите дату заказа: "/>

<DatePicker Width="200"

Name="dtpDateHistory"

SelectedDateChanged="dtpDateHistory\_SelectedDateChanged"/>

<Button Style="{StaticResource historyPageButtonStyle}"

Content="Сбросить"

Margin="10,0,10,0"

Click="btnFilterOff\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<TextBlock Text="Вывести: "/>

<Button Style="{StaticResource historyPageButtonStyle}"

Content="PDF"

Click="btnPDFExport\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<Button Style="{StaticResource historyPageButtonStyle}"

Content="CSV"

Margin="10,0,10,0"

Click="btnCSV\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

<Grid Grid.Row="1">

<ListView Name="HistoryOrderList">

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<StackPanel>

<Label Content="{Binding Client.GetFullTitle}" ContentStringFormat="Данные Клиента: {0}"/>

<Label Content="{Binding Book.GetFullTitle}" ContentStringFormat="Книга: {0}"/>

<Label Content="{Binding Provider.title}" ContentStringFormat="Поставщик: {0}"/>

<Label Content="{Binding count}" ContentStringFormat="Количество: {0} шт."/>

<Label Content="{Binding dateSupple}" ContentStringFormat="Дата покупки: {0}"/>

<Label Content="{Binding totalPrice}" ContentStringFormat="Итоговая цена: {0} руб."/>

<Label Content="{Binding description}" ContentStringFormat="Описание: {0}"/>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Grid>

</Page>

## C#

using BookLove.Context;

using System;

using System.Diagnostics;

using System.IO;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using Word = Microsoft.Office.Interop.Word;

namespace BookLove.Views.Pages.OrderPage

{

/// <summary>

/// Interaction logic for HistoryOrderPage.xaml

/// </summary>

public partial class HistoryOrderPage : Page

{

public HistoryOrderPage()

{

InitializeComponent();

HistoryOrderList.ItemsSource = AppData.db.Supple.ToList();

}

private void dtpDateHistory\_SelectedDateChanged(object sender, SelectionChangedEventArgs e)

{

HistoryOrderList.ItemsSource = AppData.db.Supple.Where(item => item.dateSupple == dtpDateHistory.SelectedDate).ToList();

}

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

{

dtpDateHistory.SelectedDate = null;

HistoryOrderList.ItemsSource = AppData.db.Supple.ToList();

}

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

{

var word = new Word.Application();

try

{

var document = word.Documents.Add();

var paragraph = word.ActiveDocument.Paragraphs.Add();

var tableRange = paragraph.Range;

var supples = AppData.db.Supple.ToList();

var table = document.Tables.Add(tableRange, supples.Count, 7);

table.Borders.Enable = 1;

table.Cell(1, 0).Range.Text = "Клиент";

table.Cell(1, 1).Range.Text = "Книга";

table.Cell(1, 2).Range.Text = "Поставщик";

table.Cell(1, 3).Range.Text = "Дата";

table.Cell(1, 4).Range.Text = "Цена";

table.Cell(1, 5).Range.Text = "Количество";

table.Cell(1, 6).Range.Text = "Описание";

int i = 2;

foreach (var item in supples)

{

table.Cell(i, 0).Range.Text = item.Client.GetFullTitle;

table.Cell(i, 1).Range.Text = item.Book.GetFullTitle;

table.Cell(i, 2).Range.Text = item.Provider.title;

table.Cell(i, 3).Range.Text = item.dateSupple.ToString();

table.Cell(i, 4).Range.Text = item.totalPrice.ToString();

table.Cell(i, 5).Range.Text = item.count.ToString();

table.Cell(i, 6).Range.Text = item.description;

i++;

}

document.SaveAs2(Environment.CurrentDirectory + @"history.pdf", Word.WdSaveFormat.wdFormatPDF);

document.Close(Word.WdSaveOptions.wdDoNotSaveChanges);

word.Quit(Word.WdSaveOptions.wdDoNotSaveChanges);

MessageBox.Show("Сохранение прошло успешно!", "Сохранено!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source + " выдал исключение!", MessageBoxButton.OK, MessageBoxImage.Error);

word.Quit(Word.WdSaveOptions.wdDoNotSaveChanges);

}

if (MessageBox.Show("Документ PDF успешно сформирован. Хотите открыть его?", "Операция прошла успешно!", MessageBoxButton.YesNo, MessageBoxImage.Information) == MessageBoxResult.Yes)

{

Process.Start(Environment.CurrentDirectory + @"history.pdf");

}

}

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

{

using (FileStream stream = new FileStream(Environment.CurrentDirectory + @"history.csv", FileMode.Create))

{

using (StreamWriter writer = new StreamWriter(stream))

{

var supples = AppData.db.Supple.ToList();

writer.WriteLine("Клиент;Книга;Поставщик;Дата;Цена;Количество;Описание");

foreach (var item in supples)

{

writer.WriteLine($"{item.Client.GetFullTitle};{item.Book.GetFullTitle};{item.dateSupple};{item.totalPrice};{item.count};{item.description}");

}

}

}

MessageBox.Show("Сохранение прошло успешно!", "Сохранено!", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

}

# Order create

## XAML

<Page x:Class="BookLove.Views.Pages.OrderPage.OrderCreatePage"

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

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

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

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

xmlns:local="clr-namespace:BookLove.Views.Pages.OrderPage"

mc:Ignorable="d"

FontSize="16"

d:DesignHeight="700" d:DesignWidth="1280"

Title="Создать заказ">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center"

Width="300">

<TextBlock Text="Поставщик: "/>

<ComboBox Name="cmbProvider"

DisplayMemberPath="titile"

SelectionChanged="ComboBoxProvider\_SelectionChanged"/>

<TextBlock Text="Книга: "/>

<ComboBox Name="cmbBook"

DisplayMemberPath="GetFullTitle"

SelectionChanged="ComboBoxBook\_SelectionChanged"/>

<TextBlock Text="Клиент: "/>

<ComboBox Name="cmbClient"

SelectionChanged="ComboBoxClient\_SelectionChanged"

DisplayMemberPath="GetFullTitle"/>

<TextBlock Text="Описание (необязательно): "/>

<TextBox Name="txbDescription" Height="100"

TextWrapping="Wrap"/>

<TextBlock Text="Цена: "/>

<StackPanel Orientation="Horizontal">

<TextBlock Width="200"

Name="txbTotalPrice"/>

<Button Content="+"

Margin="10,0,0,0"

FontSize="20"

Click="btnUp\_Click"

Width="30"/>

<TextBlock Text="0"

Name="txbCount"

Width="15"

Margin="5,0"

VerticalAlignment="Center"/>

<Button Content="-"

FontSize="20"

Width="30"

Click="btnDown\_Click"/>

</StackPanel>

<Button Style="{StaticResource nowidhtButtonStyle}"

Margin="0,10,0,0"

Content="Добавить в заказ"

Click="ButtonAddOrder\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<Button Style="{StaticResource nowidhtButtonStyle}"

Margin="0,5,0,0"

Click="ButtonFinishOrder\_Click"

Content="Оформить заказ">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

<Grid Grid.Column="1">

<Grid.RowDefinitions>

<RowDefinition Height="9\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<DataGrid x:Name="BasketList"

AutoGenerateColumns="False"

ColumnWidth="\*">

<DataGrid.Columns>

<DataGridTextColumn Header="Код" Binding="{Binding id}"/>

<DataGridTextColumn Header="Клиент" Binding="{Binding Client.GetFullTitle}"/>

<DataGridTextColumn Header="Поставщик" Binding="{Binding Provider.title}"/>

<DataGridTextColumn Header="Книга" Binding="{Binding Book.GetFullTitle}"/>

<DataGridTextColumn Header="Количество" Binding="{Binding count}"/>

<DataGridTextColumn Header="Цена" Binding="{Binding totalPrice}"/>

</DataGrid.Columns>

</DataGrid>

<StackPanel Grid.Row="1"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Orientation="Horizontal">

<Button Style="{StaticResource functionButtonStyle}"

Content="Удалить"

Click="RemoveSelectedBasket\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

</Grid>

</Page>

## C#

using BookLove.Context;

using BookLove.Model;

using BookLove.Views.Pages.SystemPage;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace BookLove.Views.Pages.OrderPage

{

/// <summary>

/// Interaction logic for OrderCreatePage.xaml

/// </summary>

public partial class OrderCreatePage : Page

{

private int \_idClient;

private int \_idProvider;

private int \_idBook;

private int \_balance = 1;

private int \_countProdcut { get; set; }

private int \_count = 1;

private double \_total { get; set; }

private double \_price { get; set; }

public OrderCreatePage()

{

InitializeComponent();

txbCount.Visibility = Visibility.Hidden;

cmbBook.ItemsSource = AppData.db.Book.ToList();

cmbBook.DisplayMemberPath = "GetFullTitle";

cmbClient.ItemsSource = AppData.db.Client.ToList();

cmbBook.DisplayMemberPath = "GetFullTitle";

cmbProvider.ItemsSource = AppData.db.Provider.ToList();

cmbProvider.DisplayMemberPath = "title";

}

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

{

txbCount.Visibility = Visibility.Visible;

if (\_countProdcut != 0)

{

if (\_countProdcut == \_count)

{

MessageBox.Show("Вы первысили количество товара, которое имеется в наличии склада.", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Exclamation);

}

else

{

\_count++;

\_balance = \_countProdcut - \_count;

\_total += \_price;

}

txbTotalPrice.Text = "Итого: " + \_total.ToString() + " руб.";

txbCount.Text = \_count.ToString();

}

else

{

MessageBox.Show("Вы первысили количество товара, которое имеется в наличии склада.", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Exclamation);

}

}

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

{

if (\_count != 0)

{

\_count--;

\_total -= \_price;

}

\_balance = \_countProdcut + \_count;

txbTotalPrice.Text = "Итого: " + \_total.ToString() + " руб.";

txbCount.Text = \_count.ToString();

}

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

{

\_idProvider = (cmbProvider.SelectedItem as Provider).id;

}

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

{

\_idBook = (cmbBook.SelectedItem as Book).id;

\_total = (cmbBook.SelectedItem as Book).price;

\_countProdcut = (cmbBook.SelectedItem as Book).count;

\_price = \_total;

txbCount.Visibility = Visibility.Visible;

if (\_idBook != 0)

{

\_count = 1;

txbCount.Text = \_count.ToString();

txbTotalPrice.Text = "Итого: " + \_total.ToString() + " руб.";

}

else

{

MessageBox.Show("Данного товара нет в наличии.", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Exclamation);

txbCount.Text = "0";

txbTotalPrice.Text = "0";

}

}

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

{

\_idClient = (cmbClient.SelectedItem as Client).id;

}

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

{

try

{

var selectedItem = BasketList.SelectedItem as Basket;

if (selectedItem != null)

{

AppData.db.Basket.Remove(selectedItem);

AppData.db.SaveChanges();

BasketList.ItemsSource = AppData.db.Basket.ToList();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

Supple supple = new Supple();

Basket basket = new Basket();

supple.idClient = \_idClient;

supple.idBook = \_idBook;

supple.dateSupple = DateTime.Now;

supple.count = \_count;

supple.totalPrice = \_total;

supple.idProvider = \_idProvider;

supple.description = txbDescription.Text;

AppData.db.Supple.Add(supple);

basket.idClient = \_idClient;

basket.idBook = \_idBook;

basket.idProvider = \_idProvider;

basket.totalPrice = \_total;

basket.count = \_count;

AppData.db.Basket.Add(basket);

AppData.db.SaveChanges();

BasketList.ItemsSource = AppData.db.Basket.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

try

{

var basket = AppData.db.Basket.ToList();

AppData.db.Basket.RemoveRange(basket);

AppData.db.SaveChanges();

PrintDialog print = new PrintDialog();

if (print.ShowDialog() == true)

{

print.PrintVisual(BasketList, "Check");

}

MessageBox.Show("Оформление чека прошло успешно.", "Чек сохранен!", MessageBoxButton.OK, MessageBoxImage.Information);

BasketList.ItemsSource = AppData.db.Basket.ToList();

NavigationService.Navigate(new MenuPage());

GC.Collect();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, ex.Source, MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

}

# Login Page

## XAML

<Page x:Class="BookLove.Views.Pages.SystemPage.LoginPage"

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

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

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

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

xmlns:local="clr-namespace:BookLove.Views.Pages.SystemPage"

mc:Ignorable="d" FontFamily="Century Gothic"

FontSize="18"

d:DesignHeight="600" d:DesignWidth="1080"

Title="Вход">

<Grid >

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Column="0">

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center">

<TextBlock Text="Будь влюблен в свою жизнь," FontSize="34"/>

<TextBlock Text="В каждую её минуту" FontSize="34"

Margin="0,5,0,0"/>

<TextBlock Text="--------------------------------------------" FontSize="34"/>

</StackPanel>

</Grid>

<StackPanel HorizontalAlignment="Center"

Grid.Column="1"

VerticalAlignment="Center">

<StackPanel Orientation="Horizontal">

<TextBlock Text="Логин: "

Width="120"/>

<TextBox Name="txbLogin"

Width="350"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Пароль: "

Width="120"/>

<PasswordBox Name="psbPassword"

Width="350"/>

<TextBox Width="350"

Name="PasswordTextBox"

Visibility="Collapsed"

Text="{Binding ElementName=psbPassword, Path=Password, Mode=TwoWay}"/>

</StackPanel>

<StackPanel Orientation="Horizontal"

Name="StackCaptche"

Visibility="Collapsed">

<StackPanel Orientation="Horizontal">

<TextBlock TextDecorations="Strikethrough"

Name="txbCaptchaText"

Width="120"/>

<TextBox Name="CaptchaTextBox"

Width="200"/>

</StackPanel>

<Button Content="Repack"

Name="btnRepack"

Style="{StaticResource nowidhtButtonStyle}"

Width="150"

Click="Repack\_Click"/>

</StackPanel>

<Button Content="Показать пароль"

Style="{StaticResource nowidhtButtonStyle}"

Name="ShowPassworButton"

Click="ShowPassworButton\_Click"

Margin="0,5,0,0">

<Button.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="15"/>

</Style>

</Button.Resources>

</Button>

<Button Content="Войти"

IsDefault="True"

Name="LoginButton"

Style="{StaticResource nowidhtButtonStyle}"

Click="LoginButton\_Click"

Margin="0,5">

<Button.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="15"/>

</Style>

</Button.Resources>

</Button>

<Button Content="Регистрация"

Style="{StaticResource nowidhtButtonStyle}"

Name="SignUpButton"

Click="SignUpButton\_Click">

<Button.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="15"/>

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

</Page>

## C#

using BookLove.Context;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace BookLove.Views.Pages.SystemPage

{

/// <summary>

/// Interaction logic for LoginPage.xaml

/// </summary>

public partial class LoginPage : Page

{

public LoginPage()

{

InitializeComponent();

}

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

{

var currentLogin = AppData.db.Login.FirstOrDefault(item => item.username == txbLogin.Text && item.password == psbPassword.Password);

if (txbCaptchaText.Text == CaptchaTextBox.Text)

{

if (currentLogin != null)

{

NavigationService.Navigate(new MenuPage());

}

else

{

MessageBox.Show("Такого пользователя не существует!", "В доступе отказано!", MessageBoxButton.OK, MessageBoxImage.Error);

txbCaptchaText.Text = GetCaptchaCode();

StackCaptche.Visibility = Visibility;

}

}

else

{

MessageBox.Show("Вы не прошли капчу, пожалуйста повторите попытку", "Вниманиме!", MessageBoxButton.OK, MessageBoxImage.Error);

txbCaptchaText.Text = GetCaptchaCode();

StackCaptche.Visibility = Visibility;

}

}

string GetCaptchaCode()

{

string alphabet = "QWERTYUIOPASDFGHJKLZXCVBNM1234567890";

string password = "";

Random random = new Random();

for (int i = 0; i < 8; i++) password += alphabet[random.Next(alphabet.Length)];

return password;

}

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

{

if (PasswordTextBox.Visibility == Visibility.Collapsed)

{

PasswordTextBox.Text = psbPassword.Password;

PasswordTextBox.Visibility = Visibility.Visible;

psbPassword.Visibility = Visibility.Collapsed;

}

else

{

PasswordTextBox.Visibility = Visibility.Collapsed;

psbPassword.Visibility = Visibility.Visible;

}

}

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

{

NavigationService.Navigate(new RegistrationPage());

}

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

{

txbCaptchaText.Text = GetCaptchaCode();

}

}

}

# Menu Page

## XAML

<Page x:Class="BookLove.Views.Pages.SystemPage.MenuPage"

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

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

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

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

xmlns:local="clr-namespace:BookLove.Views.Pages.SystemPage"

mc:Ignorable="d" FontFamily="Century Gothic"

d:DesignHeight="700" d:DesignWidth="1280"

Title="Меню">

<Grid>

<StackPanel HorizontalAlignment="Center"

VerticalAlignment="Center">

<Button Style="{StaticResource functionButtonStyle}"

Content="Книги"

Click="BookActionPage\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<Button Style="{StaticResource functionButtonStyle}"

Content="Сформировать заказ"

Margin="0,10"

Click="CreateOrderPage\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<Button Style="{StaticResource functionButtonStyle}"

Content="История"

Click="HistoryPage\_Click">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

<Button Style="{StaticResource functionButtonStyle}"

Content="Выйти"

Margin="0,10">

<Button.Resources>

<Style TargetType="Border">

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

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

</Page>

## C#

using BookLove.Model;

using BookLove.Views.Pages.BookPage;

using BookLove.Views.Pages.ClientPage;

using BookLove.Views.Pages.OrderPage;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace BookLove.Views.Pages.SystemPage

{

/// <summary>

/// Interaction logic for MenuPage.xaml

/// </summary>

public partial class MenuPage : Page

{

public MenuPage()

{

InitializeComponent();

}

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

{

NavigationService.Navigate(new BookViewPage());

}

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

{

NavigationService.Navigate(new CreateClientPage(new Client()));

}

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

{

NavigationService.Navigate(new HistoryOrderPage());

}

}

}

# Registration Page

## XAML

<Page x:Class="BookLove.Views.Pages.SystemPage.RegistrationPage"

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

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

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

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

xmlns:local="clr-namespace:BookLove.Views.Pages.SystemPage"

mc:Ignorable="d"

FontFamily="Century Gothic"

FontSize="18"

d:DesignHeight="600" d:DesignWidth="1080"

Title="Регистрация">

<Grid>

<StackPanel VerticalAlignment="Top"

HorizontalAlignment="Center"

Margin="0,10,0,0">

<TextBlock Text="Для регистрации, заполните поля ниже."

FontSize="24"/>

</StackPanel>

<StackPanel VerticalAlignment="Center"

HorizontalAlignment="Center"

Width="300">

<TextBlock Text="Электронная почта: "/>

<TextBox Name="txbEmail"/>

<TextBlock Text="Фамилия: "/>

<TextBox Name="txbLastName"/>

<TextBlock Text="Имя: "/>

<TextBox x:Name="txbFirstName"/>

<TextBlock Text="Отчество (необязательно): "/>

<TextBox Name="txbSecondName"/>

<TextBlock Text="Номер договора: "/>

<TextBox Name="txbNumberContract"/>

<Button Content="Зарегистрироваться"

IsDefault="True"

Style="{StaticResource nowidhtButtonStyle}"

Margin="0,10,0,0"

Click="btnRegistration\_Click">

<Button.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="15"/>

</Style>

</Button.Resources>

</Button>

</StackPanel>

</Grid>

</Page>

## C#

using BookLove.Context;

using BookLove.HelperObject;

using BookLove.Model;

using System;

using System.Linq;

using System.Net;

using System.Net.Mail;

using System.Windows;

using System.Windows.Controls;

namespace BookLove.Views.Pages.SystemPage

{

/// <summary>

/// Interaction logic for RegistrationPage.xaml

/// </summary>

public partial class RegistrationPage : Page

{

public RegistrationPage()

{

InitializeComponent();

}

public static string GenericPassword()

{

// Получаем количество слов и букв за слово.

int num\_letters = 8;

int num\_words = 1;

// Создаем массив букв, которые мы будем использовать.

char[] letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdifghijklmnopqrstuvwxyz123456789".ToCharArray();

// Создаем генератор случайных чисел.

Random rand = new Random();

// Сделайте слово.

string word = "";

// Делаем слова.

for (int i = 1; i <= num\_words; i++)

{

for (int j = 1; j <= num\_letters; j++)

{

// Выбор случайного числа от 0 до 25

// для выбора буквы из массива букв.

int letter\_num = rand.Next(0, letters.Length - 1);

// Добавить письмо.

word += letters[letter\_num];

}

}

return word;

}

static string currentPassword = GenericPassword();

private async void SendMessage()

{

try

{ // отправитель - устанавливаем адрес и отображаемое в письме имя

MailAddress from = new MailAddress("jewralyapp@mail.ru", "Admin JewralyApp SoftWare");

// кому отправляем

MailAddress to = new MailAddress(txbEmail.Text);

// создаем объект сообщения

MailMessage m = new MailMessage(from, to);

// тема письма

m.Subject = "Администратор одобрил ваш запрос!";

// текст письма

m.Body = "<h2>Ваш логин: " + txbEmail.Text + ", Ваш пароль: " + currentPassword + " </h2>";

// письмо представляет код html

m.IsBodyHtml = true;

// адрес smtp-сервера и порт, с которого будем отправлять письмо

SmtpClient smtp = new SmtpClient("smtp.mail.ru", 587);

// логин и пароль

smtp.Credentials = new NetworkCredential("jewralyapp@mail.ru", "@Lacky12345");

smtp.EnableSsl = true;

await smtp.SendMailAsync(m);

Console.Read();

MessageBox.Show($"{txbFirstName.Text}, ваш запрос отправлен, Администратор системы рассмотрит ваш запрос как можно быстрее. Дождитесь пожалуйста ответа.",

"Запрос успешно отправлен!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

Login login = new Login();

Registration registration = new Registration();

try

{

if (HelperClass.IsValidEmail(txbEmail.Text))

{

if (AppData.db.Login.Count(item => item.username == txbEmail.Text) > 0)

{

MessageBox.Show("Учётная запись существует. Пожалуйста, если вы потеряли доступ, обратитесь к Администартору Системы.",

"В доступе отказано!", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

else

{

SendMessage();

login.username = txbEmail.Text;

login.password = currentPassword;

registration.firstName = txbFirstName.Text;

registration.lastName = txbLastName.Text;

registration.secondName = txbSecondName.Text;

registration.idLogin = login.id;

registration.numberContract = txbNumberContract.Text;

registration.dateOfRegistration = DateTime.Now;

AppData.db.Login.Add(login);

AppData.db.Registration.Add(registration);

AppData.db.SaveChanges();

}

}

else

{

MessageBox.Show("Некорректный формат почты! Пожалуйста, внимательно прочтите, какой набор символов вы ввели и повторите попытку!",

"Некорректный формать логина!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch (Exception ex)

{

throw;

}

}

}

}