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-220

Садовникова Евгения Денисовна

Проверили:

преподаватели

\_\_\_\_\_\_\_\_ / Т.Г. Аксёнова /

\_\_\_\_\_\_\_\_/О.А. Тимакин/

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

2023

Содержание

[Проектирование программного приложения 3](#_Toc154594064)

[Руководство оператора 6](#_Toc154594065)

[Назначение программы 6](#_Toc154594066)

[Условия выполнения программы 6](#_Toc154594067)

[Выполнение программы 6](#_Toc154594068)

[Сообщения оператору 8](#_Toc154594069)

[Листинг программных модулей 11](#_Toc154594070)

# Проектирование программного приложения

Для демонстрации прав и ролей в приложении «калькулятор ипотечного кредитования» создана диаграмма вариантов использования, которая представлена на рисунке 1.

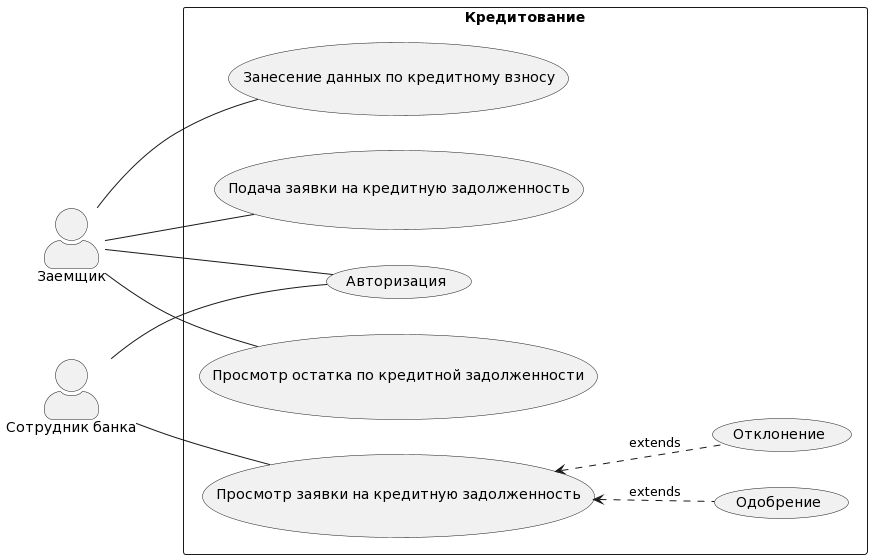


Рисунок 1. Диаграмма вариантов использования

Для отображения связей в базе данных представлена диаграмма сущность связь на рисунке 2.

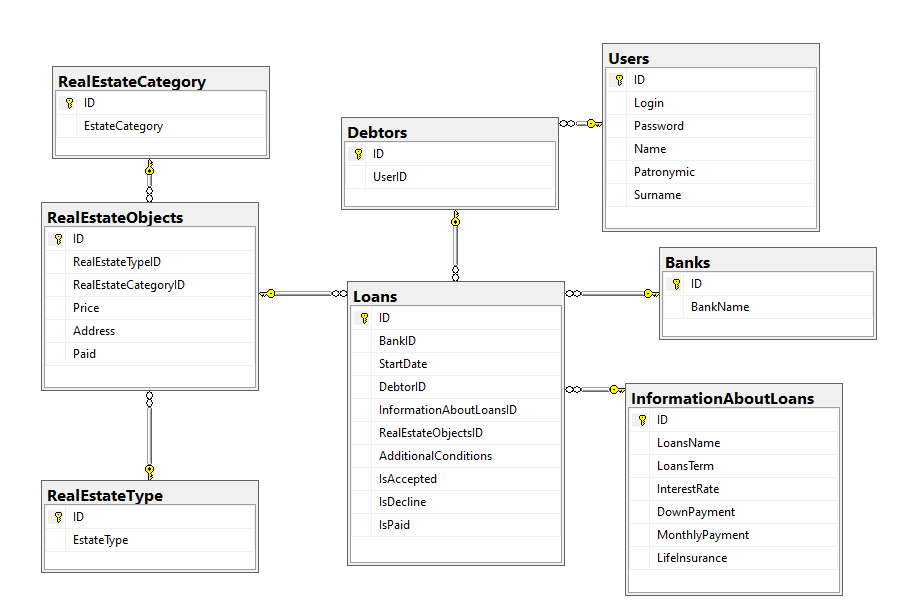


Рисунок 2. Диаграмма сущность связь

Наполнение таблиц базы данных представлено на рисунках 7 – 11

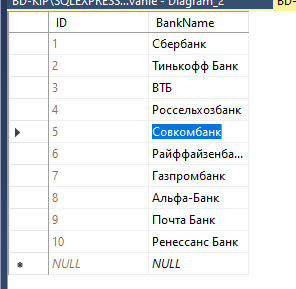


Рисунок 7. Таблица Banks

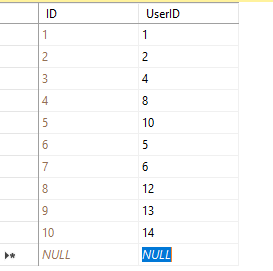


Рисунок 8. Таблица Debtors

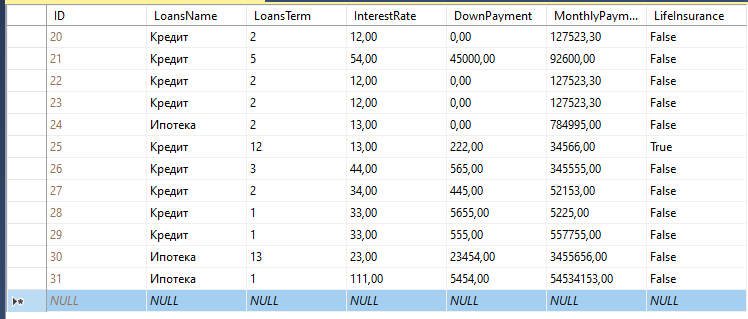


Рисунок 9. Таблица InformationAboutLoans

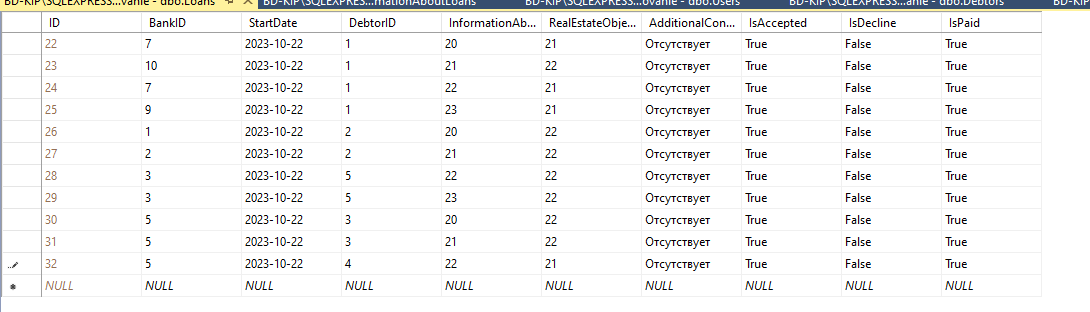


Рисунок 10. Таблица Loans

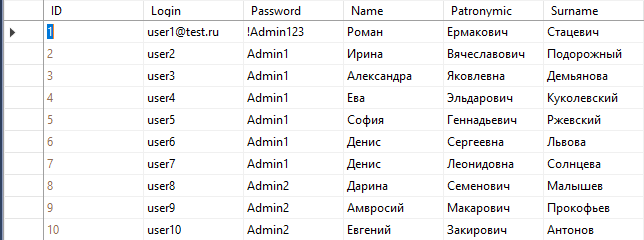


Рисунок 11. Таблица Users

# Руководство оператора

## Назначение программы

Программа предназначена для таких людей как: заемщик, сотрудник банка. Заемщик может подать заявку на кредит и ипотеку, заносить взносы и смотреть сколько он еще должен банку. Сотруднику банка предоставлена возможность одобрения и отклонения кредита.

## Условия выполнения программы

Для выполнения программы необходим персональный компьютер на базе операционной системы Windows 10/11, процессов i3 10100 и выше, объем оперативной памяти 8 гб и выше, наличие 5 гб свободного места на жестком диске, установленная среда разработки Visual Studio.

## Выполнение программы

При запуске программы пользователь попадает на начальную страницу авторизации, как показано на рисунке 11 ниже.

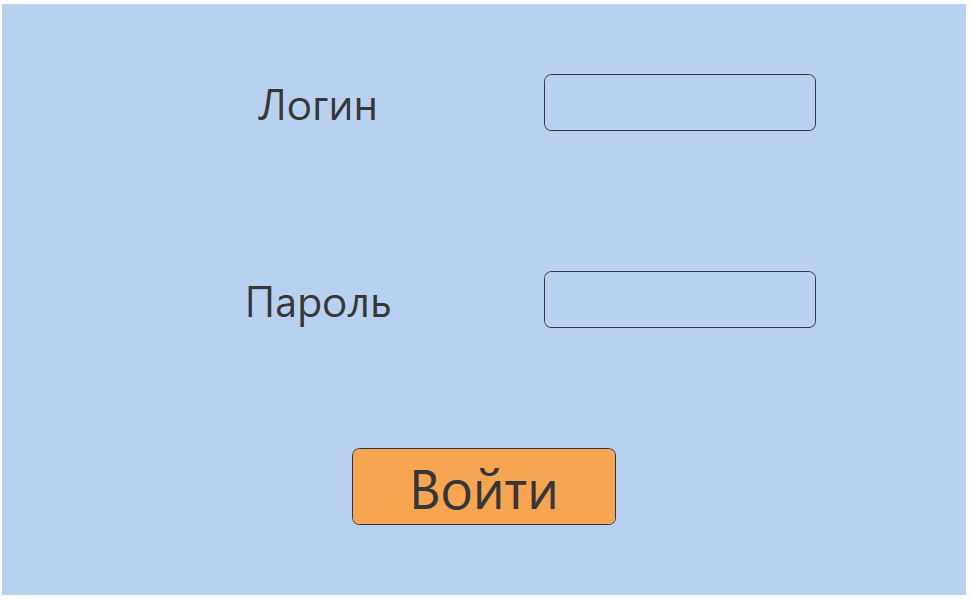


Рисунок 11. Начальная страница приложения

На странице авторизации, пользователь вводит свои данные, как показано на рисунке 12 ниже.

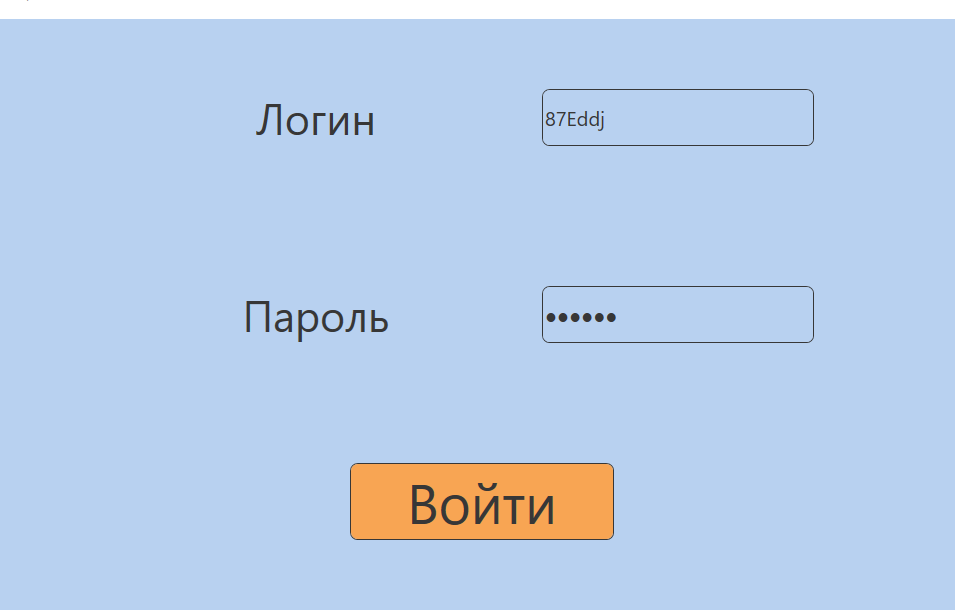


Рисунок 12. Ввод данных для авторизации

После ввода данных необходимо нажать на кнопку «Войти».

Далее пользователя переадресует на страницу меню, как показано ниже, на рисунке 14.

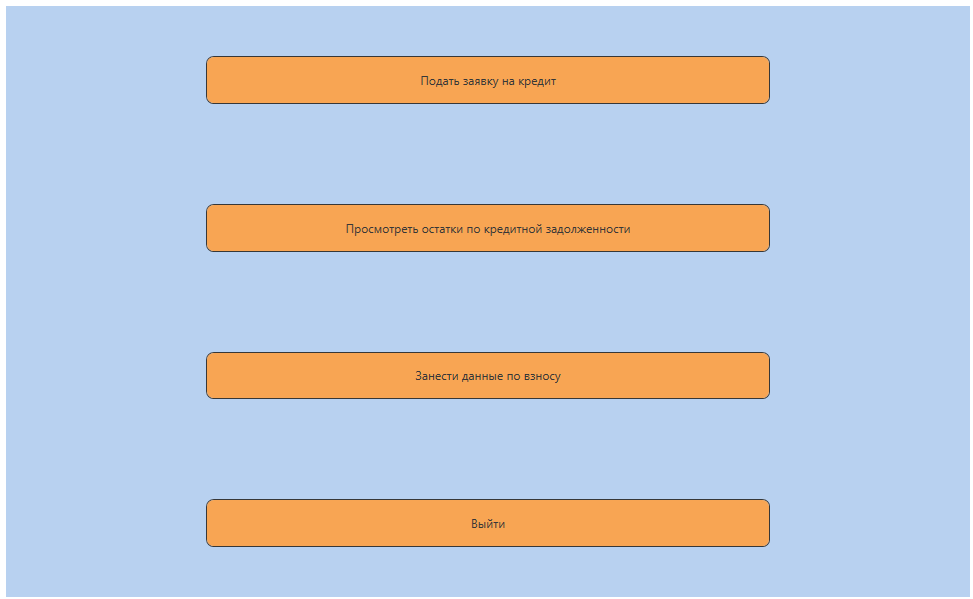


Рисунок 14. Страница меню

После выбора «Подать заявку на кредит» пользователь попадает на страницу, изображенную на рисунке 15.

Чтобы вернуться обратно в меню нужно нажать «выйти».

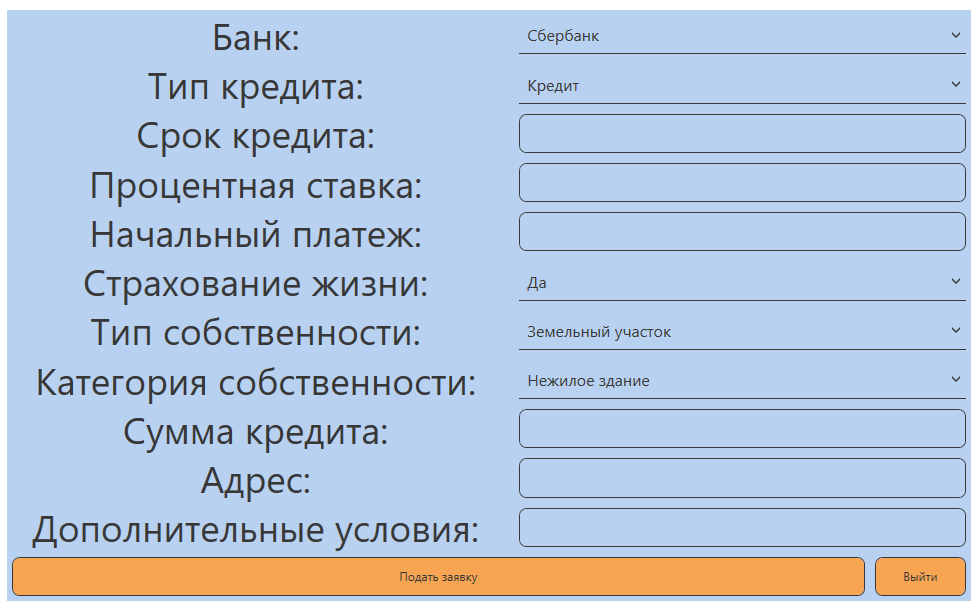


Рисунок 15. Страница подать заявку на кредит

Когда в систему входит сотрудник банка он попадает на страницу, изображенную на рисунке 16.

Чтобы вернуться на страницу авторизации нужно нажать «выйти».

Для одобрения кредита нужно нажать «одобрить», для отклонения «отклонить».

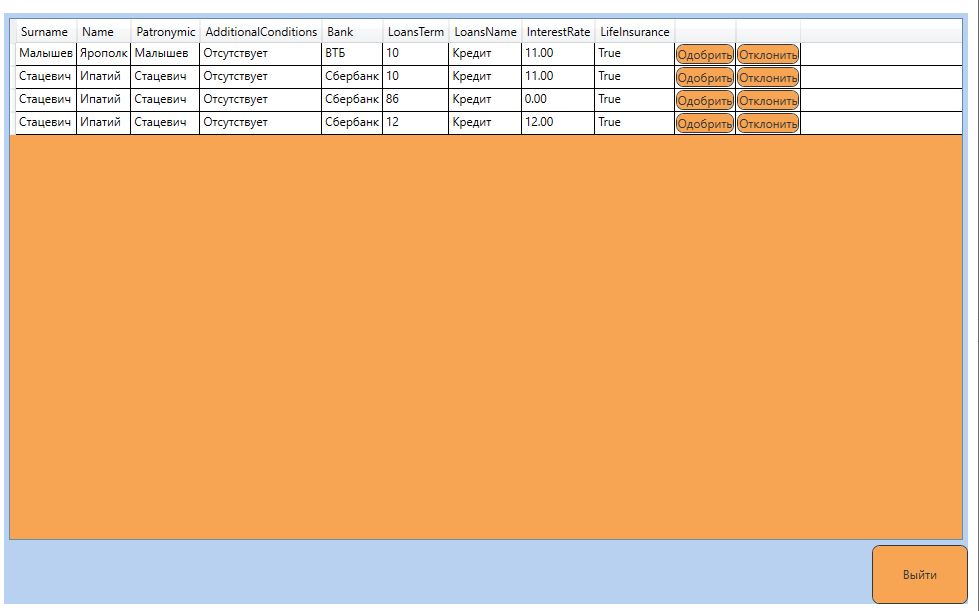


Рисунок 16. Страница сотрудника банка

## Сообщения оператору

При успешной авторизации появляется сообщение, представленное на рисунке 3.

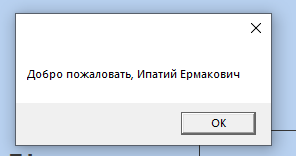


Рисунок 3. Сообщение о успешной авторизации

При успешной заполненной заявки на кредит появляется сообщение, представленное на рисунке 4.

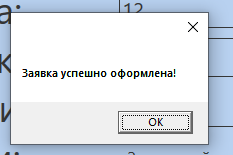


Рисунок 4. Сообщение о успешной заявки на кредит

При ошибочном заполнении данных для заявки на кредит появляется сообщение, представленное на рисунке 5.

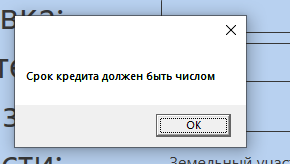


Рисунок 5. Сообщение ошибки

Когда появляется сообщение на рисунке 5, необходимо проверить правильность ввода срока кредита.

При неудачной авторизации появляется сообщение, представленное на рисунке 6.

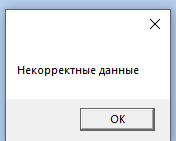


Рисунок 6. Сообщение ошибки

Когда появляется сообщение на рисунке 6, необходимо проверить правильность ввода логина и пароля.

# Листинг программных модулей

Листинг кода ApplyingForLoanPage.xaml:

<Page x:Class="220\_Sadovnikova.Pages.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\_Sadovnikova.Pages"

mc:Ignorable="d"

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

Title="ApplyingForLoan">

<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>

Листинг кода ApplyingForLoanPage.xaml.cs:

using 220\_Sadovnikova.Database;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace 220\_Sadovnikova.Pages

{

public partial class ApplyingForLoanPage : Page

{

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

{

["Да"] = true,

["Нет"] = false

};

private CreditsEntities \_db = new CreditsEntities();

private int \_debtorId;

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

{

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");

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

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);

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);

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);

}

}

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

=> NavigationService.GoBack();

}

}

Листинг кода AuthorizationPage.xaml:

<Page x:Class="220\_Sadovnikova.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\_Sadovnikova.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" Margin="150 70 0 70">

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

</Viewbox>

<TextBox Style="{DynamicResource TextBoxStyle}" x:Name="Login" Grid.Row="0" Grid.Column="1" Margin="60 70 150 70"/>

<Viewbox Grid.Row="1" Grid.Column="0" Margin="150 70 0 70">

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

</Viewbox>

<PasswordBox Style="{DynamicResource PasswordBoxStyle}" x:Name="Password" Grid.Row="1" Grid.Column="1" Margin="60 70 150 70"/>

<Button Style="{DynamicResource ButtonStyle}" Grid.Row="2" Grid.Column="0" Grid.ColumnSpan="2" Margin="350 50 350 70" Click="Button\_Click">

<Button.Content>

<Viewbox>

<TextBlock Text="Войти"/>

</Viewbox>

</Button.Content>

</Button>

</Grid>

</Page>

Листинг кода AuthorizationPage.xaml.cs:

using 220\_Sadovnikova.Database;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Navigation;

namespace PR30.Pages

{

public partial class AuthorizationPage : Page

{

private CreditsEntities \_db = new CreditsEntities();

public AuthorizationPage()

=> InitializeComponent();

~AuthorizationPage()

{

\_db.Dispose();

}

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

{

try

{

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($"Добро пожаловать, {user.Name} {user.Patronymic}");

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

if (debtor != null)

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

else

NavigationService.Navigate(new MainPageForBankEmployeer());

}

catch

{

MessageBox.Show("Некорректные данные");

}

}

}

}

Листинг кода EnterDataOnContributionPage.xaml:

<Page x:Class=" 220\_Sadovnikova.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\_Sadovnikova.Pages"

mc:Ignorable="d"

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

Title="EnterDataOnContributionPage">

<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="StartDate" Binding="{Binding StartDate}"/>

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

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

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

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

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

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

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

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

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

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

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

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

<DataGridTextColumn Header="LifeInsurance" 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="1" Grid.Column="1" Margin="5" Content="Назад" Click="Button\_Click"/>

</Grid>

</Page>

Листинг кода EnterDataOnContributionPage.xaml.cs:

using 220\_Sadovnikova.Database;

using System;

using System.Data.Entity;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace 220\_Sadovnikova.Pages

{

public partial class EnterDataOnContributionPage : Page

{

private CreditsEntities \_db = new CreditsEntities();

private int \_debtorId;

public EnterDataOnContributionPage(int debtorId)

{

InitializeComponent();

\_debtorId = debtorId;

UpdateList();

}

~EnterDataOnContributionPage()

{

\_db.Dispose();

}

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

=> NavigationService.GoBack();

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

{

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

try

{

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();

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

if (loan.IsPaid.Value)

MessageBox.Show("Задолженность погашена!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void UpdateList()

{

Loans.ItemsSource = \_db.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();

}

}

}

Листинг кода MainPageForBankEmployeer.xaml:

<Page x:Class=" 220\_Sadovnikova.Pages.MainPageForBankEmployeer"

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\_Sadovnikova.Pages"

mc:Ignorable="d"

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

Title="MainPageForBankEmployeer">

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

<Grid.RowDefinitions>

<RowDefinition Height="0.9\*"/>

<RowDefinition Height="0.1\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0.9\*"/>

<ColumnDefinition Width="0.1\*"/>

</Grid.ColumnDefinitions>

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

<DataGrid.Columns>

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

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

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

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

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

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

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

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

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

</Grid>

</Page>

Листинг кода MainPageForBankEmployeer.xaml.сs:

using 220\_Sadovnikova.Database;

using System;

using System.Data.Entity;

using System.Linq;

using System.Windows;

using System.Windows.Controls;

namespace 220\_Sadovnikova.Pages

{

public partial class MainPageForBankEmployeer : Page

{

private CreditsEntities \_db = new CreditsEntities();

public MainPageForBankEmployeer()

{

InitializeComponent();

UpdateList();

}

~MainPageForBankEmployeer()

{

\_db.Dispose();

}

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

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

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

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

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();

}

private void UpdateList()

{

Loans.ItemsSource = \_db.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();

}

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

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

}

}

Листинг кода MainPageForDebtors.xaml:

<Page x:Class=" 220\_Sadovnikova.Pages.MainPageForDebtors"

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\_Sadovnikova..Pages"

mc:Ignorable="d"

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

Title="MainPageForDebtors">

<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>

Листинг кода MainPageForDebtors.xaml.cs:

using System.Windows;

using System.Windows.Controls;

namespace 220\_Sadovnikova..Pages

{

public partial class MainPageForDebtors : Page

{

private int \_debtorId;

public MainPageForDebtors(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());

}

}