Федеральное государственное образовательное бюджетное учреждение  
высшего образования

«Финансовый университет при Правительстве Российской Федерации»

КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

**ОТЧЕТ**

**по результатам проектирования**

**и разработки программного приложения**

**по профессиональному модулю ПМ.03 Участие в интеграции программных модулей**

**специальности 09.02.03 Программирование в компьютерных системах**

Выполнил:

студент группы 4ПКС-\_\_\_

Ф.И.О. студента

Проверили:

преподаватели

\_\_\_\_\_\_\_\_ / Т.Г. Аксёнова /

\_\_\_\_\_\_\_\_/О.А. Тимакин/

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2023

Оглавление

[Проектирование программного обеспечения 3](#_Toc154589995)

[Руководство оператора 4](#_Toc154589996)

[Назначение программы 4](#_Toc154589997)

[Условия выполнения программы 4](#_Toc154589998)

[Выполнение программы 5](#_Toc154589999)

[Сообщения оператору 5](#_Toc154590000)

[Листинг программных модулей 6](#_Toc154590001)

## Проектирование программного обеспечения

Для программного приложения «Калькулятор ипотечного кредитования» были разработаны диаграммы вариантов использования и «сущность-связь».

На рисунке 1 представлена диаграмма вариантов использования для программного приложения «Калькулятор ипотечного кредитования».

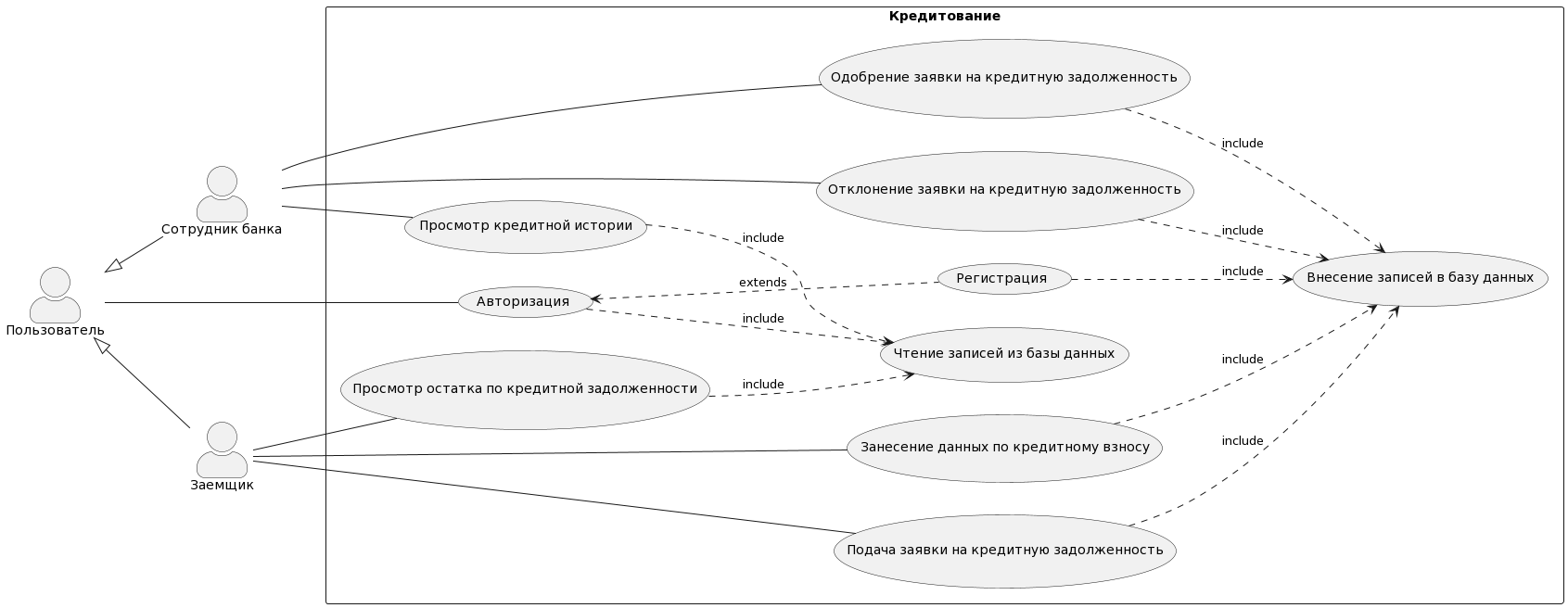


Рисунок 1. Диаграмма вариантов использования

На рисунке 2 представлена диаграмма «сущность-связь» для программного приложения «Калькулятор ипотечного кредитования».

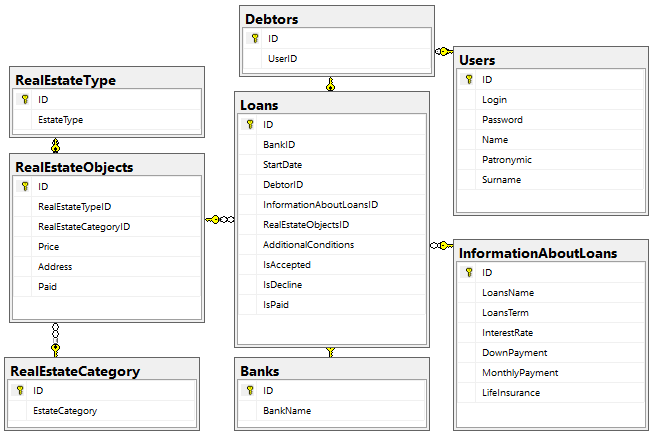


Рисунок 2. Диаграмма «сущность-связь»

## Руководство оператора

### **Назначение программы**

Назначение программного приложения «Калькулятор ипотечного кредитования» необходима для работы в банковской структуре в зависимости от роли пользователя. Для пользователей с ролью «Заемщик» доступен такой функционал, как:

* возможность оформить кредитование с указанием данных о банке, типе кредита, сроке кредита, процентной ставке, начальном взносе, страховании жизни, типе собственности, на которую оформляется кредит/ипотека, ее категория, ее адрес, сумма кредита и различные необходимые дополнительные условия;
* просмотреть остатки по кредитной задолженности, где отображена подробная информация о всех тех кредитах, которые имеет авторизированный должник;
* добавить данные о месячном взносе по необходимому кредиту.

Для пользователей, имеющих роль «Банковский сотрудник», имеется функционал просмотра заявок на кредит с возможностью отклонить или принять данную заявку на основании дополнительно предоставленной информации.

### **Условия выполнения программы**

Аппаратные требования для корректного функционирования программы представлены в таблице 1.

Таблица 1. Аппаратные требования к продукту

|  |  |  |
| --- | --- | --- |
|  | Минимальные | Рекомендуемые |
| Процессор | 1 гигагерц (ГГц) c двумя ядрами | 2 гигагерц (ГГц) c 4-мя ядрами |
| Оперативная память | 4 Гб | 8 Гб |
| Видеоадаптер и монитор | VGA/HDMI(720 x 1280) | HDMI(1920 x 1080) или выше |
| Свободное место на HDD | 1 Гб | 1 Гб |
| Устройства взаимодействия с пользователем | клавиатура и мышь | клавиатура и мышь |

Благодаря этому будет обеспечена комфортная и бесперебойная работа с программным приложением.

### **Выполнение программы**

Для загрузки и запуска программы необходимо открыть проект в Visual Studio и нажать кнопку «Пуск», представленную на рисунке 3.



Рисунок 3. Кнопка «Пуск»

Для входа в программу необходимо ввести логин в поле «Логин» и пароль в поле «Пароль» с последующим нажатием кнопки «Войти», представленной на рисунке 4.

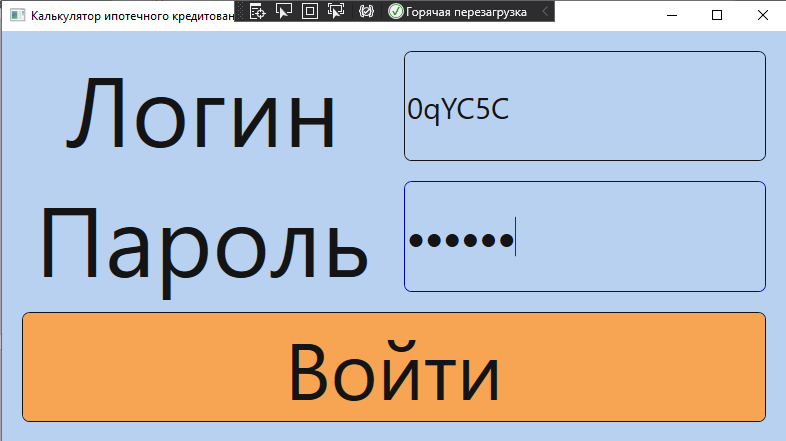


Рисунок 4. Кнопка «Войти» с заполненными полями

Для отбора данных по ФИО заемщика необходимо ввести значение в поле «ФИО заемщика», представленное на рисунке 5.

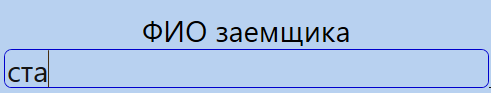


Рисунок 5. Фильтрация по ФИО заемщика

Для отбора данных по типу кредитования необходимо выбрать значение в поле «Тип кредитования», представленное на рисунке 6.

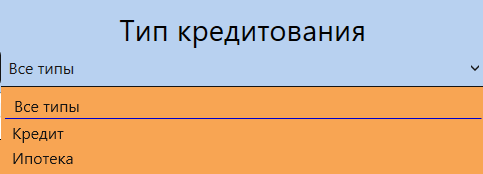


Рисунок 6. Фильтрация по типу кредитования

Для одобрения/отклонения заявки по кредиту необходимо нажать на соответствующую кнопку в строке с информацией по кредиту, представленной на рисунке 7.



Рисунок 7. Пример информации по кредиту

Для того, чтобы подать заявку на кредит, необходимо выбрать пункт меню «Подать заявку на кредит», заполнить все поля на странице и нажать кнопку «Подать заявку», представленные на рисунке 8.

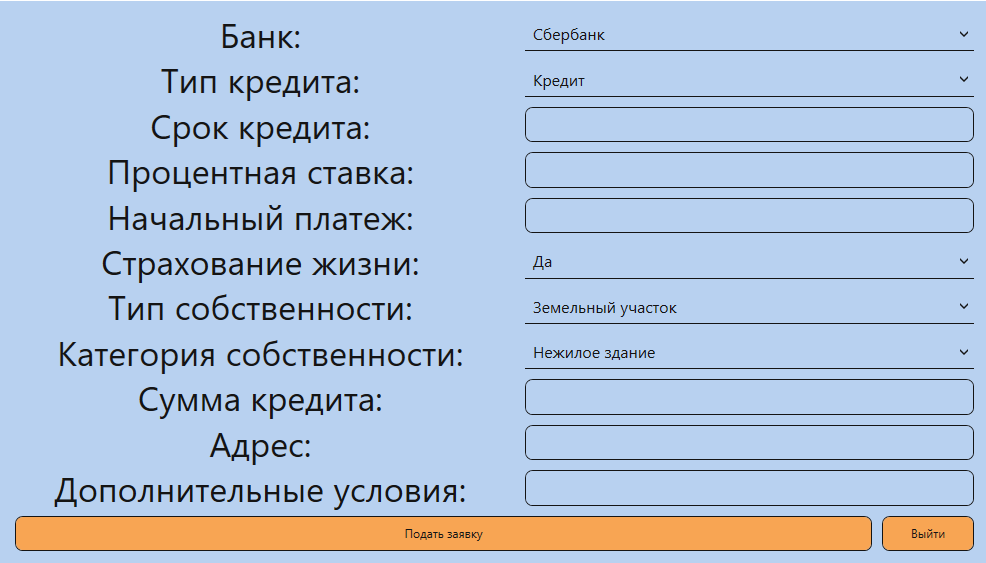


Рисунок 8. Данные для подачи заявки на кредитование

Для просмотра остатков по кредитной задолженности необходимо выбрать пункт «Просмотреть остатки по кредитной задолженности» представленные на рисунке 9.



Рисунок 9. Меню действий заемщика

Для того, чтобы внести месячный взнос по кредиту, необходимо нажать кнопку «Добавить взнос», представленную на рисунке 10.

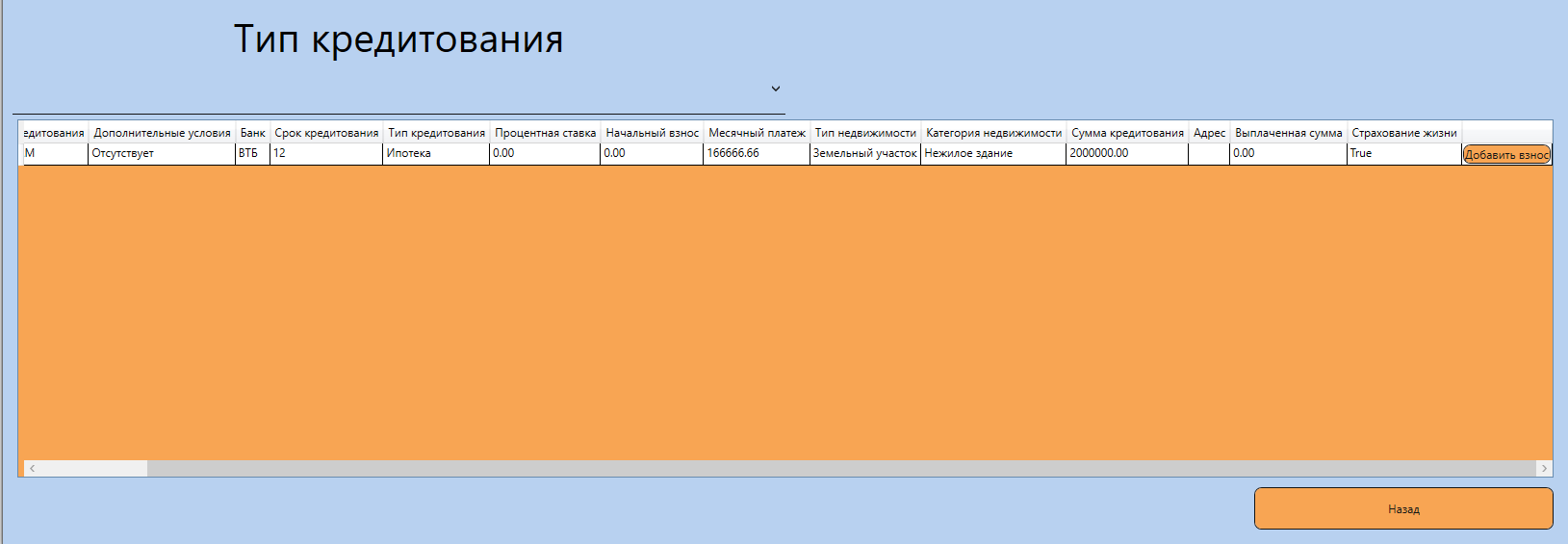


Рисунок 10. Кнопка «Добавить взнос»

Для закрытия приложения необходимо нажать на крестик в правом верхнем углу приложения, представленный на рисунке 11.



Рисунок 11. Кнопка закрытия приложения

### **Сообщения оператору**

### В ходе работы с программой могут возникнуть некоторые сообщения.

При возникновении сообщения «Рады приветствовать Вас снова, #ИмяОтчество», «Месячный платеж внесен!», «Заявка успешно оформлена», «Задолженность погашена», «Кредит для #ИмяОтчество одобрен», «Кредит для #ИмяОтчество отклонен» никаких действий предпринимать не нужно.

При возникновении сообщения с ошибкой, например, «Срок кредита должен быть числом», «Срок кредита должен быть больше или равен 1» и т.д.? требуется исправить недочет и нажать кнопку снова.

## Листинг программных модулей

Код файла ApplyingForLoanPage.xaml представлен ниже.

<Page x:Class="\_220\_Smirnov.Pages.Assets.ApplyingForLoanPage"

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:\_220\_Smirnov.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ApplyingForLoanPage">

<Grid Style="{DynamicResource GridStyle}" Margin="10">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.5\*"/>

<ColumnDefinition Width="0.35\*"/>

<ColumnDefinition Width="0.1\*"/>

</Grid.ColumnDefinitions>

<Viewbox Grid.Row="0">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Банк: "/>

</Viewbox>

<ComboBox Style="{DynamicResource ComboBoxStyle}" Grid.Row="0" Grid.Column="1" Margin="5" x:Name="Banks" DisplayMemberPath="BankName" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="1">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Тип кредита: "/>

</Viewbox>

<ComboBox Style="{DynamicResource ComboBoxStyle}" Grid.Row="1" Grid.Column="1" Margin="5" x:Name="LoansName" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="2">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Срок кредита: "/>

</Viewbox>

<TextBox Style="{DynamicResource TextBoxStyle}" Grid.Row="2" Grid.Column="1" Margin="5" x:Name="LoansTerm" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="3">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Процентная ставка: "/>

</Viewbox>

<TextBox Style="{DynamicResource TextBoxStyle}" Grid.Row="3" Grid.Column="1" Margin="5" x:Name="InterestRate" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="4">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Начальный платеж: "/>

</Viewbox>

<TextBox Style="{DynamicResource TextBoxStyle}" Grid.Row="4" Grid.Column="1" Margin="5" x:Name="DownPayment" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="5">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Страхование жизни: "/>

</Viewbox>

<ComboBox Style="{DynamicResource ComboBoxStyle}" Grid.Row="5" Grid.Column="1" Margin="5" x:Name="LifeInsurance" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="6">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Тип собственности: "/>

</Viewbox>

<ComboBox Style="{DynamicResource ComboBoxStyle}" Grid.Row="6" Grid.Column="1" Margin="5" x:Name="RealEstateType" DisplayMemberPath="EstateType" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="7">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Категория собственности: "/>

</Viewbox>

<ComboBox Style="{DynamicResource ComboBoxStyle}" Grid.Row="7" Grid.Column="1" Margin="5" x:Name="RealEstateCategory" DisplayMemberPath="EstateCategory" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="8">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Сумма кредита: "/>

</Viewbox>

<TextBox Style="{DynamicResource TextBoxStyle}" Grid.Row="8" Grid.Column="1" Margin="5" x:Name="Price" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="9">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Адрес: "/>

</Viewbox>

<TextBox Style="{DynamicResource TextBoxStyle}" Grid.Row="9" Grid.Column="1" Margin="5" x:Name="Address" Grid.ColumnSpan="2"/>

<Viewbox Grid.Row="10">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Дополнительные условия: "/>

</Viewbox>

<TextBox Style="{DynamicResource TextBoxStyle}" Grid.Row="10" Grid.Column="1" Margin="5" x:Name="AdditionalConditions" Grid.ColumnSpan="2"/>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="11" Grid.Column="0" Grid.ColumnSpan="2" Margin="5" Content="Подать заявку" Click="Button\_Click"/>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="11" Grid.Column="2" Margin="5" Content="Выйти" Click="Button\_Click\_1"/>

</Grid>

</Page>

Код файла ApplyingForLoanPage.xaml.cs представлен ниже.

using \_220\_Smirnov.Database;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace \_220\_Smirnov.Pages.Assets

{

public partial class ApplyingForLoanPage : Page

{

private Dictionary<string, bool> \_lifeInsurance = new Dictionary<string, bool>()

{

["Да"] = true,

["Нет"] = false

};

private LendingEntities \_db = new LendingEntities();

private int \_debtorId;

// Установка начальный значений

public ApplyingForLoanPage(int debtorId)

{

InitializeComponent();

\_debtorId = debtorId;

Banks.ItemsSource = \_db.Banks.ToList();

LoansName.ItemsSource = new[] { "Кредит", "Ипотека" };

LifeInsurance.ItemsSource = \_lifeInsurance.Keys;

RealEstateType.ItemsSource = \_db.RealEstateTypes.ToList();

RealEstateCategory.ItemsSource = \_db.RealEstateCategories.ToList();

new List<ComboBox> { Banks, LoansName, LifeInsurance, RealEstateType, RealEstateCategory }.ForEach(cb => cb.SelectedIndex = 0);

}

~ApplyingForLoanPage()

{

\_db.Dispose();

}

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

{

try

{

// обработка данных

if (!Int32.TryParse(LoansTerm.Text, out var loansTerm))

throw new Exception("Срок кредита должен быть числом");

if (loansTerm <= 0)

throw new Exception("Срок кредита должен быть больше или равен 1");

if (!Decimal.TryParse(InterestRate.Text, out var interestRate))

throw new Exception("Процентная ставка должна быть числом");

if (interestRate < 0)

throw new Exception("Процентная ставка не может быть отрицательна");

if (!Decimal.TryParse(DownPayment.Text, out var downPayment))

throw new Exception("Начальный платеж должен быть числом");

if (downPayment < 0)

throw new Exception("Начальный платеж не может быть отрицательным");

if (!Decimal.TryParse(Price.Text, out var price))

throw new Exception("Сумма кредита должна быть числом");

if (price <= 0)

throw new Exception("Сумма кредита не может быть меньше или равна 0");

if (downPayment >= price)

throw new Exception("Сумма начального платежа не может быть больше или равна сумме кредита");

// рассчет итоговой суммы кредита с учетом процентной ставки

decimal priceWithLoansTerm = price + price \* (100 + interestRate) / 100;

// добавление информации о кредите в базу данных

InformationAboutLoan informationAboutLoan = new InformationAboutLoan

{

LoansName = LoansName.Text,

LoansTerm = loansTerm,

InterestRate = interestRate,

DownPayment = downPayment,

MonthlyPayment = (priceWithLoansTerm - downPayment) / loansTerm,

LifeInsurance = \_lifeInsurance[LifeInsurance.Text]

};

\_db.InformationAboutLoans.Add(informationAboutLoan);

// добавление информации о недвижимости в базу данных

RealEstateObject realEstateObject = new RealEstateObject()

{

RealEstateTypeID = ((RealEstateType)RealEstateType.SelectedValue).ID,

RealEstateCategoryID = ((RealEstateCategory)RealEstateCategory.SelectedValue).ID,

Price = priceWithLoansTerm,

Address = String.IsNullOrWhiteSpace(Address.Text) ? null : Address.Text,

Paid = downPayment

};

\_db.RealEstateObjects.Add(realEstateObject);

// добавление кредита в базу данных

Loan loan = new Loan

{

BankID = ((Bank)Banks.SelectedValue).ID,

StartDate = DateTime.Now,

DebtorID = \_debtorId,

InformationAboutLoansID = informationAboutLoan.ID,

RealEstateObjectsID = realEstateObject.ID,

AdditionalConditions = String.IsNullOrWhiteSpace(AdditionalConditions.Text) ? "Отсутствует" : AdditionalConditions.Text,

IsAccepted = false,

IsDecline = false,

IsPaid = false

};

\_db.Loans.Add(loan);

\_db.SaveChanges();

MessageBox.Show("Заявка успешно оформлена!");

NavigationService.GoBack();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

// Выход назад

private void Button\_Click\_1(object sender, RoutedEventArgs e)

=> NavigationService.GoBack();

}

}

Код файла AuthorizationPage.xaml представлен ниже.

<Page x:Class="\_220\_Smirnov.Assets.Pages.AuthorizationPage"

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:\_220\_Smirnov.Assets.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AuthorizationPage">

<Grid Style="{DynamicResource GridStyle}" Margin="10">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Viewbox Grid.Row="0" Grid.Column="0">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Логин" />

</Viewbox>

<TextBox Style="{DynamicResource TextBoxStyle}" x:Name="Login" Grid.Row="0" Grid.Column="1" Margin="10"/>

<Viewbox Grid.Row="1" Grid.Column="0">

<TextBlock Style="{DynamicResource TextBlockStyle}" Text="Пароль" />

</Viewbox>

<PasswordBox Style="{DynamicResource PasswordBoxStyle}" x:Name="Password" Grid.Row="1" Grid.Column="1" Margin="10"/>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="2" Grid.Column="0" Grid.ColumnSpan="2" Margin="10" Click="Button\_Click">

<Button.Content>

<Viewbox>

<TextBlock Text="Войти"/>

</Viewbox>

</Button.Content>

</Button>

</Grid>

</Page>

Код файла AuthorizationPage.xaml.cs представлен ниже.

using \_220\_Smirnov.Database;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace \_220\_Smirnov.Assets.Pages

{

public partial class AuthorizationPage : Page

{

private LendingEntities \_db = new LendingEntities();

public AuthorizationPage()

=> InitializeComponent();

~AuthorizationPage()

{

\_db.Dispose();

}

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

{

try

{

// Поиск пользователя

User user = \_db.Users.FirstOrDefault(u => u.Login == Login.Text && u.Password == Password.Password);

if (user is null)

throw new System.Exception("Пользователь не найден");

MessageBox.Show($"Рады видеть Вас снова, {user.Name} {user.Patronymic}");

// Проверка роли пользователя и переход к необходимой для этой роли странице

Debtor debtor = \_db.Debtors.FirstOrDefault(d => d.UserID == user.ID);

if (debtor != null)

NavigationService.Navigate(new DebtorsMainPage(debtor.ID));

else

NavigationService.Navigate(new BankEmployeerMainPage());

}

catch

{

MessageBox.Show("Некорректные данные");

}

}

}

}

Код файла BankEmployeerMainPage.xaml представлен ниже.

<Page x:Class="\_220\_Smirnov.Assets.Pages.BankEmployeerMainPage"

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:\_220\_Smirnov.Assets.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="BankEmployeerMainPage">

<Grid Style="{DynamicResource GridStyle}" Margin="10">

<Grid.RowDefinitions>

<RowDefinition Height="0.2\*"/>

<RowDefinition Height="0.7\*"/>

<RowDefinition Height="0.1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.9\*"/>

<ColumnDefinition Width="0.1\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Row="0" Grid.ColumnSpan="2">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Viewbox Grid.Row="0" Grid.Column="0">

<TextBlock Text="ФИО заемщика"/>

</Viewbox>

<TextBox Style="{DynamicResource TextBoxStyle}" Grid.Row="1" Grid.Column="0" TextChanged="TextBox\_TextChanged" Name="DebtorFN"/>

<Viewbox Grid.Row="0" Grid.Column="1">

<TextBlock Text="Тип кредитования"/>

</Viewbox>

<ComboBox Name="LoansN" Grid.Column="1" Grid.Row="1" Style="{DynamicResource ComboBoxStyle}" SelectionChanged="LoansN\_SelectionChanged"/>

</Grid>

<DataGrid Style="{DynamicResource DataGridStyle}" Margin="5" Grid.ColumnSpan="2" Grid.Row="1" AutoGenerateColumns="False" x:Name="Loans">

<DataGrid.Columns>

<DataGridTextColumn Header="Фамилия" Binding="{Binding DebtorSurname}"/>

<DataGridTextColumn Header="Имя" Binding="{Binding DebtorName}"/>

<DataGridTextColumn Header="Отчество" Binding="{Binding DebtorPatronymic}"/>

<DataGridTextColumn Header="Дополнительные условия" Binding="{Binding AdditionalConditions}"/>

<DataGridTextColumn Header="Банк" Binding="{Binding Bank}"/>

<DataGridTextColumn Header="Срок кредитования" Binding="{Binding LoansTerm}"/>

<DataGridTextColumn Header="Тип кредитования" Binding="{Binding LoansName}"/>

<DataGridTextColumn Header="Процентная ставка" Binding="{Binding InterestRate}"/>

<DataGridTextColumn Header="Страхование жизни" Binding="{Binding LifeInsurance}"/>

<DataGridTemplateColumn>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Style="{DynamicResource ButtonStyle}" Content="Одобрить" Click="Button\_Click" Tag="{Binding Id}"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTemplateColumn>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Style="{DynamicResource ButtonStyle}" Content="Отклонить" Click="Button\_Click\_1" Tag="{Binding Id}"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Content="Преобразовать в txt" Click="Button\_Click\_3" Grid.Column="0" Grid.Row="2" Style="{DynamicResource ButtonStyle}"/>

<Button Content="Выйти" Click="Button\_Click\_2" Grid.Column="1" Grid.Row="2" Style="{DynamicResource ButtonStyle}"/>

</Grid>

</Page>

Код файла BankEmployeerMainPage.xaml.cs представлен ниже.

using \_220\_Smirnov.Database;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.IO;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace \_220\_Smirnov.Assets.Pages

{

public partial class BankEmployeerMainPage : Page

{

private LendingEntities \_db = new LendingEntities();

// Установка начальный значений

public BankEmployeerMainPage()

{

InitializeComponent();

UpdateList(\_db.Loans);

LoansN.ItemsSource = new[] { "Все типы", "Кредит", "Ипотека" };

LoansN.SelectedIndex = 0;

}

~BankEmployeerMainPage()

{

\_db.Dispose();

}

// Одобрение кредита

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

=> Click(Int32.Parse(((Button)sender).Tag.ToString()), "одобрен", true, false);

// Отклонение кредита

private void Button\_Click\_1(object sender, RoutedEventArgs e)

=> Click(Int32.Parse(((Button)sender).Tag.ToString()), "отклонен", false, true);

// Процесс установки значений одобрения/отклонения кредита

private void Click(int loanId, string message, bool isAccepted, bool isDecline)

{

var loan = \_db.Loans.Find(loanId);

\_db.Entry(loan).State = EntityState.Modified;

loan.IsAccepted = isAccepted;

loan.IsDecline = isDecline;

\_db.SaveChanges();

MessageBox.Show($"Кредит для {loan.Debtor.User.Surname} {loan.Debtor.User.Name} {loan.Debtor.User.Patronymic} {message}!");

UpdateList(\_db.Loans);

}

// Обновление списка заявок

private void UpdateList(IEnumerable<Loan> loans)

{

Loans.ItemsSource = loans.Where(l => new[] { l.IsDecline, l.IsAccepted, l.IsPaid }.All(p => p.Value == false)).Select(l => new

{

Id = l.ID,

DebtorSurname = l.Debtor.User.Surname,

DebtorName = l.Debtor.User.Name,

DebtorPatronymic = l.Debtor.User.Surname,

l.AdditionalConditions,

Bank = l.Bank.BankName,

l.InformationAboutLoan.LoansTerm,

l.InformationAboutLoan.LoansName,

l.InformationAboutLoan.InterestRate,

l.InformationAboutLoan.LifeInsurance

}).ToList();

}

// выход

private void Button\_Click\_2(object sender, RoutedEventArgs e)

=> NavigationService.Navigate(new AuthorizationPage());

// Фильтрация при редактировании ФИО

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

=> UpdateList(\_db.Loans.Where(l => (l.Debtor.User.Surname + " " + l.Debtor.User.Name + " " + l.Debtor.User.Patronymic).Contains(DebtorFN.Text) && (LoansN.SelectedValue.ToString() == "Все типы" ? true : l.InformationAboutLoan.LoansName == LoansN.SelectedValue.ToString())));

// Фильтрация при редактировании типа кредита

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

=> UpdateList(\_db.Loans.Where(l => (l.Debtor.User.Surname + " " + l.Debtor.User.Name + " " + l.Debtor.User.Patronymic).Contains(DebtorFN.Text) && (LoansN.SelectedValue.ToString() == "Все типы" ? true : l.InformationAboutLoan.LoansName == LoansN.SelectedValue.ToString())));

// Экспорт в txt

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

StringBuilder stringBuilder = new StringBuilder();

foreach (var l in Loans.ItemsSource)

stringBuilder.AppendLine(String.Join(" ", l.GetType().GetProperties().Select(x => x.GetValue(l))));

File.WriteAllText($"{Environment.GetFolderPath(Environment.SpecialFolder.Desktop)}/AllCredits.txt", stringBuilder.ToString());

MessageBox.Show($"Файл сохранен как {($"{Environment.GetFolderPath(Environment.SpecialFolder.Desktop)}/AllCredits.txt")}");

}

}

}

Код файла DebtorsMainPage.xaml представлен ниже.

<Page x:Class="\_220\_Smirnov.Assets.Pages.DebtorsMainPage"

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:\_220\_Smirnov.Assets.Pages"

mc:Ignorable="d" d:DesignHeight="450" d:DesignWidth="800"

Title="DebtorsMainPage">

<Grid Style="{DynamicResource GridStyle}" Margin="10">

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="0" Margin="5" Content="Подать заявку на кредит" Click="Button\_Click"/>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="1" Margin="5" Content="Просмотреть остатки по кредитной задолженности" Click="Button\_Click\_1"/>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="2" Margin="5" Content="Занести данные по взносу" Click="Button\_Click\_2"/>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="3" Margin="5" Content="Выйти" Click="Button\_Click\_3"/>

</Grid>

</Page>

Код файла DebtorsMainPage.xaml.cs представлен ниже.

using \_220\_Smirnov.Pages.Assets;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace \_220\_Smirnov.Assets.Pages

{

public partial class DebtorsMainPage : Page

{

private int \_debtorId;

public DebtorsMainPage(int debtorId)

{

InitializeComponent();

\_debtorId = debtorId;

}

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

=> NavigationService.Navigate(new ApplyingForLoanPage(\_debtorId));

private void Button\_Click\_1(object sender, RoutedEventArgs e)

=> NavigationService.Navigate(new ViewingLoanBalancesPage(\_debtorId));

private void Button\_Click\_2(object sender, RoutedEventArgs e)

=> NavigationService.Navigate(new EnterDataOnContributionPage(\_debtorId));

private void Button\_Click\_3(object sender, RoutedEventArgs e)

=> NavigationService.Navigate(new AuthorizationPage());

}

}

Код файла EnterDataOnContributionPage.xaml представлен ниже.

<Page x:Class="\_220\_Smirnov.Assets.Pages.EnterDataOnContributionPage"

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:\_220\_Smirnov.Assets.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="EnterDataOnContributionPage">

<Grid Style="{DynamicResource GridStyle}" Margin="10">

<Grid.RowDefinitions>

<RowDefinition Height="0.2\*"/>

<RowDefinition Height="0.7\*"/>

<RowDefinition Height="0.1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.8\*"/>

<ColumnDefinition Width="0.2\*"/>

</Grid.ColumnDefinitions>

<Grid Grid.Row="0" Grid.ColumnSpan="2">

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Viewbox Grid.Row="0" Grid.Column="0">

<TextBlock Text="Тип кредитования"/>

</Viewbox>

<ComboBox Name="LoansN" Grid.Column="0" Grid.Row="1" Style="{DynamicResource ComboBoxStyle}" SelectionChanged="LoansN\_SelectionChanged"/>

</Grid>

<DataGrid Style="{DynamicResource DataGridStyle}" Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2" Margin="5" AutoGenerateColumns="False" Name="Loans">

<DataGrid.Columns>

<DataGridTextColumn Header="Дата оформления кредитования" Binding="{Binding StartDate}"/>

<DataGridTextColumn Header="Дополнительные условия" Binding="{Binding AdditionalConditions}"/>

<DataGridTextColumn Header="Банк" Binding="{Binding Bank}"/>

<DataGridTextColumn Header="Срок кредитования" Binding="{Binding LoansTerm}"/>

<DataGridTextColumn Header="Тип кредитования" Binding="{Binding LoansName}"/>

<DataGridTextColumn Header="Процентная ставка" Binding="{Binding InterestRate}"/>

<DataGridTextColumn Header="Начальный взнос" Binding="{Binding DownPayment}"/>

<DataGridTextColumn Header="Месячный платеж" Binding="{Binding MonthlyPayment}"/>

<DataGridTextColumn Header="Тип недвижимости" Binding="{Binding EstateType}"/>

<DataGridTextColumn Header="Категория недвижимости" Binding="{Binding EstateCategory}"/>

<DataGridTextColumn Header="Сумма кредитования" Binding="{Binding Price}"/>

<DataGridTextColumn Header="Адрес" Binding="{Binding Address}"/>

<DataGridTextColumn Header="Выплаченная сумма" Binding="{Binding Paid}"/>

<DataGridTextColumn Header="Страхование жизни" Binding="{Binding LifeInsurance}"/>

<DataGridTemplateColumn>

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Style="{DynamicResource ButtonStyle}" Content="Добавить взнос" Click="Button\_Click\_1" Tag="{Binding Id}"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="2" Grid.Column="1" Margin="5" Content="Назад" Click="Button\_Click"/>

</Grid>

</Page>

Код файла EnterDataOnContributionPage.xaml.cs представлен ниже.

using \_220\_Smirnov.Database;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace \_220\_Smirnov.Assets.Pages

{

public partial class EnterDataOnContributionPage : Page

{

private LendingEntities \_db = new LendingEntities();

private int \_debtorId;

// Установка начальный значений

public EnterDataOnContributionPage(int debtorId)

{

InitializeComponent();

\_debtorId = debtorId;

UpdateList(\_db.Loans);

}

~EnterDataOnContributionPage()

{

\_db.Dispose();

}

// переход назад

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

=> NavigationService.GoBack();

// месячный взнос

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

try

{

int loanId = Int32.Parse(((Button)sender).Tag.ToString());

Loan loan = \_db.Loans.Find(loanId);

RealEstateObject reo = loan.RealEstateObject;

\_db.Entry(reo).State = EntityState.Modified;

reo.Paid += loan.InformationAboutLoan.MonthlyPayment;

if (reo.Paid >= reo.Price)

loan.IsPaid = true;

\_db.SaveChanges();

UpdateList(\_db.Loans);

MessageBox.Show("Месячный платеж внесен!");

if (loan.IsPaid.Value)

MessageBox.Show("Задолженность погашена!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

// Обновление списка заявок

private void UpdateList(IEnumerable<Loan> loans)

{

Loans.ItemsSource = loans.Where(l => l.DebtorID == \_debtorId && l.IsAccepted == true && l.IsPaid == false).Select(l => new

{

Id = l.ID,

l.StartDate,

l.InformationAboutLoan.DownPayment,

l.InformationAboutLoan.MonthlyPayment,

l.RealEstateObject.RealEstateType.EstateType,

l.RealEstateObject.RealEstateCategory.EstateCategory,

l.RealEstateObject.Price,

l.RealEstateObject.Address,

l.RealEstateObject.Paid,

l.AdditionalConditions,

Bank = l.Bank.BankName,

l.InformationAboutLoan.LoansTerm,

l.InformationAboutLoan.LoansName,

l.InformationAboutLoan.InterestRate,

l.InformationAboutLoan.LifeInsurance,

}).ToList();

}

// Фильтрация при изменении типа кредитования

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

=> UpdateList(\_db.Loans.Where(l => LoansN.SelectedValue.ToString() == "Все типы" ? true : l.InformationAboutLoan.LoansName == LoansN.SelectedValue.ToString()));

}

}

Код файла ViewingLoanBalancesPage.xaml представлен ниже.

<Page x:Class="\_220\_Smirnov.Assets.Pages.ViewingLoanBalancesPage"

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:\_220\_Smirnov.Assets.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="ViewingLoanBalancesPage">

<Grid Style="{DynamicResource GridStyle}" Margin="10">

<Grid.RowDefinitions>

<RowDefinition Height="0.9\*"/>

<RowDefinition Height="0.1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.8\*"/>

<ColumnDefinition Width="0.2\*"/>

</Grid.ColumnDefinitions>

<DataGrid Style="{DynamicResource DataGridStyle}" Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2" Margin="5" AutoGenerateColumns="False" Name="Loans">

<DataGrid.Columns>

<DataGridTextColumn Header="Стартовый платеж" Binding="{Binding StartDate}"/>

<DataGridTextColumn Header="Дополнительные условия" Binding="{Binding AdditionalConditions}"/>

<DataGridTextColumn Header="Банк" Binding="{Binding Bank}"/>

<DataGridTextColumn Header="Срок кредитования" Binding="{Binding LoansTerm}"/>

<DataGridTextColumn Header="Тип кредитования" Binding="{Binding LoansName}"/>

<DataGridTextColumn Header="Процентная ставка" Binding="{Binding InterestRate}"/>

<DataGridTextColumn Header="Начальный взнос" Binding="{Binding DownPayment}"/>

<DataGridTextColumn Header="Месячный платеж" Binding="{Binding MonthlyPayment}"/>

<DataGridTextColumn Header="Тип недвижимости" Binding="{Binding EstateType}"/>

<DataGridTextColumn Header="Категория недвижимости" Binding="{Binding EstateCategory}"/>

<DataGridTextColumn Header="Сумма кредитования" Binding="{Binding Price}"/>

<DataGridTextColumn Header="Адрес" Binding="{Binding Address}"/>

<DataGridTextColumn Header="Выплаченная сумма" Binding="{Binding Paid}"/>

<DataGridTextColumn Header="Страхование жизни" Binding="{Binding LifeInsurance}"/>

</DataGrid.Columns>

</DataGrid>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="1" Grid.Column="1" Margin="5" Content="Назад" Click="Button\_Click"/>

</Grid>

</Page>

Код файла ViewingLoanBalancesPage.xaml.cs представлен ниже.

using \_220\_Smirnov.Database;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace \_220\_Smirnov.Assets.Pages

{

public partial class ViewingLoanBalancesPage : Page

{

private LendingEntities \_db = new LendingEntities();

private int \_debtorId;

public ViewingLoanBalancesPage(int debtorId)

{

InitializeComponent();

\_debtorId = debtorId;

UpdateList();

}

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

=> NavigationService.GoBack();

private void UpdateList()

{

Loans.ItemsSource = \_db.Loans.Where(l => l.DebtorID == \_debtorId && l.IsAccepted == true && l.IsPaid == false).Select(l => new

{

l.StartDate,

l.InformationAboutLoan.DownPayment,

l.InformationAboutLoan.MonthlyPayment,

l.RealEstateObject.RealEstateType.EstateType,

l.RealEstateObject.RealEstateCategory.EstateCategory,

l.RealEstateObject.Price,

l.RealEstateObject.Address,

l.RealEstateObject.Paid,

l.AdditionalConditions,

Bank = l.Bank.BankName,

l.InformationAboutLoan.LoansTerm,

l.InformationAboutLoan.LoansName,

l.InformationAboutLoan.InterestRate,

l.InformationAboutLoan.LifeInsurance,

}).ToList();

}

}

}

Код файла App.xaml представлен ниже.

<Application x:Class="\_220\_Smirnov.App"

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

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

xmlns:local="clr-namespace:\_220\_Smirnov"

StartupUri="MainWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="Assets\Styles\Elements\Border.xaml"/>

<ResourceDictionary Source="Assets\Styles\Elements\Button.xaml"/>

<ResourceDictionary Source="Assets\Styles\Elements\ComboBox.xaml"/>

<ResourceDictionary Source="Assets\Styles\Elements\DataGrid.xaml"/>

<ResourceDictionary Source="Assets\Styles\Elements\Grid.xaml"/>

<ResourceDictionary Source="Assets\Styles\Elements\PasswordBox.xaml"/>

<ResourceDictionary Source="Assets\Styles\Elements\TextBlock.xaml"/>

<ResourceDictionary Source="Assets\Styles\Elements\TextBox.xaml"/>

<ResourceDictionary Source="Assets\Styles\Colors\Colors.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

Код файла MainWindow.xaml представлен ниже.

<Window x:Class="\_220\_Smirnov.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:\_220\_Smirnov" Background="{DynamicResource MainBackground}"

mc:Ignorable="d" Title="Калькулятор ипотечного кредитования" Height="450" Width="800">

<Frame NavigationUIVisibility="Hidden" Name="MainFrame"/>

</Window>

Код файла MainWindow.xaml представлен ниже.

using \_220\_Smirnov.Assets.Pages;

using System.Windows;

namespace \_220\_Smirnov

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame.Navigate(new AuthorizationPage());

}

}

}