

**Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)**

Колледж информатики и программирования

ОТЧЁТ
По учебной практике

Специальность 09.02.07 «Информационные системы и программирование»
(код) (наименование)

Профессиональный модуль ПМ.01 Разработка модулей программного обеспечения для компьютерных систем
(код) (наименование)

Междисциплинарный курс МДК.01.02 Поддержка и тестирование программных модулей
(код) (наименование)

Выполнил:

Студент 4 курса 4ИСИП-422 учебной группы
(номер) (номер)

(подпись) М.О.Джабаров
(инициалы, фамилия)

Проверил:

Руководитель практики от Колледжа
информатики и программирования
Преподаватель ВКК, к.п.н. Т.Г. Аксёнова
(квалификационная (инициалы, фамилия)
категория или звание,
должность)

Москва – 2025

Перечень работ, выполненных в ходе учебной практики

| № п/п | Виды работ | Оценка |
|--------------|---|---------------|
| 1 | Разработка программного модуля многооконного интерфейса | Отлично |
| 2 | Тестирование программного модуля доступа к БД | Отлично |

Ссылка на git

https://github.com/Mexti1/Djabarov_Payment/tree/Finish

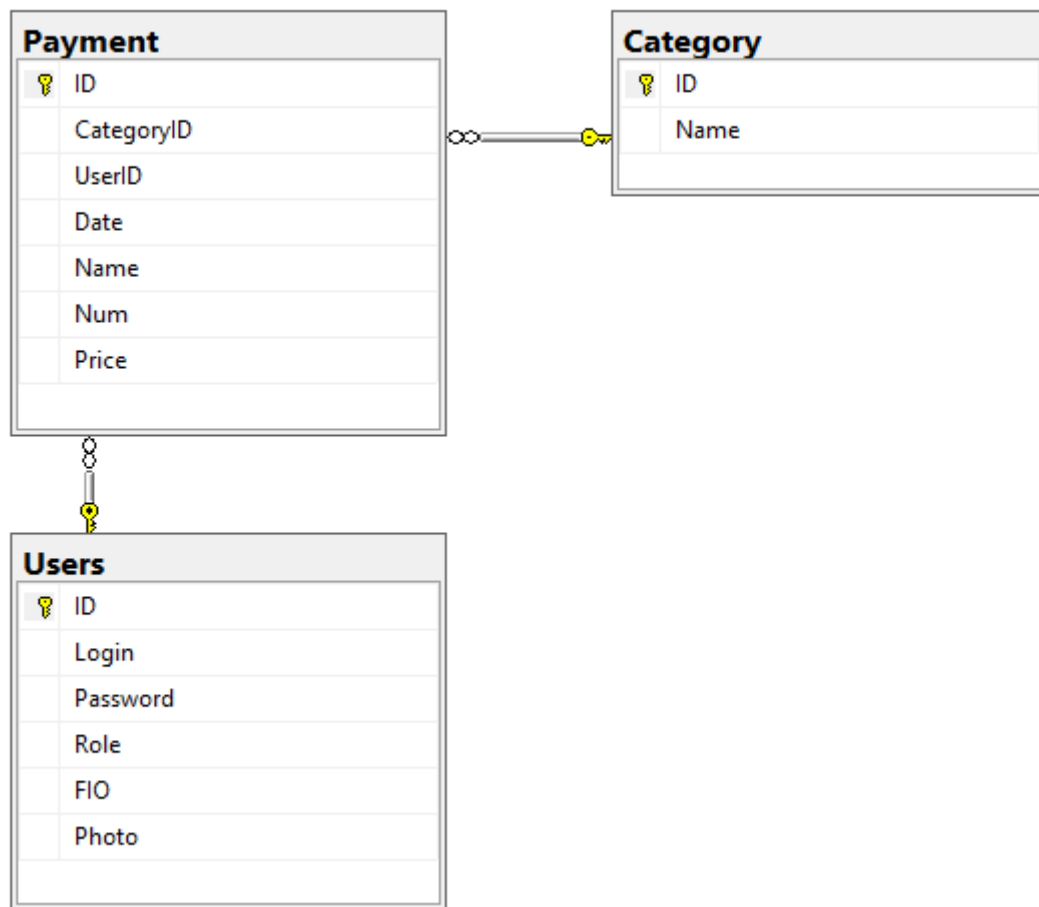


Рисунок 1

Листинг №1 – Скрипт БД

```
USE [master]
GO
/***** Object:  Database [Djabarov_DB_Payment]      Script Date: 10.11.2025
2:34:37 *****/
CREATE DATABASE [Djabarov_DB_Payment]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'Djabarov_DB_Payment', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\Djabarov_DB_Payment.mdf' , SIZE =
8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
    LOG ON
    ( NAME = N'Djabarov_DB_Payment_log', FILENAME = N'C:\Program Files\Microsoft
SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\Djabarov_DB_Payment_log.ldf' ,
SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
    WITH CATALOG_COLLATION = DATABASE_DEFAULT, LEDGER = OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET COMPATIBILITY_LEVEL = 160
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [Djabarov_DB_Payment].[dbo].[sp_fulltext_database] @action = 'enable'
end
```

```

GO
ALTER DATABASE [Djabarov_DB_Payment] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET ANSI_NULLS OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET ANSI_PADDING OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET ARITHABORT OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [Djabarov_DB_Payment] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [Djabarov_DB_Payment] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET  ENABLE_BROKER
GO
ALTER DATABASE [Djabarov_DB_Payment] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [Djabarov_DB_Payment] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET RECOVERY FULL
GO
ALTER DATABASE [Djabarov_DB_Payment] SET  MULTI_USER
GO
ALTER DATABASE [Djabarov_DB_Payment] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [Djabarov_DB_Payment] SET DB_CHAINING OFF
GO
ALTER DATABASE [Djabarov_DB_Payment] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [Djabarov_DB_Payment] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [Djabarov_DB_Payment] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [Djabarov_DB_Payment] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO
EXEC sys.sp_db_vardecimal_storage_format N'Djabarov_DB_Payment', N'ON'
GO

```

```

ALTER DATABASE [Djabarov_DB_Payment] SET QUERY_STORE = ON
GO
ALTER DATABASE [Djabarov_DB_Payment] SET QUERY_STORE (OPERATION_MODE =
READ_WRITE, CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 30),
DATA_FLUSH_INTERVAL_SECONDS = 900, INTERVAL_LENGTH_MINUTES = 60,
MAX_STORAGE_SIZE_MB = 1000, QUERY_CAPTURE_MODE = AUTO,
SIZE_BASED_CLEANUP_MODE = AUTO, MAX_PLANS_PER_QUERY = 200,
WAIT_STATS_CAPTURE_MODE = ON)
GO
USE [Djabarov_DB_Payment]
GO
/***** Object:  Table [dbo].[Category]      Script Date: 10.11.2025 2:34:38
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Category](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](200) NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY =
OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object:  Table [dbo].[Payment]      Script Date: 10.11.2025 2:34:38
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Payment](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [CategoryID] [int] NOT NULL,
    [UserID] [int] NOT NULL,
    [Date] [datetime] NOT NULL,
    [Name] [nvarchar](200) NOT NULL,
    [Num] [int] NOT NULL,
    [Price] [decimal](18, 2) NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY =
OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object:  Table [dbo].[Users]      Script Date: 10.11.2025 2:34:38
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Users](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Login] [nvarchar](100) NOT NULL,
    [Password] [nvarchar](255) NOT NULL,
    [Role] [nvarchar](50) NOT NULL,
    [FIO] [nvarchar](200) NULL,
    [Photo] [nvarchar](500) NULL,
    PRIMARY KEY CLUSTERED

```

```

(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY =
OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Payment] WITH CHECK ADD CONSTRAINT
[FK_Payment_Category] FOREIGN KEY([CategoryID])
REFERENCES [dbo].[Category] ([ID])
GO
ALTER TABLE [dbo].[Payment] CHECK CONSTRAINT [FK_Payment_Category]
GO
ALTER TABLE [dbo].[Payment] WITH CHECK ADD CONSTRAINT [FK_Payment_User]
FOREIGN KEY([UserID])
REFERENCES [dbo].[Users] ([ID])
GO
ALTER TABLE [dbo].[Payment] CHECK CONSTRAINT [FK_Payment_User]
GO
USE [master]
GO
ALTER DATABASE [Djabarov_DB_Payment] SET READ_WRITE
GO

```

| | Имя столбца | Тип данных | Разрешить ... |
|----|-------------|---------------|--------------------------|
| PK | ID | int | <input type="checkbox"/> |
| | Name | nvarchar(200) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 2

| | Имя столбца | Тип данных | Разрешить ... |
|----|-------------|----------------|--------------------------|
| PK | | int | <input type="checkbox"/> |
| | CategoryID | int | <input type="checkbox"/> |
| | UserID | int | <input type="checkbox"/> |
| | Date | datetime | <input type="checkbox"/> |
| | Name | nvarchar(200) | <input type="checkbox"/> |
| | Num | int | <input type="checkbox"/> |
| | Price | decimal(18, 2) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3

| | Имя столбца | Тип данных | Разрешить ... |
|----|-------------|---------------|-------------------------------------|
| PK | ID | int | <input type="checkbox"/> |
| | Login | nvarchar(100) | <input type="checkbox"/> |
| | Password | nvarchar(255) | <input type="checkbox"/> |
| | Role | nvarchar(50) | <input type="checkbox"/> |
| | FIO | nvarchar(200) | <input checked="" type="checkbox"/> |
| | Photo | nvarchar(500) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 4


| Data Dictionary | | | | |
|-----------------|------------|------------------------|------------|-------------------------|
| Category | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED ? | NOTES |
| PK | ID | INT | Y | Auto increment |
| | Name | VARCHAR(MAX) | Y | |
| Payment | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED ? | NOTES |
| PK | ID | INT | Y | Auto increment |
| FK | UserID | INT | Y | Foreign Key to User |
| FK | CategoryID | INT | Y | Foreign Key to Category |
| | Date | DATE | Y | |
| | Name | NVARCHAR(50) | Y | |
| | Num | DECIMAL(18,0) | Y | |
| | Price | DECIMAL(18,0) | Y | |
| User | | | | |
| KEY | FIELD NAME | DATA TYPE / FIELD SIZE | REQUIRED ? | NOTES |
| PK | ID | INT | Y | Auto increment |
| | Login | VARCHAR(MAX) | Y | |
| | Password | VARCHAR(MAX) | Y | |
| | Role | VARCHAR(50) | N | |
| | FIO | VARCHAR(MAX) | N | |
| | Photo | NVARCHAR(MAX) | N | |

Рисунок 5

Система учета платежей - Авторизация

← Назад

Система учета платежей



Логин

Введите логин

Пароль

Введите пароль

Вход

Регистрация

Смена пароля

10.11.2025 00:04:05

Рисунок 6

Система учета платежей

Логин

Введите логин

Пароль

Введите пароль

Повторите пароль

Повторите пароль


Роль

User

ФИО

Введите ФИО

Введите текст с картинки:



Обновить

Введите текст с картинки

Рисунок 7

Система учета платежей

Пароль

Повторите пароль

Роль

Успех

Пользователь успешно зарегистрирован!

ОК

Введите текст с картинки.

GUPVV

Обновить

GUPVV

Зарегистрироваться

Рисунок 8

Смена пароля

Логин

111

Текущий пароль

●●●●

Новый

●●●●

Подтверждение нового пароля

●●●●●●

Сменить пароль

Успех



Пароль успешно изменен!

OK

Рисунок 9

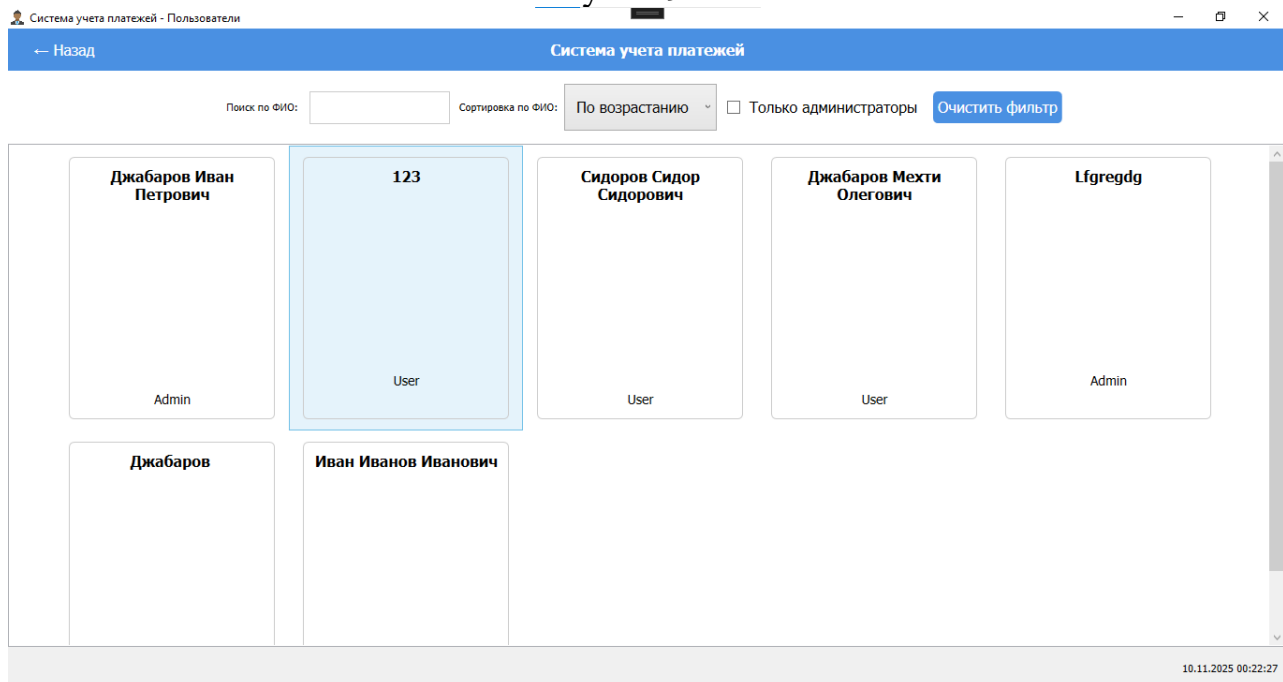


Рисунок 10

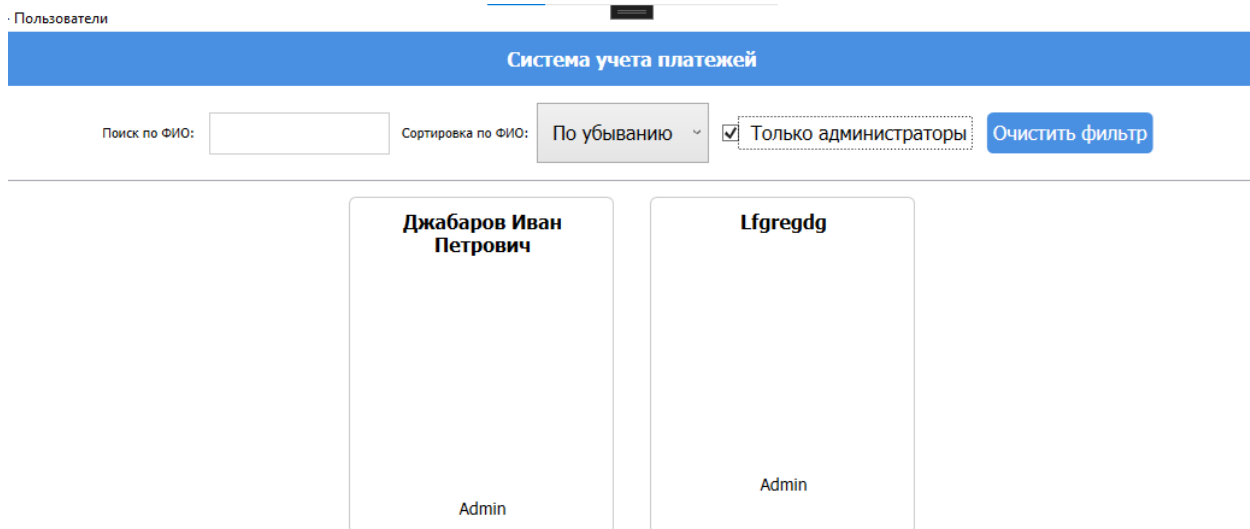


Рисунок 11

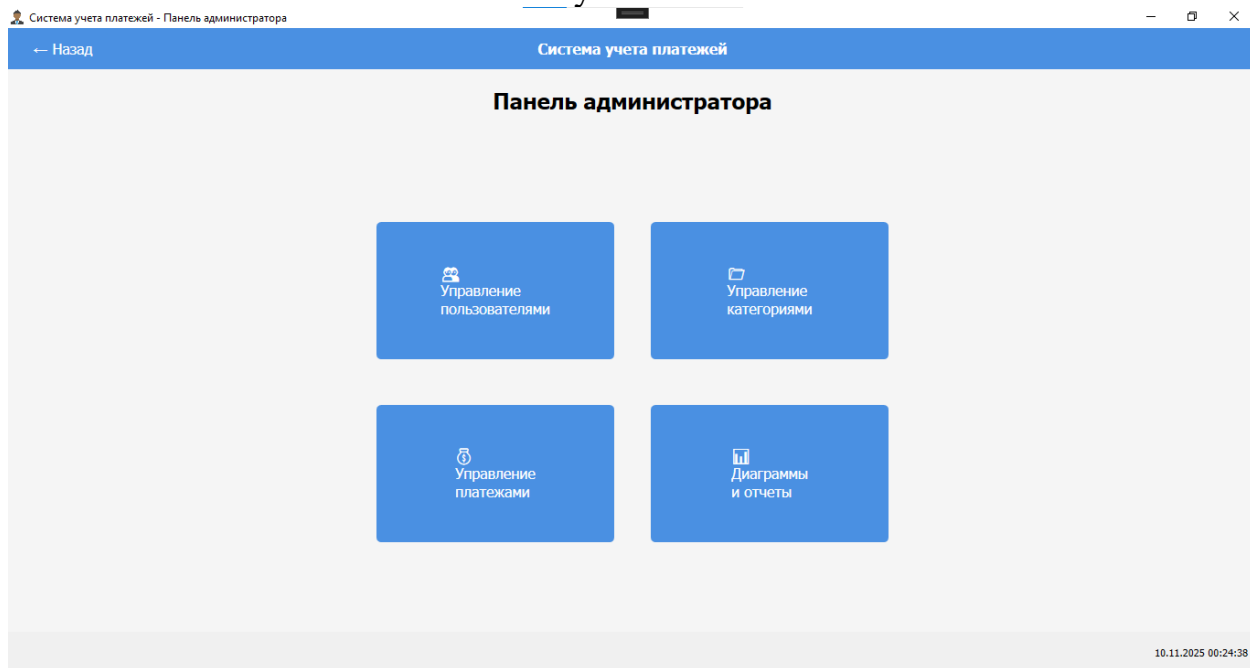


Рисунок 12

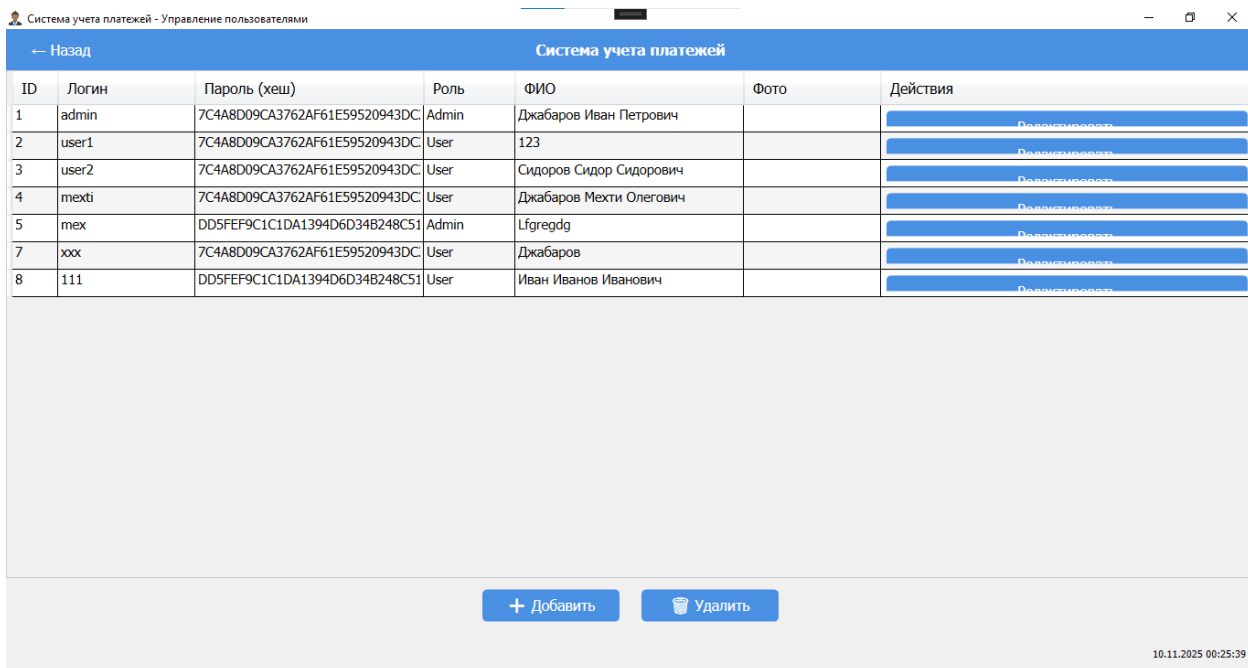


Рисунок 13

Система учета платежей

Добавление/Редактирование пользователя

Логин

Пароль

Роль

ФИО

Фото (путь к файлу, необязательно)

Сохранить Очистить

Рисунок 14

Система учета платежей

Добавление/Редактирование пользователя

Логин

Пароль

Роль

ФИО

Фото (путь к файлу, необязательно)



Сохранить



Очистить

Рисунок 15

Система учета платежей

Добавление/Редактирование пользователя

Логин

Пароль

Роль

ФИО

Фото (путь к файлу, необязательно)



Сохранить



Очистить

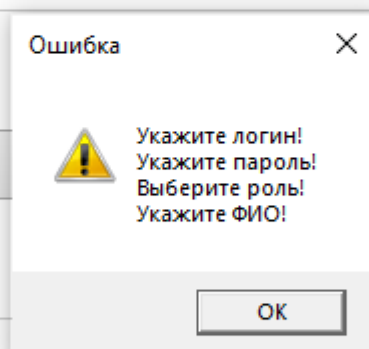


Рисунок 16

Система учета платежей

Добавление/Редактирование пользователя

Логин

123

Пароль

123456

Роль

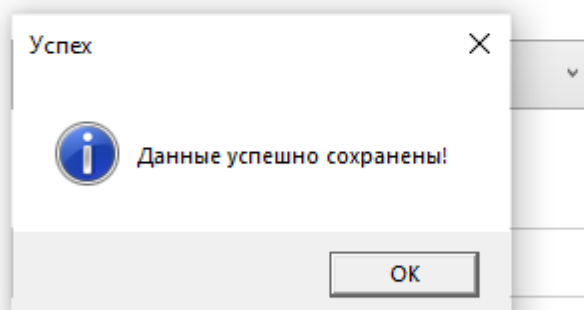


Фото (путь к файлу, необязательно)

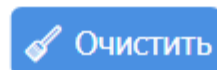
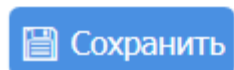


Рисунок 17

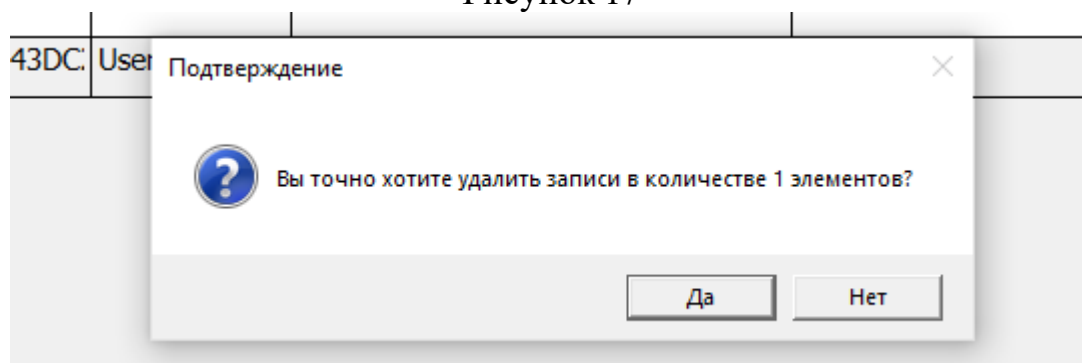


Рисунок 18

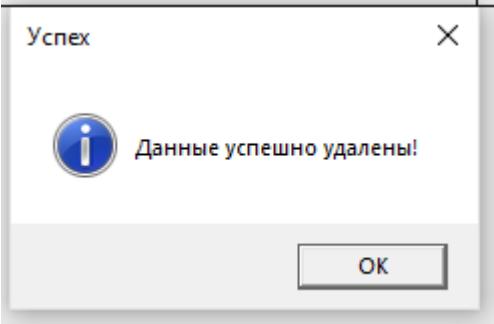


Рисунок 19

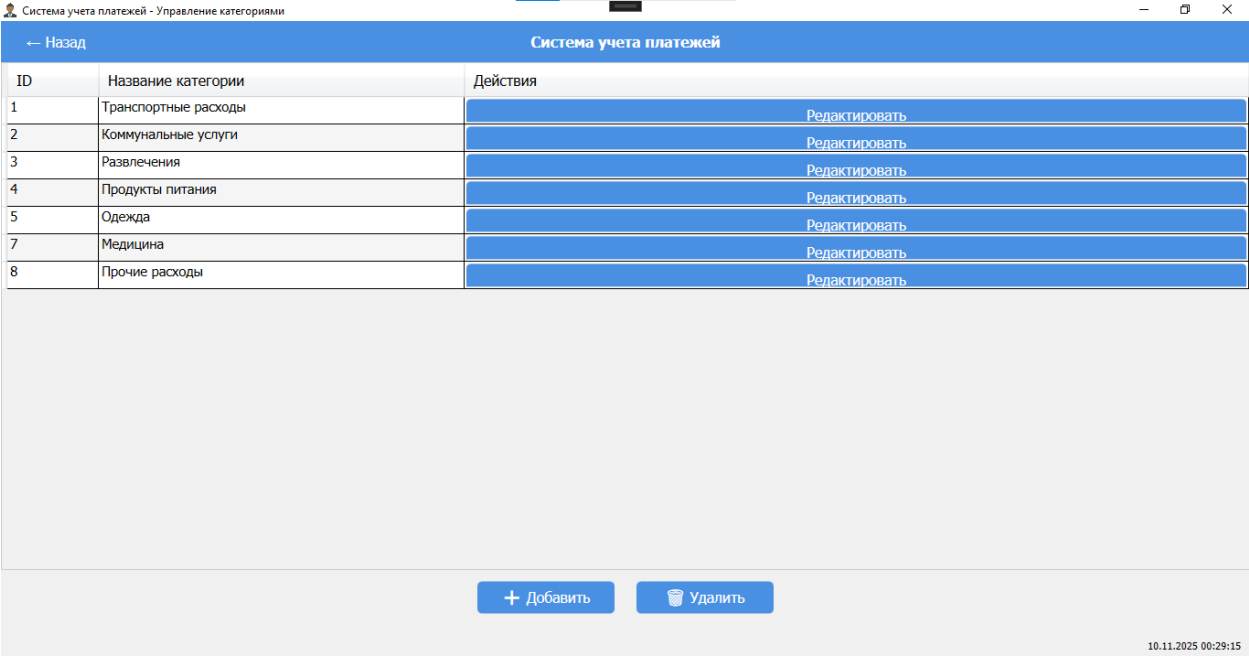


Рисунок 20

Добавление/Редактирование категории

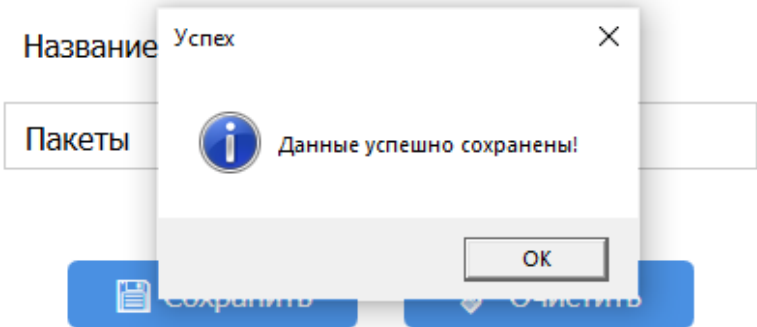


Рисунок 21

| | | |
|----|--------|---------------|
| 12 | Пакеты | Редактировать |
| | | Редактировать |

Рисунок 22

Система учета платежей - Управление платежами

← Назад

Система учета платежей

| ID | Дата | Пользователь | Название | Количество | Цена | Категория | Действия |
|----|------------|--------------|---------------|------------|---------------|----------------------|---------------|
| 1 | 15.01.2025 | | Бензин | 50 | 55.50 руб. | Транспортные расходы | Редактировать |
| 2 | 20.01.2025 | | Электричество | 1 | 2,500.00 руб. | Коммунальные услуги | Редактировать |
| 3 | 05.02.2025 | | Кино | 2 | 500.00 руб. | Развлечения | Редактировать |
| 4 | 10.02.2025 | | Продукты | 1 | 3,500.00 руб. | Продукты питания | Редактировать |
| 5 | 18.01.2025 | | Проездной | 1 | 2,000.00 руб. | Транспортные расходы | Редактировать |
| 6 | 25.01.2025 | | Интернет | 1 | 800.00 руб. | Коммунальные услуги | Редактировать |
| 7 | 01.02.2025 | | Куртка | 1 | 5,000.00 руб. | Одежда | Редактировать |
| 9 | 01.03.2025 | | Аптека | 3 | 450.00 руб. | Медицина | Редактировать |

+ Добавить

Удалить

10.11.2025 00:35:52

Рисунок 23

Система учета платежей

Добавление/Редактирование платежа

111

Дата платежа

10.11.2025

15

Количество

100

Цена

1000

Пользователь

Успех

×



Данные успешно сохранены!

ОК



Сохранить



Очистить

Рисунок 24

| № | Дата платежа | Номер платежа | Количество | Цена | Всего | Пакеты | Действия |
|----|--------------|---------------|------------|---------------|-------|--------|----------|
| 11 | 10.11.2025 | 111 | 100 | 1,000.00 руб. | | Пакеты | Действия |

Рисунок 25

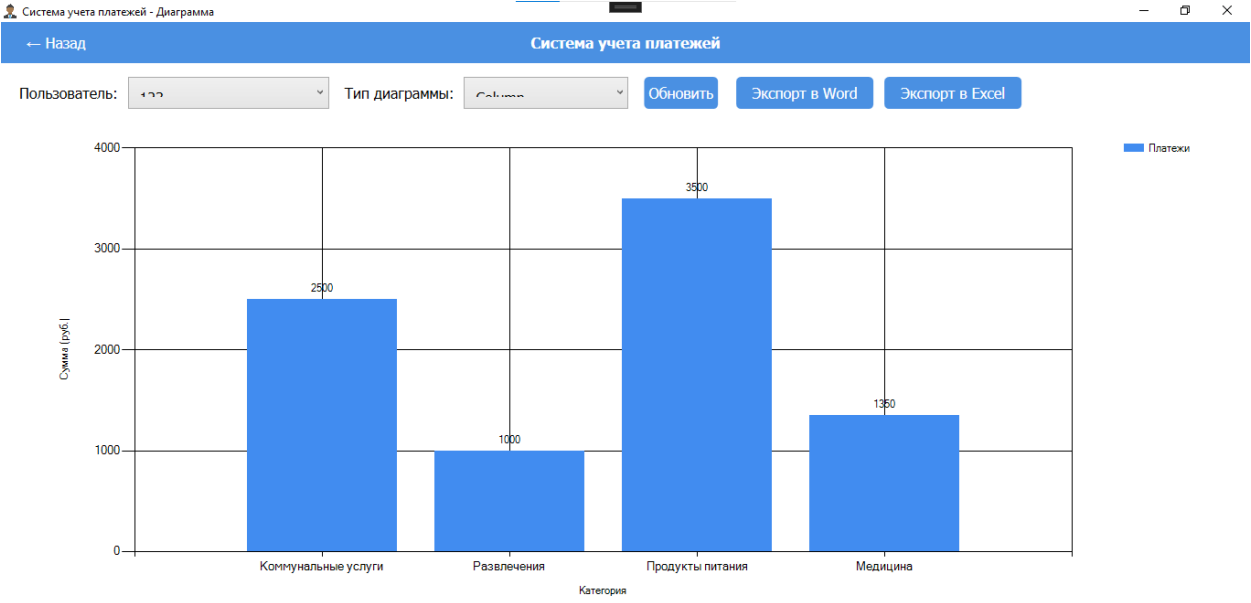


Рисунок 26

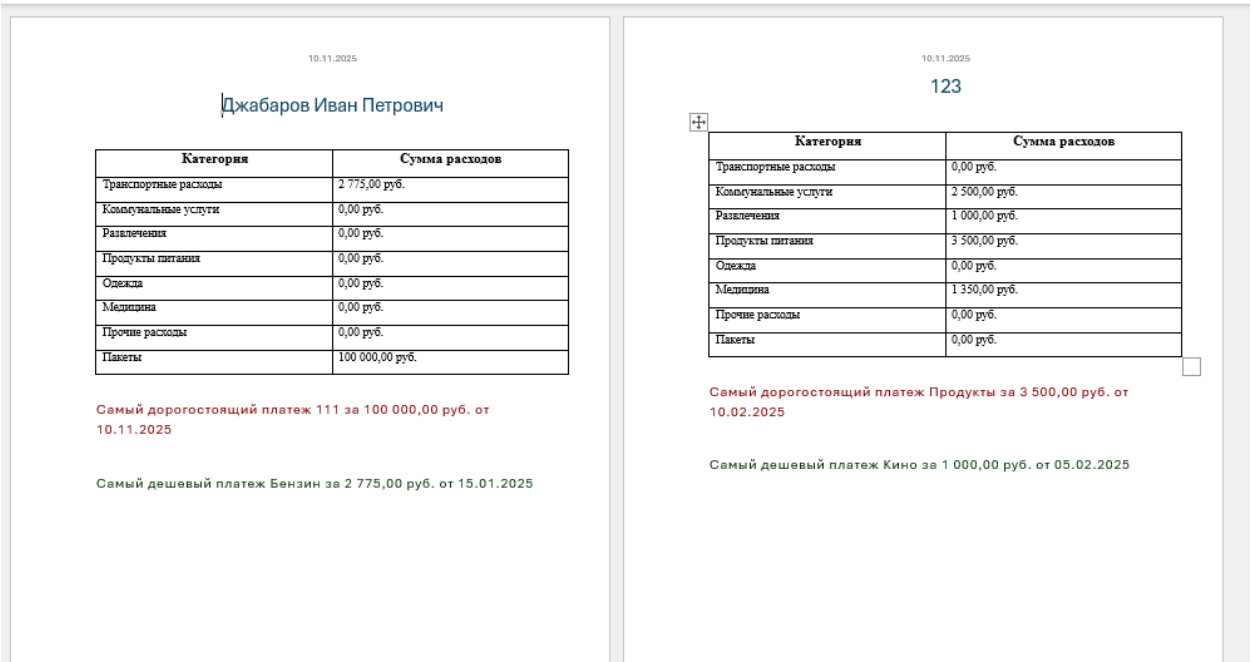


Рисунок 27

| A1 | | | | | Дата платежа | | | | |
|----|---------------------|---------------|-----------|------------|--------------|---|---|---|---|
| | A | B | C | D | E | F | G | H | I |
| 1 | Дата платежа | Название | Стоимость | Количество | Сумма | | | | |
| 2 | Коммунальные услуги | | | | | | | | |
| 3 | 20.01.2025 | Электричество | 2500,00 | 1 | 2500,00 | | | | |
| 4 | ИТОГО: | | | | 2500,00 | | | | |
| 5 | Медицина | | | | | | | | |
| 6 | 01.03.2025 | Аптека | 450,00 | 3 | 1350,00 | | | | |
| 7 | ИТОГО: | | | | 1350,00 | | | | |
| 8 | Продукты питания | | | | | | | | |
| 9 | 10.02.2025 | Продукты | 3500,00 | 1 | 3500,00 | | | | |
| 10 | ИТОГО: | | | | 3500,00 | | | | |
| 11 | Развлечения | | | | | | | | |
| 12 | 05.02.2025 | Кино | 500,00 | 2 | 1000,00 | | | | |
| 13 | ИТОГО: | | | | 1000,00 | | | | |
| 14 | | | | | | | | | |
| 15 | | | | | | | | | |
| 16 | | | | | | | | | |

Рисунок 28

| | ID | Name |
|---|------|----------------------|
| ▶ | 1 | Транспортные расходы |
| | 2 | Коммунальные услуги |
| | 3 | Развлечения |
| | 4 | Продукты питания |
| | 5 | Одежда |
| | 7 | Медицина |
| | 8 | Прочие расходы |
| | 12 | Пакеты |
| * | NULL | NULL |

Рисунок 29

| | ID | CategoryID | UserID | Date | Name | Num | Price |
|---|------|------------|--------|------------------------|---------------|------|---------|
| ▶ | 1 | 1 | 1 | 2025-01-15 00:00:00... | Бензин | 50 | 55,50 |
| | 2 | 2 | 2 | 2025-01-20 00:00:00... | Электричество | 1 | 2500,00 |
| | 3 | 3 | 2 | 2025-02-05 00:00:00... | Кино | 2 | 500,00 |
| | 4 | 4 | 2 | 2025-02-10 00:00:00... | Продукты | 1 | 3500,00 |
| | 5 | 1 | 3 | 2025-01-18 00:00:00... | Проездной | 1 | 2000,00 |
| | 6 | 2 | 3 | 2025-01-25 00:00:00... | Интернет | 1 | 800,00 |
| | 7 | 5 | 3 | 2025-02-01 00:00:00... | Куртка | 1 | 5000,00 |
| | 9 | 7 | 2 | 2025-03-01 00:00:00... | Аптека | 3 | 450,00 |
| | 11 | 12 | 1 | 2025-11-10 00:36:10... | 111 | 100 | 1000,00 |
| * | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

Рисунок 30

| DESKTOP-UQ1JDR1....yment - dbo.Users | | | | | | |
|---------------------------------------|------|-------|----------------------------------|-------|-----------------|-------|
| DESKTOP-UQ1JDR1.D...nt - dbo.Category | | | | | | |
| | ID | Login | Password | Role | FIO | Photo |
| ▶ | 1 | admin | 7C4A8D09CA3762AF61E59520943DC... | Admin | Джабаров Ива... | NULL |
| | 2 | user1 | 7C4A8D09CA3762AF61E59520943DC... | User | 123 | NULL |
| | 3 | user2 | 7C4A8D09CA3762AF61E59520943DC... | User | Сидоров Сидо... | NULL |
| | 4 | mexti | 7C4A8D09CA3762AF61E59520943DC... | User | Джабаров Мех... | NULL |
| | 5 | mex | DD5FEF9C1C1DA1394D6D34B248C51... | Admin | Lfgregdg | NULL |
| | 7 | xxx | 7C4A8D09CA3762AF61E59520943DC... | User | Джабаров | NULL |
| * | NULL | NULL | NULL | NULL | NULL | NULL |

Рисунок 31

Листинг №1 - Код "AddCategoryPage.xaml"

```
<Page x:Class="Djabarov_Payment.Pages.AddCategoryPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Категория">
    <Grid>
        <StackPanel HorizontalAlignment="Center"
                    VerticalAlignment="Center"
                    Width="400">

            <TextBlock Text="Добавление/Редактирование категории"
                      FontSize="18"
                      FontWeight="Bold"
                      HorizontalAlignment="Center"
                      Margin="0,0,0,30"/>

            <Label Content="Название категории"/>
            <TextBox x:Name="TBCategoryName"
                    Text="{Binding Name}"
                    Height="35"
                    ToolTip="Введите название категории"/>

            <StackPanel Orientation="Horizontal"
                      HorizontalAlignment="Center"
                      Margin="0,30,0,0">
                <Button x:Name="ButtonSaveCategory"
                        Content="💾 Сохранить"
                        Width="150"
                        Click="ButtonSaveCategory_Click"/>

                <Button x:Name="ButtonClean"
                        Content="🧼 Очистить"
                        Width="150"
                        Click="ButtonClean_Click"/>
            </StackPanel>
        </StackPanel>
    </Grid>
</Page>
```

Листинг №2 - Код "AddCategoryPage.xaml.cs"

```
using System;
using System.Text;
using System.Windows;
using System.Windows.Controls;

namespace Djabarov_Payment.Pages
{
```

```

    /// <summary>
    /// Страница добавления/редактирования категории
    /// </summary>
    public partial class AddCategoryPage : Page
    {
        private Category _currentCategory = new Category();

        public AddCategoryPage(Category selectedCategory)
        {
            InitializeComponent();

            if (selectedCategory != null)
                _currentCategory = selectedCategory;

            DataContext = _currentCategory;
        }

        /// <summary>
        /// Сохранение категории
        /// </summary>
        private void ButtonSaveCategory_Click(object sender, RoutedEventArgs
e)
        {
            StringBuilder errors = new StringBuilder();

            if (string.IsNullOrEmpty(_currentCategory.Name))
                errors.AppendLine("Укажите название категории!");

            if (errors.Length > 0)
            {
                MessageBox.Show(errors.ToString(), "Ошибка",
                    MessageBoxButton.OK, MessageBoxImage.Warning);
                return;
            }

            if (_currentCategory.ID == 0)
                Entities.GetContext().Category.Add(_currentCategory);

            try
            {
                Entities.GetContext().SaveChanges();
                MessageBox.Show("Данные успешно сохранены!",
                    "Успех", MessageBoxButton.OK,
                MessageBoxImage.Information);
                NavigationService?.GoBack();
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Ошибка при сохранении:\n{ex.Message}",
                    "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
            }
        }

        /// <summary>
        /// Очистка поля
        /// </summary>
        private void ButtonClean_Click(object sender, RoutedEventArgs e)
        {
            TBCategoryName.Text = "";
        }
    }
}

```

Листинг №3 - Код "AddPaymentPage.xaml"

```
<Page x:Class="Djabarov_Payment.Pages.AddPaymentPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Платёж">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*/>
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>

        <!-- Заголовок -->
        <TextBlock Grid.Row="0"
            Text="Добавление/Редактирование платежа"
            FontSize="18"
            FontWeight="Bold"
            HorizontalAlignment="Center"
            Margin="0,10,0,10"/>

        <!-- Форма -->
        <StackPanel Grid.Row="1"
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            >

            <Label Content="Название платежа"/>
            <TextBox x:Name="TBPaymentName"
                Text="{Binding Name}"
                ToolTip="Введите название платежа"/>

            <Label Content="Дата платежа"/>
            <DatePicker x:Name="DPDate"
                SelectedDate="{Binding Date}"
                ToolTip="Выберите дату платежа"/>

            <Label Content="Количество"/>
            <TextBox x:Name="TBCount"
                Text="{Binding Num}"
                ToolTip="Введите количество"/>

            <Label Content="Цена"/>
            <TextBox x:Name="TBAmount"
                Text="{Binding Price}"
                ToolTip="Введите цену"/>

            <Label Content="Пользователь"/>
            <ComboBox x:Name="CBUser"
                SelectedValuePath="ID"
                SelectedValue="{Binding Path=UserID}"
                ToolTip="Выберите пользователя"/>

            <Label Content="Категория"/>
            <ComboBox x:Name="CBCategory"
                SelectedValuePath="ID"
                SelectedValue="{Binding Path=CategoryID}"
                ToolTip="Выберите категорию"/>
        </StackPanel>

        <!-- Кнопки -->
        <StackPanel Grid.Row="2"
            Orientation="Horizontal"
            HorizontalAlignment="Center">
```

```

        Margin="0,10,0,10">
        <Button x:Name="ButtonSave"
            Content="💾 Сохранить"
            Click="ButtonSave_Click"/>

        <Button x:Name="ButtonClean"
            Content="🧹 Очистить"
            Click="ButtonClean_Click"/>
    </StackPanel>
</Grid>
</Page>

```

Листинг №4 - Код "AddPaymentPage.xaml.cs"

```

using System;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;

namespace Djabarov_Payment.Pages
{
    /// <summary>
    /// Страница добавления/редактирования платежа
    /// </summary>
    public partial class AddPaymentPage : Page
    {
        private Payment _currentPayment = new Payment();

        public AddPaymentPage(Payment selectedPayment)
        {
            InitializeComponent();

            // Загрузка списков пользователей и категорий
            CBCategory.ItemsSource =
Entities.GetContext().Category.ToList();
            CBCategory.DisplayMemberPath = "Name";

            CBUser.ItemsSource = Entities.GetContext().Users.ToList();
            CBUser.DisplayMemberPath = "FIO";

            if (selectedPayment != null)
                _currentPayment = selectedPayment;
            else
                _currentPayment.Date = DateTime.Now; // Устанавливаем
текущую дату по умолчанию

            DataContext = _currentPayment;
        }

        /// <summary>
        /// Сохранение платежа
        /// </summary>
        private void ButtonSave_Click(object sender, RoutedEventArgs e)
        {
            StringBuilder errors = new StringBuilder();

            if (string.IsNullOrEmpty(_currentPayment.Name))
                errors.AppendLine("Укажите название платежа!");

            if (_currentPayment.Date == DateTime.MinValue)
                errors.AppendLine("Укажите дату!");

```



```

        if (_currentPayment.Num <= 0)
            errors.AppendLine("Укажите корректное количество!");

        if (_currentPayment.Price <= 0)
            errors.AppendLine("Укажите корректную цену!");

        if (_currentPayment.UserID == 0)
            errors.AppendLine("Выберите пользователя!");

        if (_currentPayment.CategoryID == 0)
            errors.AppendLine("Выберите категорию!");

        if (errors.Length > 0)
        {
            MessageBox.Show(errors.ToString(), "Ошибка",
                MessageBoxButton.OK, MessageBoxImage.Warning);
            return;
        }

        if (_currentPayment.ID == 0)
            Entities.GetContext().Payment.Add(_currentPayment);

        try
        {
            Entities.GetContext().SaveChanges();
            MessageBox.Show("Данные успешно сохранены!",
                "Успех", MessageBoxButton.OK,
                MessageBoxImage.Information);
            NavigationService?.GoBack();
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Ошибка при сохранении:\n{ex.Message}",
                "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }

    /// <summary>
    /// Очистка полей формы
    /// </summary>
    private void ButtonClean_Click(object sender, RoutedEventArgs e)
    {
        TBPaymentName.Text = "";
        TBAmount.Text = "";
        TBCount.Text = "";
        DPDate.SelectedDate = null;
        CBCategory.SelectedIndex = -1;
        CBUser.SelectedIndex = -1;
    }
}

```

Листинг №5 - Код "AddUserPage.xaml"

```

<Page x:Class="Djabarov_Payment.Pages.AddUserPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Пользователь">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>

```

```

        <RowDefinition Height="*" />
        <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>

    <TextBlock Grid.Row="0"
        Text="Добавление/Редактирование пользователя"
        FontSize="18"
        FontWeight="Bold"
        HorizontalAlignment="Center"
        Margin="0,10,0,10" />

    <StackPanel Grid.Row="1"
        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        >

        <Label Content="Логин" />
        <TextBox x:Name="TBLogin"
            Text="{Binding Login}"
            ToolTip="Введите логин пользователя" />

        <Label Content="Пароль" />
        <TextBox x:Name="TBPass"
            Text="{Binding Password}"
            ToolTip="Введите пароль (будет захеширован при
сохранении)" />

        <Label Content="Роль" />
        <ComboBox Name="cmbRole"
            SelectedItem="{Binding Role}"
            ToolTip="Выберите роль пользователя">
            <ComboBoxItem Content="Admin" />
            <ComboBoxItem Content="User" />
        </ComboBox>

        <Label Content="ФИО" />
        <TextBox x:Name="TBFio"
            Text="{Binding FIO}"
            ToolTip="Введите ФИО пользователя" />

        <Label Content="Фото (путь к файлу, необязательно)" />
        <TextBox x:Name="TBPhoto"
            Text="{Binding Photo}"
            ToolTip="Введите путь к фото или оставьте пустым" />
    </StackPanel>

    <StackPanel Grid.Row="2"
        Orientation="Horizontal"
        HorizontalAlignment="Center"
        Margin="0,10,0,10">

        <Button x:Name="ButtonSave"
            Content="💾 Сохранить"
            Click="ButtonSave_Click" />

        <Button x:Name="ButtonClean"
            Content="🧹 Очистить"

            Click="ButtonClean_Click" />
    </StackPanel>
</Grid>
</Page>

```

Листинг №6 - Код "AddUserPage.xaml.cs"

```
using System;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Windows;
using System.Windows.Controls;

namespace Djabarov_Payment.Pages
{
    /// <summary>
    /// Страница добавления/редактирования пользователя
    /// </summary>
    public partial class AddUserPage : Page
    {
        private User _currentUser = new User();

        public AddUserPage(User selectedUser)
        {
            InitializeComponent();

            if (selectedUser != null)
                _currentUser = selectedUser;

            DataContext = _currentUser;
        }

        /// <summary>
        /// Получение хеша пароля SHA1
        /// </summary>
        public static string GetHash(String password)
        {
            using (var hash = SHA1.Create())
            {
                return
string.Concat(hash.ComputeHash(Encoding.UTF8.GetBytes(password))
                .Select(x => x.ToString("X2")));
            }
        }

        /// <summary>
        /// Сохранение пользователя
        /// </summary>
        private void ButtonSave_Click(object sender, RoutedEventArgs e)
        {
            StringBuilder errors = new StringBuilder();

            if (string.IsNullOrEmpty(_currentUser.Login))
                errors.AppendLine("Укажите логин!");

            if (string.IsNullOrEmpty(_currentUser.Password))
                errors.AppendLine("Укажите пароль!");

            if ((_currentUser.Role == null) || (cmbRole.Text == ""))
                errors.AppendLine("Выберите роль!");
            else
                _currentUser.Role = cmbRole.Text;

            if (string.IsNullOrEmpty(_currentUser.FIO))
                errors.AppendLine("Укажите ФИО!");

            if (errors.Length > 0)
            {

```

```

        MessageBox.Show(errors.ToString(), "Ошибка",
            MessageBoxButton.OK, MessageBoxImage.Warning);
        return;
    }

    // Проверка на существующий логин при добавлении нового
пользователя
    if (_currentUser.ID == 0)
    {
        var existingUser = Entities.GetContext().Users
            .FirstOrDefault(u => u.Login == _currentUser.Login);

        if (existingUser != null)
        {
            MessageBox.Show("Пользователь с таким логином уже
существует!",
                "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
            return;
        }

        // Хешируем пароль при добавлении нового пользователя
        _currentUser.Password = GetHash(_currentUser.Password);
        Entities.GetContext().Users.Add(_currentUser);
    }
    else
    {
        // При редактировании: если пароль изменился и не является
хешем
        if (!string.IsNullOrEmpty(TBPass.Text) && TBPass.Text.Length
< 40)
        {
            _currentUser.Password = GetHash(TBPass.Text);
        }
    }

    try
    {
        Entities.GetContext().SaveChanges();
        MessageBox.Show("Данные успешно сохранены!",
            "Успех", MessageBoxButton.OK,
MessageBoxImage.Information);
        NavigationService?.GoBack();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при сохранении:\n{ex.Message}",
            "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
    }
}

/// <summary>
/// Очистка полей формы
/// </summary>
private void ButtonClean_Click(object sender, RoutedEventArgs e)
{
    TBLogin.Text = "";
    TBPass.Text = "";
    cmbRole.SelectedIndex = -1;
    TBFio.Text = "";
    TBPhoto.Text = "";
}
}
}

```

Листинг №7 - Код "AdminPage.xaml"

```
<Page x:Class="Djabarov_Payment.Pages.AdminPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Панель администратора">
    <Grid Background="#F5F5F5">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <TextBlock Text="Панель администратора"
                  Grid.Row="0"
                  FontSize="24"
                  FontWeight="Bold"
                  HorizontalAlignment="Center"
                  Margin="0,20,0,20"/>

        <UniformGrid Grid.Row="1"
                    Columns="2"
                    Rows="2"
                    HorizontalAlignment="Center"
                    VerticalAlignment="Center"
                    Width="600"
                    Height="400">

            <Button x:Name="BtnTab1"
                    Content="👤&#x0a;Управление&#x0a;пользователями"
                    FontSize="16"
                    Margin="20"
                    Height="150"
                    Click="BtnTab1_Click"/>

            <Button x:Name="BtnTab2"
                    Content="📁&#x0a;Управление&#x0a;категориями"
                    FontSize="16"
                    Margin="20"
                    Height="150"
                    Click="BtnTab2_Click"/>

            <Button x:Name="BtnTab3"
                    Content="💰&#x0a;Управление&#x0a;платежами"
                    FontSize="16"
                    Margin="20"
                    Height="150"
                    Click="BtnTab3_Click"/>

            <Button x:Name="BtnTab4"
                    Content="📊&#x0a;Диаграммы&#x0a;и отчеты"
                    FontSize="16"
                    Margin="20"
                    Height="150"
                    Click="BtnTab4_Click"/>
        </UniformGrid>
    </Grid>
</Page>
```

Листинг №8 - Код "AdminPage.xaml.cs"

```

using System.Windows;
using System.Windows.Controls;

namespace Djabarov_Payment.Pages
{
    /// <summary>
    /// Панель администратора с навигацией по таблицам
    /// </summary>
    public partial class AdminPage : Page
    {
        public AdminPage()
        {
            InitializeComponent();

            /// <summary>
            /// Переход к управлению пользователями
            /// </summary>
            private void BtnTab1_Click(object sender, RoutedEventArgs e)
            {
                NavigationService?.Navigate(new UsersTabPage());
            }

            /// <summary>
            /// Переход к управлению категориями
            /// </summary>
            private void BtnTab2_Click(object sender, RoutedEventArgs e)
            {
                NavigationService?.Navigate(new CategoryTabPage());
            }

            /// <summary>
            /// Переход к управлению платежами
            /// </summary>
            private void BtnTab3_Click(object sender, RoutedEventArgs e)
            {
                NavigationService?.Navigate(new PaymentTabPage());
            }

            /// <summary>
            /// Переход к диаграммам и отчетам
            /// </summary>
            private void BtnTab4_Click(object sender, RoutedEventArgs e)
            {
                NavigationService?.Navigate(new DiagrammPage());
            }
        }
    }
}

```

Листинг №9 - Код "App.xaml"

```

<Application x:Class="Djabarov_Payment.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
StartupUri="MainWindow.xaml">
    <Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="Dictionary.xaml"/>
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Application.Resources>
</Application>

```

```
</Application.Resources>
</Application>
```

Листинг №10 - Код "App.xaml.cs"

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Threading.Tasks;
using System.Windows;

namespace Djabarov_Payment
{
    /// <summary>
    /// Логика взаимодействия для App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}
```

Листинг №11 - Код "AuthPage.xaml"

```
<Page x:Class="Djabarov_Payment.Pages.AuthPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Авторизация">

    <Grid>
        <StackPanel HorizontalAlignment="Center"
                    VerticalAlignment="Center"
                    Width="300">

            <TextBlock Text="🔑" FontSize="100"
HorizontalAlignment="Center"/>

            <Label Content="Логин" x:Name="labelLogin"/>
            <Grid>
                <TextBox x:Name="TextBoxLogin"
                        Background="Transparent"
                        ToolTip="Введите логин"
                        TextChanged="TextBoxLogin_TextChanged" />
                <Label x:Name="txtHintLogin"
                        Content="Введите логин"
                        Foreground="Gray"
                        IsHitTestVisible="False"
                        VerticalAlignment="Center"
                        Padding="12,0,0,0"
                        MouseLeftButtonUp="txtHintLogin_MouseLeftButtonUp"/>
            </Grid>

            <Label Content="Пароль" x:Name="labelPass" Margin="0,10,0,0"/>
            <Grid>
                <PasswordBox x:Name="PasswordBox"
                        Background="Transparent"
                        ToolTip="Введите пароль"
                        PasswordChanged="PasswordBox_PasswordChanged"
                />
            </Grid>
        </StackPanel>
    </Grid>
</Page>
```

```

        <Label x:Name="txtHintPass"
            Content="Введите пароль"
            Foreground="Gray"
            IsHitTestVisible="False"
            VerticalAlignment="Center"
            Padding="12,0,0,0"
            MouseLeftButtonUp="txtHintPass_MouseLeftButtonUp"/>
    </Grid>

    <Label Content="Капча:" Name="labelCaptcha" Margin="0,10,0,0"
Visibility="Hidden"/>
    <TextBox x:Name="captcha"

CommandManager.PreviewExecuted="textBox_PreviewExecuted"
        ContextMenu="{x:Null}"
        Visibility="Hidden"
        Height="42"
        Foreground="Red" FontStyle="Italic" FontWeight="Heavy"
        FontStretch="UltraCondensed" IsReadOnly="True"
        HorizontalContentAlignment="Center"
        VerticalContentAlignment="Center" FontSize="20" />
    <TextBox x:Name="captchaInput" Visibility="Hidden" />

    <Button x:Name="ButtonEnter"
        Content="Вход"
        ToolTip="Кнопка входа"
        Click="ButtonEnter_OnClick" />

    <Button x:Name="ButtonReg"
        Content="Регистрация"
        ToolTip="Кнопка регистрации"
        Click="ButtonReg_Click" />

    <Button x:Name="ButtonChangePassword"
        Content="Смена пароля"
        ToolTip="Кнопка смены пароля"
        Click="ButtonChangePassword_Click" />

    <Button x:Name="submitCaptcha"
        Visibility="Hidden"
        Content="Подтвердить"
        ToolTip="Подтвердить ввод капчи"
        Click="submitCaptcha_Click" />
    </StackPanel>
</Grid>
</Page>

```

Листинг №12 - Код "AuthPage.xaml.cs"

```

using System;
using System.Data.Entity;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;

namespace Djabarov_Payment.Pages
{
    /// <summary>
    /// Страница авторизации пользователей

```



```

/// </summary>
public partial class AuthPage : Page
{
    private int failedAttempts = 0;
    private User currentUser;

    public AuthPage()
    {
        InitializeComponent();
    }

    /// <summary>
    /// Получение хеша пароля SHA1
    /// </summary>
    public static string GetHash(String password)
    {
        using (var hash = SHA1.Create())
        {
            return
string.Concat(hash.ComputeHash(Encoding.UTF8.GetBytes(password))
                .Select(x => x.ToString("X2")));
        }
    }

    /// <summary>
    /// Обработчик кнопки входа
    /// </summary>
    private void ButtonEnter_OnClick(object sender, RoutedEventArgs e)
    {
        if (string.IsNullOrEmpty(TextBoxLogin.Text) ||
string.IsNullOrEmpty>PasswordBox.Password))
        {
            MessageBox.Show("Введите логин и пароль", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Warning);
            return;
        }

        string hashedPassword = GetHash>PasswordBox.Password);
        using (var db = new Entities())
        {
            var user = db.Users
                .AsNoTracking()
                .FirstOrDefault(u => u.Login == TextBoxLogin.Text &&
u.Password == hashedPassword);

            if (user == null)
            {
                MessageBox.Show("Пользователь с такими данными не
найден!",
                    "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
                failedAttempts++;

                if (failedAttempts >= 3)
                {
                    if (captcha.Visibility != Visibility.Visible)
                    {
                        CaptchaSwitch();
                    }
                    CaptchaChange();
                }
            }
            else
            {

```

```

        MessageBox.Show("Авторизация успешна!", "Успех",
            MessageBoxButton.OK, MessageBoxImage.Information);

        switch (user.Role)
        {
            case "User":
                NavigationService?.Navigate(new UserPage());
                break;
            case "Admin":
                NavigationService?.Navigate(new AdminPage());
                break;
        }
    }
}

/// <summary>
/// Переход на страницу регистрации
/// </summary>
private void ButtonReg_Click(object sender, RoutedEventArgs e)
{
    NavigationService?.Navigate(new RegPage());
}

/// <summary>
/// Переход на страницу смены пароля
/// </summary>
private void ButtonChangePassword_Click(object sender,
RoutedEventArgs e)
{
    NavigationService?.Navigate(new ChangePassPage());
}

/// <summary>
/// Переключение видимости капчи
/// </summary>
public void CaptchaSwitch()
{
    switch (captcha.Visibility)
    {
        case Visibility.Visible:
            TextBoxLogin.Clear();
            PasswordBox.Clear();
            captcha.Visibility = Visibility.Hidden;
            captchaInput.Visibility = Visibility.Hidden;
            labelCaptcha.Visibility = Visibility.Hidden;
            submitCaptcha.Visibility = Visibility.Hidden;

            labelLogin.Visibility = Visibility.Visible;
            labelPass.Visibility = Visibility.Visible;
            TextBoxLogin.Visibility = Visibility.Visible;
            txtHintLogin.Visibility = Visibility.Visible;
            PasswordBox.Visibility = Visibility.Visible;
            txtHintPass.Visibility = Visibility.Visible;

            ButtonChangePassword.Visibility = Visibility.Visible;
            ButtonEnter.Visibility = Visibility.Visible;
            ButtonReg.Visibility = Visibility.Visible;
            return;

        case Visibility.Hidden:
            captcha.Visibility = Visibility.Visible;
            captchaInput.Visibility = Visibility.Visible;
            labelCaptcha.Visibility = Visibility.Visible;
    }
}

```

```

        submitCaptcha.Visibility = Visibility.Visible;

        labelLogin.Visibility = Visibility.Hidden;
        labelPass.Visibility = Visibility.Hidden;
        TextBoxLogin.Visibility = Visibility.Hidden;
        txtHintLogin.Visibility = Visibility.Hidden;
        PasswordBox.Visibility = Visibility.Hidden;
        txtHintPass.Visibility = Visibility.Hidden;

        ButtonChangePassword.Visibility = Visibility.Hidden;
        ButtonEnter.Visibility = Visibility.Hidden;
        ButtonReg.Visibility = Visibility.Hidden;
        return;
    }
}

/// <summary>
/// Генерация новой капчи
/// </summary>
public void CaptchaChange()
{
    String allowchar = "";
    allowchar =
"A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z";
    allowchar +=
"a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,y,z";
    allowchar += "1,2,3,4,5,6,7,8,9,0";

    char[] a = { ' ' };
    String[] ar = allowchar.Split(a);
    String pwd = "";
    string temp = "";
    Random r = new Random();

    for (int i = 0; i < 6; i++)
    {
        temp = ar[(r.Next(0, ar.Length))];
        pwd += temp;
    }
    captcha.Text = pwd;
}

/// <summary>
/// Подтверждение капчи
/// </summary>
private void submitCaptcha_Click(object sender, RoutedEventArgs e)
{
    if (captchaInput.Text != captcha.Text)
    {
        MessageBox.Show("Неверно введена капча", "Ошибка",
            MessageBoxButton.OK, MessageBoxImage.Error);
        CaptchaChange();
    }
    else
    {
        MessageBox.Show("Капча введена успешно, можете продолжить
авторизацию",
            "Успех", MessageBoxButton.OK,
            MessageBoxImage.Information);
        CaptchaSwitch();
        failedAttempts = 0;
    }
}

```

```

        /// <summary>
        /// Запрет копирования капчи
        /// </summary>
        private void textBox_PreviewExecuted(object sender,
        ExecutedRoutedEventArgs e)
        {
            if (e.Command == ApplicationCommands.Copy ||
                e.Command == ApplicationCommands.Cut ||
                e.Command == ApplicationCommands.Paste)
            {
                e.Handled = true;
            }
        }

        // Обработчики для placeholder текста
        private void TextBoxLogin_TextChanged(object sender,
        TextChangedEventArgs e)
        {
            txtHintLogin.Visibility =
            string.IsNullOrEmpty(TextBoxLogin.Text) ?
                Visibility.Visible : Visibility.Hidden;
        }

        private void PasswordBox_PasswordChanged(object sender,
        RoutedEventArgs e)
        {
            txtHintPass.Visibility =
            string.IsNullOrEmpty>PasswordBox.Password) ?
                Visibility.Visible : Visibility.Hidden;
        }

        private void txtHintLogin_MouseLeftButtonUp(object sender,
        MouseButtonEventArgs e)
        {
            TextBoxLogin.Focus();
        }

        private void txtHintPass_MouseLeftButtonUp(object sender,
        MouseButtonEventArgs e)
        {
            PasswordBox.Focus();
        }
    }
}

```

Листинг №13 - Код "CategoryTabPage.xaml"

```

<Page x:Class="Djabarov_Payment.Pages.CategoryTabPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Управление категориями">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="60" />
        </Grid.RowDefinitions>

        <!-- Таблица катергорий -->
        <DataGrid Grid.Row="0" x:Name="DataGridCategory"
                  AutoGenerateColumns="False"
                  IsReadOnly="True"
                  SelectionMode="Extended">

```

```

        <DataGrid.Columns>
            <DataGridTextColumn Header="ID" Binding="{Binding ID}"
Width="100"/>
            <DataGridTextColumn Header="Название категории"
Binding="{Binding Name}" Width="400"/>
            <DataGridTemplateColumn Width="*" Header="Действия">
                <DataGridTemplateColumn.CellTemplate>
                    <DataTemplate>
                        <Button x:Name="ButtonEdit"
                            Content="Редактировать"
                            ToolTip="Редактировать категорию"
                            Click="ButtonEdit_Click"
                            Margin="1"/>
                    </DataTemplate>
                </DataGridTemplateColumn.CellTemplate>
            </DataGridTemplateColumn>
        </DataGrid.Columns>
    </DataGrid>

    <!-- Панель кнопок -->
    <Grid Grid.Row="1" Background="#F0F0F0">
        <StackPanel Orientation="Horizontal"
            HorizontalAlignment="Center"
            VerticalAlignment="Center">
            <Button x:Name="ButtonAdd"
                ToolTip="Добавить новую категорию"
                Content="✚ Добавить"
                Width="150"
                Click="ButtonAdd_Click"/>

            <Button x:Name="ButtonDel"
                ToolTip="Удалить выбранные категории"
                Content="🗑 Удалить"
                Width="150"
                Click="ButtonDel_Click"/>
        </StackPanel>
    </Grid>
</Grid>
</Page>

```

Листинг №14 - Код "CategoryTabPage.xaml.cs"

```

using System;
using System.Linq;
using System.Windows;
using System.Windows.Controls;

namespace Djabarov_Payment.Pages
{
    /// <summary>
    /// Страница управления таблицей категорий
    /// </summary>
    public partial class CategoryTabPage : Page
    {
        public CategoryTabPage()
        {
            InitializeComponent();
            DataGridCategory.ItemsSource =
Entities.GetContext().Category.ToList();
            this.IsVisibleChanged += Page_IsVisibleChanged;
        }
    }
}

```

```

        /// <summary>
        /// Обновление данных при возврате на страницу
        /// </summary>
        private void Page_IsVisibleChanged(object sender,
DependencyPropertyChangedEventArgs e)
        {
            if (Visibility == Visibility.Visible)
            {

Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(x =>
x.Reload());
                DataGridCategory.ItemsSource =
Entities.GetContext().Category.ToList();
            }
        }

        /// <summary>
        /// Добавление новой категории
        /// </summary>
        private void ButtonAdd_Click(object sender, RoutedEventArgs e)
        {
            NavigationService?.Navigate(new AddCategoryPage(null));
        }

        /// <summary>
        /// Удаление выбранных категорий
        /// </summary>
        private void ButtonDel_Click(object sender, RoutedEventArgs e)
        {
            var categoryForRemoving =
DataGridCategory.SelectedItems.Cast<Category>().ToList();

            if (categoryForRemoving.Count == 0)
            {
                MessageBox.Show("Выберите категории для удаления!",
                    "Предупреждение", MessageBoxButton.OK,
MessageBoxImage.Warning);
                return;
            }

            if (MessageBox.Show($"Вы точно хотите удалить записи в
количестве {categoryForRemoving.Count()} элементов?",
                "Подтверждение", MessageBoxButton.YesNo,
MessageBoxImage.Question) == MessageBoxResult.Yes)
            {
                try
                {
Entities.GetContext().Category.RemoveRange(categoryForRemoving);
                    Entities.GetContext().SaveChanges();
                    MessageBox.Show("Данные успешно удалены!",
                        "Успех", MessageBoxButton.OK,
MessageBoxImage.Information);
                    DataGridCategory.ItemsSource =
Entities.GetContext().Category.ToList();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Ошибка при удалении:\n{ex.Message}",
                        "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
                }
            }
        }
    }

```

```

    }

    /// <summary>
    /// Редактирование категории
    /// </summary>
    private void ButtonEdit_Click(object sender, RoutedEventArgs e)
    {
        NavigationService?.Navigate(new AddCategoryPage((sender as
        Button).DataContext as Category));
    }
}

```

Листинг №15 - Код "ChangePassPage.xaml"

```

<Page x:Class="Djabarov_Payment.Pages.ChangePassPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Смена пароля">
    <Grid>
        <StackPanel HorizontalAlignment="Center" VerticalAlignment="Center"
        Width="300">
            <TextBlock Text="Смена пароля" FontSize="20" FontWeight="Bold"
            HorizontalAlignment="Center" Margin="0,0,0,20"/>

            <Label Content="Логин"/>
            <TextBox x:Name="TbLogin" ToolTip="Введите логин" Height="35"/>

            <Label Content="Текущий пароль" Margin="0,10,0,0"/>
            <PasswordBox x:Name="CurrentPasswordBox" ToolTip="Введите
текущий пароль" Height="35"/>

            <Label Content="Новый пароль" Margin="0,10,0,0"/>
            <PasswordBox x:Name="NewPasswordBox" ToolTip="Введите новый
пароль" Height="35"/>

            <Label Content="Подтверждение нового пароля" Margin="0,10,0,0"/>
            <PasswordBox x:Name="ConfirmPasswordBox" ToolTip="Подтвердите
новый пароль" Height="35"/>

            <Button x:Name="ButtonSave" Content="Сменить пароль"
            Margin="0,20,0,0" Click="ButtonSave_Click"/>
        </StackPanel>
    </Grid>
</Page>

```

Листинг №16 - Код "ChangePassPage.xaml.cs"

```

using System;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Windows;
using System.Windows.Controls;

namespace Djabarov_Payment.Pages
{
    /// <summary>
    /// Страница смены пароля
    /// </summary>
    public partial class ChangePassPage : Page

```

```

{
    public ChangePassPage()
    {
        InitializeComponent();
    }

    /// <summary>
    /// Получение хеша пароля SHA1
    /// </summary>
    public static string GetHash(String password)
    {
        using (var hash = SHA1.Create())
        {
            return
string.Concat(hash.ComputeHash(Encoding.UTF8.GetBytes(password))
                .Select(x => x.ToString("X2")));
        }
    }

    /// <summary>
    /// Обработчик кнопки смены пароля
    /// </summary>
    private void ButtonSave_Click(object sender, RoutedEventArgs e)
    {
        // Проверка заполнения всех полей
        if (string.IsNullOrEmpty(CurrentPasswordBox.Password) ||
            string.IsNullOrEmpty(NewPasswordBox.Password) ||
            string.IsNullOrEmpty(ConfirmPasswordBox.Password) ||
            string.IsNullOrEmpty(TbLogin.Text))
        {
            MessageBox.Show("Все поля обязательны к заполнению!",
                "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);
            return;
        }

        // Проверка пользователя
        string hashedPass = GetHash(CurrentPasswordBox.Password);
        using (var db = new Entities())
        {
            var user = db.Users
                .FirstOrDefault(u => u.Login == TbLogin.Text &&
u.Password == hashedPass);

            if (user == null)
            {
                MessageBox.Show("Текущий пароль/логин неверный!",
                    "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
                return;
            }

            // Проверка нового пароля
            if (NewPasswordBox.Password.Length < 6)
            {
                MessageBox.Show("Пароль слишком короткий, должно быть
минимум 6 символов!",
                    "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Warning);
                return;
            }

            bool en = true;
            bool number = false;

```



```

        for (int i = 0; i < NewPasswordBox.Password.Length; i++)
        {
            if (NewPasswordBox.Password[i] >= '0' &&
NewPasswordBox.Password[i] <= '9')
                number = true;
            else if (!((NewPasswordBox.Password[i] >= 'A' &&
NewPasswordBox.Password[i] <= 'Z') ||
                (NewPasswordBox.Password[i] >= 'a' &&
NewPasswordBox.Password[i] <= 'z'))))
                en = false;
        }

        if (!en)
        {
            MessageBox.Show("Используйте только английскую
раскладку!",
                "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Warning);
            return;
        }

        if (!number)
        {
            MessageBox.Show("Добавьте хотя бы одну цифру!",
                "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Warning);
            return;
        }

        // Проверка совпадения паролей
        if (NewPasswordBox.Password != ConfirmPasswordBox.Password)
        {
            MessageBox.Show("Новые пароли не совпадают!",
                "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
            return;
        }

        // Сохранение нового пароля
        if (en && number)
        {
            user.Password = GetHash(NewPasswordBox.Password);
            db.SaveChanges();

            MessageBox.Show("Пароль успешно изменен!",
                "Успех", MessageBoxButton.OK,
MessageBoxImage.Information);

            NavigationService?.Navigate(new AuthPage());
        }
    }
}
}
}

```

Листинг №17 - Код "DiagrammPage.xaml"

```

<Page x:Class="Djabarov_Payment.Pages.DiagrammPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:wf="clr-
namespace:System.Windows.Forms;assembly=System.Windows.Forms"

```

```

        xmlns:wfi="clr-
namespace:System.Windows.Forms.Integration;assembly=WindowsFormsIntegration"
        Title="Диаграмма">

        <Grid Margin="10">
            <Grid.RowDefinitions>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="*/>
            </Grid.RowDefinitions>

            <StackPanel Orientation="Horizontal" VerticalAlignment="Top"
Margin="10,0,0,10" HorizontalAlignment="Left">
                <TextBlock Text="Пользователь:" VerticalAlignment="Center"
Margin="0,0,6,0"/>
                <ComboBox x:Name="CmbUser" Width="220" DisplayMemberPath="FIO"/>

                <TextBlock Text="Тип диаграммы:" VerticalAlignment="Center"
Margin="12,0,6,0"/>
                <ComboBox x:Name="CmbDiagram" Width="180"/>

                <Button Content="Обновить" Margin="12,0,0,0"
Click="ButtonRefresh_Click"/>

                <Button x:Name="ButtonWord" Content="Экспорт в Word" Width="150"
Margin="20,0,0,0" Click="ButtonWord_Click"/>
                <Button x:Name="ButtonExcel" Content="Экспорт в Excel"
Width="150" Margin="12,0,0,0" Click="ButtonExcel_Click"/>
            </StackPanel>

            <wfi:WindowsFormsHost x:Name="ChartHost" Grid.Row="1"
HorizontalAlignment="Stretch" VerticalAlignment="Stretch" />

            <TextBlock x:Name="NoDataText"
                Grid.Row="1"
                Text="Нет данных для отображения"
                FontSize="18"
                FontWeight="Bold"
                HorizontalAlignment="Center"
                VerticalAlignment="Center"
                Visibility="Collapsed"/>
        </Grid>
</Page>

```

Листинг №18 - Код "DiagrammPage.xaml.cs"

```

using System;
using System.Linq;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Forms.DataVisualization.Charting;
using System.Windows.Forms.Integration; // WindowsFormsHost
using System.Reflection;
using System.Runtime.InteropServices;
using Word = Microsoft.Office.Interop.Word;
using Excel = Microsoft.Office.Interop.Excel;

namespace Djabarov_Payment.Pages
{
    public partial class DiagrammPage : Page
    {
        private Chart chart;
    }
}

```

```

public DiagrammPage()
{
    InitializeComponent();

    InitializeChart();
    LoadData();

    // Подписка на событие SelectionChanged — безопасно, ведь
элементы XAML уже инициализированы
    CmbUser.SelectionChanged += UpdateChart;
    CmbDiagram.SelectionChanged += UpdateChart;
}

private void InitializeChart()
{
    chart = new Chart
    {
        Dock = System.Windows.Forms.DockStyle.Fill
    };

    var chartArea = new ChartArea("Main");
    chartArea.AxisX.Title = "Категория";
    chartArea.AxisY.Title = "Сумма (руб.)";
    chartArea.AxisX.LabelStyle.Angle = -45;
    chart.ChartAreas.Clear();
    chart.ChartAreas.Add(chartArea);

    // Создаём серию (одна на страницу)
    var series = new Series("Платежи")
    {
        IsValueShownAsLabel = true,
        ChartType = SeriesChartType.Column
    };
    chart.Series.Clear();
    chart.Series.Add(series);

    chart.Legends.Clear();
    chart.Legends.Add(new Legend("Legend"));

    // Привязать WinForms Chart к WindowsFormsHost в XAML
    ChartHost.Child = chart;
}

private void LoadData()
{
    try
    {
        // Подгружаем пользователей (DisplayMemberPath в XAML
установлен как FIO)
        var users = Entities.GetContext().Users.ToList();
        CmbUser.ItemsSource = users;
        if (users.Count > 0)
            CmbUser.SelectedIndex = 0;

        // Заполняем типы диаграмм (enum SeriesChartType)
        CmbDiagram.ItemsSource =
Enum.GetValues(typeof(SeriesChartType)).Cast<SeriesChartType>().ToList();

        // По умолчанию Column (если есть)
        var defaultIndex =
((SeriesChartType[])Enum.GetValues(typeof(SeriesChartType))).ToList().IndexO
f(SeriesChartType.Column);
        if (defaultIndex >= 0)

```

```

        CmbDiagram.SelectedIndex = defaultIndex;
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при загрузке
данных:\n{ex.Message}",
            "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

    // Обновим диаграмму первый раз
    UpdateChart(null, null);
}

// Обработчик как для SelectionChanged (sender может быть null при
первом вызове)
private void UpdateChart(object sender, SelectionChangedEventArgs e)
{
    // Защита: если элементы ещё не инициализированы
    if (!IsLoaded) return;

    var currentUser = CmbUser.SelectedItem as User;
    if (currentUser == null)

    {
        // Нет выбранного пользователя — очищаем график
        ClearChartAndShowNoData();
        return;
    }

    var currentType = (SeriesChartType?)CmbDiagram.SelectedItem;
    if (currentType == null)

    {
        ClearChartAndShowNoData();
        return;
    }

    try
    {
        var series = chart.Series.FirstOrDefault(s => s.Name ==
"Платежи");
        if (series == null)
        {
            series = new Series("Платежи") { IsValueShownAsLabel =
true };
            chart.Series.Add(series);
        }

        series.ChartType = currentType.Value;
        series.Points.Clear();

        var categories = Entities.GetContext().Category.ToList();
        bool hasData = false;

        foreach (var category in categories)
        {
            // Сумма = Price * Num для платежей выбранного
пользователя и категории
            var totalSum = Entities.GetContext().Payment
                .Where(p => p.UserID == currentUser.ID &&
p.CategoryID == category.ID)
                .Select(p => (decimal?) (p.Price * p.Num))
                .Sum() ?? 0m;

```

```

        if (totalSum > 0)
        {
            series.Points.AddXY(category.Name,
(double)totalSum);
            hasData = true;
        }
    }


    NoDataText.Visibility = hasData ? Visibility.Collapsed :
Visibility.Visible;
    ChartHost.Visibility = hasData ? Visibility.Visible :
Visibility.Collapsed;

    // Обновляем легенду/формат подписей (при необходимости)
    chart.Invalidate();
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при обновлении
диаграммы:\n{ex.Message}",
        "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
}
}

private void ClearChartAndShowNoData()
{
    var series = chart.Series.FirstOrDefault(s => s.Name ==
"Платежи");
    if (series != null) series.Points.Clear();

    ChartHost.Visibility = Visibility.Collapsed;
    NoDataText.Visibility = Visibility.Visible;
}

// Кнопка вручную обновить (в XAML привязана к Click)
private void ButtonRefresh_Click(object sender, RoutedEventArgs e)
{
    UpdateChart(null, null);
}

//  (Шаг 13) Обработчик для кнопки экспорта в Word
private void ButtonWord_Click(object sender, RoutedEventArgs e)
{
    var allUsers = Entities.GetContext().Users.ToList();
    var allCategories = Entities.GetContext().Category.ToList();

    Word.Application application = null;
    Word.Document document = null;

    try
    {
        application = new Word.Application();
        document = application.Documents.Add();

        foreach (var user in allUsers)
        {
            // Добавление ФИО пользователя
            Word.Paragraph userParagraph =
document.Paragraphs.Add();
            Word.Range userRange = userParagraph.Range;
            userRange.Text = user.FIO;
            userParagraph.set_Style("Заголовок 1"); // Убедитесь,
что стиль "Заголовок 1" существует в Word
            userRange.ParagraphFormat.Alignment =

```

```

Word.WdParagraphAlignment.wdAlignParagraphCenter;
    userRange.InsertParagraphAfter();
    document.Paragraphs.Add();

    // Добавление таблицы
    Word.Paragraph tableParagraph =
document.Paragraphs.Add();
    Word.Range tableRange = tableParagraph.Range;
    Word.Table paymentsTable =
document.Tables.Add(tableRange, allCategories.Count() + 1, 2);
    paymentsTable.Borders.InsideLineStyle =
Word.WdLineStyle.wdLineStyleSingle;
    paymentsTable.Borders.OutsideLineStyle =
Word.WdLineStyle.wdLineStyleSingle;
    paymentsTable.Range.Cells.VerticalAlignment =
Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

    // Заголовки таблицы
    Word.Range cellRange;
    cellRange = paymentsTable.Cell(1, 1).Range;
    cellRange.Text = "Категория";
    cellRange = paymentsTable.Cell(1, 2).Range;
    cellRange.Text = "Сумма расходов";
    paymentsTable.Rows[1].Range.Font.Name = "Times New
Roman";

    paymentsTable.Rows[1].Range.Font.Size = 14;
    paymentsTable.Rows[1].Range.Bold = 1;
    paymentsTable.Rows[1].Range.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter;

    // Заполнение таблицы
    for (int i = 0; i < allCategories.Count(); i++)
    {
        var currentCategory = allCategories[i];
        cellRange = paymentsTable.Cell(i + 2, 1).Range;
        cellRange.Text = currentCategory.Name;
        cellRange.Font.Name = "Times New Roman";
        cellRange.Font.Size = 12;

        cellRange = paymentsTable.Cell(i + 2, 2).Range;
        var sum = user.Payments
            .Where(u => u.CategoryID ==
currentCategory.ID)
            .Sum(u => (decimal?) (u.Price * u.Num)
?? 0m;

        cellRange.Text = sum.ToString("N2") + " руб.";
        cellRange.Font.Name = "Times New Roman";
        cellRange.Font.Size = 12;
    }
    document.Paragraphs.Add();

    // Максимальный платеж
    Payment maxPayment = user.Payments.OrderByDescending(u
=> u.Price * u.Num).FirstOrDefault();
    if (maxPayment != null)
    {
        Word.Paragraph maxPaymentParagraph =
document.Paragraphs.Add();
        Word.Range maxPaymentRange =
maxPaymentParagraph.Range;
        maxPaymentRange.Text = $"Самый дорогостоящий платеж
{maxPayment.Name} за {(maxPayment.Price * maxPayment.Num).ToString("N2")}
руб. от {maxPayment.Date.ToString("dd.MM.yyyy")}";
    }

```

```

        maxPaymentParagraph.set_Style("Подзаголовок");
        maxPaymentRange.Font.Color =
Word.WdColor.wdColorDarkRed;
        maxPaymentRange.InsertParagraphAfter();
    }
    document.Paragraphs.Add();

    // Минимальный платеж
    Payment minPayment = user.Payments.OrderBy(u => u.Price
* u.Num).FirstOrDefault();
    if (minPayment != null)
    {
        Word.Paragraph minPaymentParagraph =
document.Paragraphs.Add();
        Word.Range minPaymentRange =
minPaymentParagraph.Range;
        minPaymentRange.Text = $"Самый дешевый платеж
{minPayment.Name} за {(minPayment.Price * minPayment.Num).ToString("N2")}
руб. от {minPayment.Date.ToString("dd.MM.yyyy")}";
        minPaymentParagraph.set_Style("Подзаголовок");
        minPaymentRange.Font.Color =
Word.WdColor.wdColorDarkGreen;
        minPaymentRange.InsertParagraphAfter();
    }

    // Разрыв страницы
    if (user != allUsers.LastOrDefault())
    {
document.Words.Last.InsertBreak(Word.WdBreakType.wdPageBreak);
    }

    // Колонтитулы
    foreach (Word.Section section in document.Sections)
    {
        // Нижний колонтитул (Номер страницы)
        Word.HeaderFooter footer =
section.Footers[Word.WdHeaderFooterIndex.wdHeaderFooterPrimary];
        footer.PageNumbers.Add(Word.WdPageNumberAlignment.wdAlignPageNumberCenter);

        // Верхний колонтитул (Дата)
        Word.Range headerRange =
section.Headers[Word.WdHeaderFooterIndex.wdHeaderFooterPrimary].Range;
        headerRange.Text = DateTime.Now.ToString("dd.MM.yyyy");
        headerRange.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphCenter;
        headerRange.Font.ColorIndex = Word.WdColorIndex.wdBlack;
        headerRange.Font.Size = 10;
    }

    // (Путь сохранения из методички. Рекомендуется заменить на
SaveFileDialog)
    string savePath =
@"C:\Users\Mekhty\Downloads\Payments.docx";
    document.SaveAs2(savePath);
    document.SaveAs2(@"C:\Users\Mekhty\Downloads\Payments.pdf",
Word.WdExportFormat.wdExportFormatPDF);


    application.Visible = true;

    MessageBox.Show($"Отчеты успешно сохранены по пути:
{savePath}", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при экспорте в
Word:\n{ex.Message}", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        if (application != null)
        {
            application.Quit(false, Missing.Value, Missing.Value);
        }
    }
    finally
    {
        // Освобождение COM-объектов
        if (document != null) Marshal.ReleaseComObject(document);
        if (application != null)
        Marshal.ReleaseComObject(application);
    }
}

//  (Шаг 14) Обработчик для кнопки экспорта в Excel
private void ButtonExcel_Click(object sender, RoutedEventArgs e)
{
    var allUsers = Entities.GetContext().Users.ToList().OrderBy(u =>
u.FIO).ToList();

    Excel.Application application = null;
    Excel.Workbook workbook = null;

    try
    {
        application = new Excel.Application();
        application.SheetsInNewWorkbook = allUsers.Count();
        workbook = application.Workbooks.Add(Type.Missing);

        for (int i = 0; i < allUsers.Count(); i++)
        {
            int startRowIndex = 1;
            Excel.Worksheet worksheet =
(Excel.Worksheet)application.Worksheets.Item[i + 1];
            worksheet.Name = allUsers[i].FIO;

            // Заголовки столбцов
            worksheet.Cells[startRowIndex, 1] = "Дата платежа";
            worksheet.Cells[startRowIndex, 2] = "Название";
            worksheet.Cells[startRowIndex, 3] = "Стоимость";
            worksheet.Cells[startRowIndex, 4] = "Количество";
            worksheet.Cells[startRowIndex, 5] = "Сумма";

            Excel.Range columnHeaderRange =
worksheet.Range[worksheet.Cells[1, 1], worksheet.Cells[1, 5]];
            columnHeaderRange.HorizontalAlignment =
Excel.XlHAlign.xlHAlignCenter;
            columnHeaderRange.Font.Bold = true;
            startRowIndex++;

            var userCategories = allUsers[i].Payments.OrderBy(u =>
u.Date)
                                .GroupBy(u => u.Category)
                                .OrderBy(u => u.Key.Name);

            foreach (var groupCategory in userCategories)
            {
                // Название категории
                Excel.Range headerRange =

```



```

worksheet.Range[worksheet.Cells[startRowIndex, 1],
worksheet.Cells[startRowIndex, 5]];
        headerRange.Merge();
        headerRange.Value = groupCategory.Key.Name;
        headerRange.HorizontalAlignment =
Excel.XlHAlign.xlHAlignCenter;
        headerRange.Font.Italic = true;
        startRowIndex++;

        // Платежи
        foreach (var payment in groupCategory)
        {
            worksheet.Cells[startRowIndex, 1] =
payment.Date.ToString("dd.MM.yyyy");
            worksheet.Cells[startRowIndex, 2] =
payment.Name;
            worksheet.Cells[startRowIndex, 3] =
payment.Price;
            (worksheet.Cells[startRowIndex, 3] as
Excel.Range).NumberFormat = "0.00";
            worksheet.Cells[startRowIndex, 4] = payment.Num;
            worksheet.Cells[startRowIndex, 5].Formula =
$"=C{startRowIndex}*D{startRowIndex}";
            (worksheet.Cells[startRowIndex, 5] as
Excel.Range).NumberFormat = "0.00";
            startRowIndex++;
        }

        // ИТОГО по категории
        Excel.Range sumRange =
worksheet.Range[worksheet.Cells[startRowIndex, 1],
worksheet.Cells[startRowIndex, 4]];
        sumRange.Merge();
        sumRange.Value = "ИТОГО:";
        sumRange.HorizontalAlignment =
Excel.XlHAlign.xlHAlignRight;

        worksheet.Cells[startRowIndex, 5].Formula =
$"=SUM(E{startRowIndex - groupCategory.Count()}:E{startRowIndex - 1})";
        (worksheet.Cells[startRowIndex, 5] as
Excel.Range).NumberFormat = "0.00";

        sumRange.Font.Bold = true;
        worksheet.Cells[startRowIndex, 5].Font.Bold = true;
        startRowIndex++;
    }

    // Границы
    Excel.Range rangeBorders =
worksheet.Range[worksheet.Cells[1, 1], worksheet.Cells[startRowIndex - 1,
5]];

    rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeBottom].LineStyle =
Excel.XlLineStyle.xlContinuous;

    rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeLeft].LineStyle =
Excel.XlLineStyle.xlContinuous;

    rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeRight].LineStyle =
Excel.XlLineStyle.xlContinuous;

    rangeBorders.Borders[Excel.XlBordersIndex.xlEdgeTop].LineStyle =
Excel.XlLineStyle.xlContinuous;

```

```

rangeBorders.Borders[Excel.XlBordersIndex.xlInsideHorizontal].LineStyle =
Excel.XlLineStyle.xlContinuous;

rangeBorders.Borders[Excel.XlBordersIndex.xlInsideVertical].LineStyle =
Excel.XlLineStyle.xlContinuous;

        worksheet.Columns.AutoFit();
    }

    application.Visible = true;
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при экспорте в
Excel:\n{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
    if (workbook != null) workbook.Close(false, Missing.Value,
Missing.Value);
    if (application != null) application.Quit();
}
finally
{
    // Освобождение COM-объектов
    if (workbook != null) Marshal.ReleaseComObject(workbook);
    if (application != null)
Marshal.ReleaseComObject(application);
}
}
}
}

```

Листинг №19 - Код "Dictionary.xaml"

```

<ResourceDictionary
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <!-- Стиль для кнопок -->
    <Style TargetType="Button">
        <Setter Property="Height" Value="35"/>
        <Setter Property="Padding" Value="8"/>
        <Setter Property="Margin" Value="12"/>
        <Setter Property="FontFamily" Value="Tahoma"/>
        <Setter Property="FontSize" Value="12pt"/>
        <Setter Property="Background" Value="#4A90E2"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderThickness" Value="0"/>
        <Setter Property="Cursor" Value="Hand"/>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Background="{TemplateBinding Background}"
                        BorderBrush="{TemplateBinding BorderBrush}"
                        BorderThickness="{TemplateBinding
BorderThickness}"
                        CornerRadius="5">
                        <ContentPresenter HorizontalAlignment="Center"
                            VerticalAlignment="Center"/>
                    </Border>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>

```

```

        <Style.Triggers>
            <Trigger Property="IsMouseOver" Value="True">
                <Setter Property="Background" Value="#357ABD"/>
            </Trigger>
        </Style.Triggers>
    </Style>

    <!-- Стил ь для TextBox -->
    <Style TargetType="TextBox">
        <Setter Property="Height" Value="35"/>
        <Setter Property="Padding" Value="8"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="FontFamily" Value="Tahoma"/>
        <Setter Property="FontSize" Value="12pt"/>
        <Setter Property="VerticalContentAlignment" Value="Center"/>
        <Setter Property="BorderBrush" Value="#CCCCCC"/>
        <Setter Property="BorderThickness" Value="1"/>
    </Style>

    <!-- Стил ь для PasswordBox -->
    <Style TargetType="PasswordBox">
        <Setter Property="Height" Value="35"/>
        <Setter Property="Padding" Value="8"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="FontFamily" Value="Tahoma"/>
        <Setter Property="FontSize" Value="12pt"/>
        <Setter Property="VerticalContentAlignment" Value="Center"/>
        <Setter Property="BorderBrush" Value="#CCCCCC"/>
        <Setter Property="BorderThickness" Value="1"/>
    </Style>

    <!-- Стил ь для Label -->
    <Style TargetType="Label">
        <Setter Property="FontFamily" Value="Tahoma"/>
        <Setter Property="FontSize" Value="12pt"/>
        <Setter Property="VerticalAlignment" Value="Center"/>
        <Setter Property="Margin" Value="5"/>
    </Style>

    <!-- Стил ь для ComboBox -->
    <Style TargetType="ComboBox">
        <Setter Property="Height" Value="35"/>
        <Setter Property="Padding" Value="8"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="FontFamily" Value="Tahoma"/>
        <Setter Property="FontSize" Value="12pt"/>
        <Setter Property="VerticalContentAlignment" Value="Center"/>
    </Style>

    <!-- Стил ь для DataGrid -->
    <Style TargetType="DataGrid">
        <Setter Property="AutoGenerateColumns" Value="False"/>
        <Setter Property="IsReadOnly" Value="True"/>
        <Setter Property="FontFamily" Value="Tahoma"/>
        <Setter Property="FontSize" Value="11pt"/>
        <Setter Property="AlternatingRowBackground" Value="#F5F5F5"/>
        <Setter Property="RowHeight" Value="30"/>
        <Setter Property="GridLinesVisibility" Value="All"/>
        <Setter Property="BorderBrush" Value="#CCCCCC"/>
        <Setter Property="BorderThickness" Value="1"/>
    </Style>

    <!-- Стил ь для CheckBox -->
    <Style TargetType="CheckBox">

```

```

        <Setter Property="FontFamily" Value="Tahoma"/>
        <Setter Property="FontSize" Value="12pt"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="VerticalAlignment" Value="Center"/>
    </Style>

    <!-- Стил ь для TextBlock -->
    <Style TargetType="TextBlock">
        <Setter Property="FontFamily" Value="Tahoma"/>
        <Setter Property="FontSize" Value="12pt"/>
        <Setter Property="Margin" Value="5"/>
    </Style>

</ResourceDictionary>

```

Листинг №20 - Код "MainWindow.xaml"

```

<Window x:Class="Djabarov_Payment.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Система учета платежей"
        Height="600"
        Width="900"
        WindowStartupLocation="CenterScreen"
        Loaded="Window_Loaded"
        Closing="Window_Closing">

    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="45"/>
            <RowDefinition Height="*/>
            <RowDefinition Height="45"/>
        </Grid.RowDefinitions>

        <!-- Верхняя панель -->
        <Grid Grid.Row="0" Background="#4A90E2">
            <Button x:Name="ButtonBack"
                    Content="← Назад"
                    HorizontalAlignment="Left"
                    Width="100"
                    Margin="10,5,0,5"
                    Click="ButtonBack_Click"
                    ToolTip="Вернуться на предыдущую страницу"/>

            <TextBlock Text="Система учета платежей"
                    VerticalAlignment="Center"
                    HorizontalAlignment="Center"
                    FontSize="16"
                    FontWeight="Bold"
                    Foreground="White"/>
        </Grid>

        <!-- Основная область контента -->
        <Frame x:Name="MainFrame"
                Grid.Row="1"
                NavigationUIVisibility="Hidden"
                Navigated="MainFrame_Navigated"/>

        <!-- Нижняя панель -->
        <Grid Grid.Row="2" Background="#F0F0F0">
            <TextBlock x:Name="DateTimeNow"
                    HorizontalAlignment="Right"

```

```

        VerticalAlignment="Center"
        Margin="0,0,10,0"
        FontSize="11"/>
    </Grid>
</Grid>
</Window>

```

Листинг №21 - Код "MainWindow.xaml.cs"

```

using System;
using System.Windows;
using System.Windows.Controls; // Убедитесь, что это есть
using System.Windows.Navigation; // Убедитесь, что это есть
using System.Windows.Threading;
using Djabarov_Payment.Pages;

namespace Djabarov_Payment
{
    /// <summary>
    /// Главное окно приложения
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            MainFrame.Navigate(new AuthPage());
            // (Не нужно дублировать событие здесь, если вы добавили его в
XAML)
            // MainFrame.Navigated += MainFrame_Navigated;
        }

        /// <summary>
        /// Обработчик загрузки окна - инициализация таймера для отображения
времени
        /// </summary>
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            var timer = new DispatcherTimer();
            timer.Interval = new TimeSpan(0, 0, 1);
            timer.IsEnabled = true;
            timer.Tick += (o, t) => { DateTimeNow.Text =
DateTime.Now.ToString("dd.MM.yyyy HH:mm:ss"); };
            timer.Start();
        }

        /// <summary>
        /// Обработчик закрытия окна с подтверждением
        /// </summary>
        private void Window_Closing(object sender,
System.ComponentModel.CancelEventArgs e)
        {
            if (MessageBox.Show("Вы действительно хотите закрыть
приложение?",
                "Подтверждение",
                MessageBoxButton.YesNo,
                MessageBoxImage.Question) == MessageBoxResult.No)
            {
                e.Cancel = true;
            }
            else
            {

```

```

        e.Cancel = false;
    }
}

/// <summary>
/// Обработчик кнопки "Назад"
/// </summary>
private void ButtonBack_Click(object sender, RoutedEventArgs e)
{
    if (MainFrame.CanGoBack)
    {
        MainFrame.GoBack();
    }
}

private void MainFrame_Navigated(object sender, NavigationEventArgs
e)
{
    // Проверяем, что во Frame есть контент и это Page
    if (MainFrame.Content is Page page)
    {
        // Устанавливаем Title окна, используя Title страницы
        this.Title = "Система учета платежей - " + page.Title;
    }
    else
    {
        this.Title = "Система учета платежей";
    }
}
}
}

```

Листинг №22 - Код "PaymentTabPage.xaml"

```

<Page x:Class="Djabarov_Payment.Pages.PaymentTabPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Управление платежами">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="60" />
        </Grid.RowDefinitions>

        <!-- Таблица платежей -->
        <DataGrid Grid.Row="0" x:Name="DataGridPayment"
            AutoGenerateColumns="False"
            IsReadOnly="True"
            SelectionMode="Extended">
            <DataGrid.Columns>
                <DataGridTextColumn Header="ID" Binding="{Binding ID}"
Width="50" />
                <DataGridTextColumn Header="Дата" Binding="{Binding Date,
StringFormat='dd.MM.yyyy'}" Width="100" />
                <DataGridTextColumn Header="Пользователь" Binding="{Binding
Users.FIO}" Width="200" />
                <DataGridTextColumn Header="Название" Binding="{Binding
Name}" Width="150" />
                <DataGridTextColumn Header="Количество" Binding="{Binding
Num}" Width="100" />
                <DataGridTextColumn Header="Цена" Binding="{Binding Price,

```

```

StringFormat='{0:N2} руб.'}" Width="100"/>
    <DataGridTextColumn Header="Категория" Binding="{Binding
Category.Name}" Width="150"/>

    <DataGridTemplateColumn Width="*" Header="Действия">
        <DataGridTemplateColumn.CellTemplate>
            <DataTemplate>
                <Button x:Name="ButtonEdit"
                    Content="Редактировать"
                    ToolTip="Редактировать платёж"
                    Click="ButtonEdit_Click"
                    Margin="5"/>
            </DataTemplate>
        </DataGridTemplateColumn.CellTemplate>
    </DataGridTemplateColumn>
</DataGrid.Columns>
</DataGrid>

<!-- Панель кнопок -->
<Grid Grid.Row="1" Background="#F0F0F0">
    <StackPanel Orientation="Horizontal"
        HorizontalAlignment="Center"
        VerticalAlignment="Center">
        <Button x:Name="ButtonAdd"
            ToolTip="Добавить новый платёж"
            Content="✚ Добавить"
            Width="150"
            Click="ButtonAdd_Click"/>

        <Button x:Name="ButtonDel"
            ToolTip="Удалить выбранные платежи"
            Content="🗑 Удалить"
            Width="150"
            Click="ButtonDel_Click"/>
    </StackPanel>
</Grid>
</Grid>
</Page>

```

Листинг №23 - Код "PaymentTabPage.xaml.cs"

```

using System;
using System.Linq;
using System.Windows;
using System.Windows.Controls;

namespace Djabarov_Payment.Pages
{
    /// <summary>
    /// Страница управления таблицей платежей
    /// </summary>
    public partial class PaymentTabPage : Page
    {
        public PaymentTabPage()
        {
            InitializeComponent();
            DataGridPayment.ItemsSource =
Entities.GetContext().Payment.ToList();
            this.IsVisibleChanged += Page_IsVisibleChanged;
        }

        /// <summary>

```

```

        /// Обновление данных при возврате на страницу
        /// </summary>
        private void Page_IsVisibleChanged(object sender,
DependencyPropertyChangedEventArgs e)
        {
            if (Visibility == Visibility.Visible)
            {

Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(x =>
x.Reload());
                DataGridPayment.ItemsSource =
Entities.GetContext().Payment.ToList();
            }
        }

        /// <summary>
        /// Добавление нового платежа
        /// </summary>
        private void ButtonAdd_Click(object sender, RoutedEventArgs e)
        {
            NavigationService?.Navigate(new AddPaymentPage(null));
        }

        /// <summary>
        /// Удаление выбранных платежей
        /// </summary>
        private void ButtonDel_Click(object sender, RoutedEventArgs e)
        {
            var paymentForRemoving =
DataGridPayment.SelectedItems.Cast<Payment>().ToList();

            if (paymentForRemoving.Count == 0)
            {
                MessageBox.Show("Выберите платежи для удаления!",
                    "Предупреждение", MessageBoxButton.OK,
MessageBoxImage.Warning);
                return;
            }

            if (MessageBox.Show($"Вы точно хотите удалить записи в
количестве {paymentForRemoving.Count()} элементов?",
                "Подтверждение", MessageBoxButton.YesNo,
MessageBoxImage.Question) == MessageBoxResult.Yes)
            {
                try
                {
Entities.GetContext().Payment.RemoveRange(paymentForRemoving);
                    Entities.GetContext().SaveChanges();
                    MessageBox.Show("Данные успешно удалены!",
                        "Успех", MessageBoxButton.OK,
MessageBoxImage.Information);
                    DataGridPayment.ItemsSource =
Entities.GetContext().Payment.ToList();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Ошибка при удалении:\n{ex.Message}",
                        "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
                }
            }
        }
    }
}

```



```

        /// <summary>
        /// Редактирование платежа
        /// </summary>
        private void ButtonEdit_Click(object sender, RoutedEventArgs e)
        {
            NavigationService?.Navigate(new AddPaymentPage((sender as
            Button).DataContext as Payment));
        }
    }
}

```

Листинг №24 - Код "RegPage.xaml"

```

<Page x:Class="Djabarov_Payment.Pages.RegPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="Регистрация">

    <Grid Background="#F5F5F5">
        <ScrollView VerticalScrollBarVisibility="Auto"
                    HorizontalScrollBarVisibility="Disabled">
            <StackPanel MaxWidth="500" Width="320" Margin="20"
                        HorizontalAlignment="Center"
                        VerticalAlignment="Center">

                <TextBlock Text="Регистрация нового пользователя"
                            FontSize="20" FontWeight="Bold"
                            HorizontalAlignment="Center"
                            Margin="0,20,0,30"
                            Foreground="#333"/>

                <Label Content="Логин"/>
                <Grid>
                    <TextBox x:Name="txtbxLog"
                            TextChanged="txtbxLog_TextChanged"/>
                    <Label x:Name="lblLogHitn"
                            Content="Введите логин"
                            Foreground="#999" IsHitTestVisible="False"
                            VerticalAlignment="Center" Padding="12,0,0,0"
                            MouseLeftButtonUp="lblLogHitn_MouseLeftButtonUp"/>
                </Grid>

                <Label Content="Пароль" Margin="0,8,0,0"/>
                <Grid>
                    <PasswordBox x:Name="passBxFrst"
                                PasswordChanged="passBxFrst_PasswordChanged"/>
                    <Label x:Name="lblPassHitn"
                            Content="Введите пароль"
                            Foreground="#999" IsHitTestVisible="False"
                            VerticalAlignment="Center" Padding="12,0,0,0"
                            MouseLeftButtonUp="lblPassHitn_MouseLeftButtonUp"/>
                </Grid>

                <Label Content="Повторите пароль" Margin="0,8,0,0"/>
                <Grid>
                    <PasswordBox x:Name="passBxScnd"
                                PasswordChanged="passBxScnd_PasswordChanged"/>
                    <Label x:Name="lblPassSecHitn"
                            Content="Введите пароль"
                            Foreground="#999" IsHitTestVisible="False"
                            VerticalAlignment="Center" Padding="12,0,0,0"
                            MouseLeftButtonUp="lblPassSecHitn_MouseLeftButtonUp"/>
                </Grid>
            </StackPanel>
        </ScrollView>
    </Grid>

```

```

Content="Повторите пароль"
Foreground="#999" IsHitTestVisible="False"
VerticalAlignment="Center" Padding="12,0,0,0"

MouseLeftButtonUp="lblPassSecHitn_MouseLeftButtonUp"/>
</Grid>

<Label Content="Роль" Margin="0,8,0,0"/>
<ComboBox x:Name="comboBoxRole"
    Margin="5,5,5,0"
    Height="60">
    <ComboBoxItem Content="User"/>
    <ComboBoxItem Content="Admin"/>
</ComboBox>

<Label Content="ФИО" Margin="0,8,0,0"/>
<Grid>
    <TextBox x:Name="textBoxFIO"
        TextChanged="textBoxFIO_TextChanged"/>
    <Label x:Name="lblFioHitn"
        Content="Введите ФИО"
        Foreground="#999" IsHitTestVisible="False"
        VerticalAlignment="Center" Padding="12,0,0,0"

MouseLeftButtonUp="lblFioHitn_MouseLeftButtonUp"/>
</Grid>

<!-- ===== CAPTCHA ===== -->
<Label Content="Введите текст с картинки:"
Margin="0,8,0,0"/>
<StackPanel Orientation="Horizontal"
VerticalAlignment="Center">
    <Image x:Name="imgCaptcha" Width="120" Height="40"
Margin="0,0,10,0"/>
    <Button Content="Обновить"
Click="BtnRefreshCaptcha_Click" Width="90" Height="40"/>
</StackPanel>

<Grid Margin="0,5,0,10">
    <TextBox x:Name="txtCaptchaInput"
        TextChanged="txtCaptchaInput_TextChanged"/>
    <Label x:Name="lblCaptchaHint"
        Content="Введите текст с картинки"
        Foreground="#999"
        IsHitTestVisible="False"
        VerticalAlignment="Center"
        Padding="12,0,0,0"

MouseLeftButtonUp="lblCaptchaHint_MouseLeftButtonUp"/>
</Grid>
<!-- ===== -->

<Button x:Name="regButton"
    Content="Зарегистрироваться"
    Margin="0,20,0,0"
    FontSize="16" FontWeight="SemiBold"
    Click="regButton_Click"
    Cursor="Hand"/>

</StackPanel>
</ScrollView>
</Grid>
</Page>

```

Листинг №25 - Код "RegPage.xaml.cs"

```
using System;
using System.Data.Entity;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;

namespace Djabarov_Payment.Pages
{
    public partial class RegPage : Page
    {
        private string currentCaptcha = "";

        public RegPage()
        {
            InitializeComponent();
            comboBoxRole.SelectedIndex = 0;
            GenerateCaptcha();
        }

        // 🗑️ Хэширование пароля SHA1
        public static string GetHash(string password)
        {
            using (var hash = SHA1.Create())
            {
                return
string.Concat(hash.ComputeHash(Encoding.UTF8.GetBytes(password))
                .Select(x => x.ToString("X2")));
            }
        }

        // 🧠 Генерация капчи (WPF, без System.Drawing)
        private void GenerateCaptcha()
        {
            string chars = "ABCDEFGHJKLMNPQRSTUVWXYZ23456789";
            Random rand = new Random();
            currentCaptcha = new string(Enumerable.Range(0, 5)
                .Select(x => chars[rand.Next(chars.Length)]).ToArray());

            int width = 120;
            int height = 40;

            DrawingVisual visual = new DrawingVisual();
            using (DrawingContext dc = visual.RenderOpen())
            {
                dc.DrawRectangle(Brushes.White, null, new Rect(0, 0, width,
height));

                // шум (линии)
                for (int i = 0; i < 5; i++)
                {
                    dc.DrawLine(
                        new Pen(new SolidColorBrush(Color.FromRgb(
                            (byte)rand.Next(255), (byte)rand.Next(255),
                            (byte)rand.Next(255))), 1),
                        new Point(rand.Next(width), rand.Next(height)),
                        new Point(rand.Next(width), rand.Next(height)));
                }
            }
        }
    }
}
```


```


    }

    // текст капчи
    FormattedText text = new FormattedText(
        currentCaptcha,
        System.Globalization.CultureInfo.InvariantCulture,
        FlowDirection.LeftToRight,
        new Typeface("Arial"),
        24,
        Brushes.Black,
        1.25);

    dc.DrawText(text, new Point(10, 5));
}

RenderTargetBitmap bmp = new RenderTargetBitmap(width, height,
96, 96, PixelFormats.Pbgra32);
bmp.Render(visual);
imgCaptcha.Source = bmp;
}

//  Обновить капчу
private void BtnRefreshCaptcha_Click(object sender, RoutedEventArgs
e)
{
    GenerateCaptcha();
    txtCaptchaInput.Clear();
}

//  Кнопка регистрации
private void regButton_Click(object sender, RoutedEventArgs e)
{
    // Проверка капчи
    if (txtCaptchaInput.Text.Trim().ToUpper() !=
currentCaptcha.ToUpper())
    {
        MessageBox.Show("Неверно введена капча!", "Ошибка",
            MessageBoxButton.OK, MessageBoxImage.Error);
        txtCaptchaInput.Clear();
        GenerateCaptcha();
        return;
    }

    // Проверка заполнения всех полей
    if (string.IsNullOrEmpty(txtbxLog.Text) ||
        string.IsNullOrEmpty(txtbxFIO.Text) ||
        string.IsNullOrEmpty(passBxFrst.Password) ||
        string.IsNullOrEmpty(passBxScnd.Password))
    {
        MessageBox.Show("Заполните все поля!", "Ошибка",
            MessageBoxButton.OK, MessageBoxImage.Warning);
        return;
    }

    // Проверка существующего логина
    using (var db = new Entities())
    {
        var existingUser = db.Users
            .AsNoTracking()
            .FirstOrDefault(u => u.Login == txtbxLog.Text);

        if (existingUser != null)
        {

```

```

        MessageBox.Show("Пользователь с таким логином уже
существует!",
        "Ошибка", MessageBoxButtons.OK,
MessageBoxImage.Error);
        return;
    }

    // Проверка пароля
    if (passBxFrst.Password.Length < 6)
    {
        MessageBox.Show("Пароль слишком короткий (минимум 6
СИМВОЛОВ)!",
        "Ошибка", MessageBoxButtons.OK, MessageBoxImage.Warning);
        return;
    }

    bool en = true;
    bool number = false;

    foreach (char c in passBxFrst.Password)
    {
        if (char.IsDigit(c)) number = true;
        else if (!(c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z')) en
= false;
    }

    if (!en)
    {
        MessageBox.Show("Используйте только английские буквы!",
"Ошибка",
        MessageBoxButtons.OK, MessageBoxImage.Warning);
        return;
    }

    if (!number)
    {
        MessageBox.Show("Добавьте хотя бы одну цифру в пароль!",
"Ошибка",
        MessageBoxButtons.OK, MessageBoxImage.Warning);
        return;
    }

    if (passBxFrst.Password != passBxScnd.Password)
    {
        MessageBox.Show("Пароли не совпадают!", "Ошибка",
        MessageBoxButtons.OK, MessageBoxImage.Warning);
        return;
    }

    //  Добавление пользователя
    using (var db = new Entities())
    {
        User newUser = new User
        {
            FIO = txtbxFIO.Text,
            Login = txtbxLog.Text,
            Password = GetHash(passBxFrst.Password),
            Role = comboBxRole.Text
        };

        db.Users.Add(newUser);
        db.SaveChanges();
    }

```

```

        MessageBox.Show("Пользователь успешно зарегистрирован!",
            "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

        NavigationService?.Navigate(new AuthPage());
    }

    // ◆ Обновление placeholder'ов
    private void txtbxLog_TextChanged(object sender,
        TextChangedEventArgs e)
    {
        lblLogHitn.Visibility = txtbxLog.Text.Length > 0 ?
        Visibility.Hidden : Visibility.Visible;
    }

    private void passBxFrst_PasswordChanged(object sender,
        RoutedEventArgs e)
    {
        lblPassHitn.Visibility = passBxFrst.Password.Length > 0 ?
        Visibility.Hidden : Visibility.Visible;
    }

    private void passBxScnd_PasswordChanged(object sender,
        RoutedEventArgs e)
    {
        lblPassSecHitn.Visibility = passBxScnd.Password.Length > 0 ?
        Visibility.Hidden : Visibility.Visible;
    }

    private void txtbxFIO_TextChanged(object sender,
        TextChangedEventArgs e)
    {
        lblFioHitn.Visibility = txtbxFIO.Text.Length > 0 ?
        Visibility.Hidden : Visibility.Visible;
    }

    private void txtCaptchaInput_TextChanged(object sender,
        TextChangedEventArgs e)
    {
        lblCaptchaHint.Visibility = txtCaptchaInput.Text.Length > 0 ?
        Visibility.Hidden : Visibility.Visible;
    }

    private void lblLogHitn_MouseLeftButtonUp(object sender,
        MouseButtonEventArgs e) => txtbxLog.Focus();
    private void lblPassHitn_MouseLeftButtonUp(object sender,
        MouseButtonEventArgs e) => passBxFrst.Focus();
    private void lblPassSecHitn_MouseLeftButtonUp(object sender,
        MouseButtonEventArgs e) => passBxScnd.Focus();
    private void lblFioHitn_MouseLeftButtonUp(object sender,
        MouseButtonEventArgs e) => txtbxFIO.Focus();
    private void lblCaptchaHint_MouseLeftButtonUp(object sender,
        MouseButtonEventArgs e) => txtCaptchaInput.Focus();
    }
}

```

Листинг №26 - Код "UserPage.xaml"

```

<Page x:Class="Djabarov_Payment.Pages.UserPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Пользователи">

```

```

<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>

    <!-- Панель фильтров -->
    <WrapPanel Orientation="Horizontal" HorizontalAlignment="Center"
Margin="10">
        <TextBlock Text="Поиск по ФИО:" FontSize="11"
VerticalAlignment="Center"/>
        <TextBox x:Name="fioFilterTextBox" Margin="7,7,0,7"
MinWidth="150"
                TextChanged="fioFilterTextBox_TextChanged"/>

        <TextBlock Text="Сортировка по ФИО:" Margin="10,0,0,0"
FontSize="11"
                VerticalAlignment="Center"/>
        <ComboBox SelectionChanged="sortComboBox_SelectionChanged"
Margin="7,0,0,0" x:Name="sortComboBox"
SelectedIndex="0" Height="50"
                VerticalContentAlignment="Center"
VerticalAlignment="Center">
            <ComboBoxItem Content="По возрастанию"/>
            <ComboBoxItem Content="По убыванию"/>
        </ComboBox>

        <CheckBox IsChecked="False"
                Checked="onlyAdminCheckBox_Checked"
                Unchecked="onlyAdminCheckBox_Unchecked"
                Content="Только администраторы"
                x:Name="onlyAdminCheckBox"
                VerticalContentAlignment="Center" Margin="10,0,0,0"/>

        <Button x:Name="clearFiltersButton" Content="Очистить фильтр"
                VerticalAlignment="Center" HorizontalAlignment="Center"
                Padding="5,8,5,8" Height="34" FontSize="10"
                Click="clearFiltersButton_Click"/>
    </WrapPanel>

    <!-- Список пользователей -->
    <ListView Name="ListUser" Grid.Row="1"
                HorizontalContentAlignment="Center"
                ScrollViewer.HorizontalScrollBarVisibility="Disabled">
        <ListView.ItemsPanel>
            <ItemsPanelTemplate>
                <WrapPanel Orientation="Horizontal"
HorizontalAlignment="Center"/>
            </ItemsPanelTemplate>
        </ListView.ItemsPanel>

        <ListView.ItemTemplate>
            <DataTemplate>
                <Border BorderBrush="#CCCCCC" BorderThickness="1"
                        CornerRadius="5" Margin="10" Padding="10"
                        Width="220" Height="280">
                    <Grid>
                        <Grid.RowDefinitions>
                            <RowDefinition Height="auto"/>
                            <RowDefinition Height="180"/>
                            <RowDefinition Height="auto"/>
                        </Grid.RowDefinitions>

                        <TextBlock Grid.Row="0" TextAlignment="Center"

```

```

Width="200" TextWrapping="Wrap"
Text="{Binding FIO}" FontSize="16"
FontWeight="Bold" Margin="0,0,0,10"/>

<Image Width="180" Height="180" Grid.Row="1"
Stretch="UniformToFill">
<Image.Source>
<Binding Path="Photo">
<!--<Binding.TargetNullValue>

<ImageSource>pack://application:,,,/Resources/default.png</ImageSource>
</Binding.TargetNullValue>-->
</Binding>
</Image.Source>
</Image>

<TextBlock Grid.Row="2" TextAlignment="Center"
Width="200" FontSize="14"
Text="{Binding Role}"
Margin="0,10,0,0"/>
</Grid>
</Border>
</DataTemplate>
</ListView.ItemTemplate>
</ListView>
</Grid>
</Page>

```

Листинг №27 - Код "UserPage.xaml.cs"

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows;
using System.Windows.Controls;
using Djabarov_Payment.Pages;

namespace Djabarov_Payment.Pages
{
    /// <summary>
    /// Страница отображения пользователей с фильтрацией
    /// </summary>
    public partial class UserPage : Page
    {
        public UserPage()
        {
            InitializeComponent();
            var currentUsers = Entities.GetContext().Users.ToList();
            ListUser.ItemsSource = currentUsers;
        }

        /// <summary>
        /// Очистка всех фильтров
        /// </summary>
        private void clearFiltersButton_Click(object sender, RoutedEventArgs
e)
        {
            fioFilterTextBox.Text = "";
            sortComboBox.SelectedIndex = 0;
            onlyAdminCheckBox.IsChecked = false;
        }
    }
}

```



```

        /// <summary>
        /// Обработчик изменения текста поиска
        /// </summary>
        private void fioFilterTextBox_TextChanged(object sender,
TextChangedEventArgs e)
        {
            UpdateUsers();
        }

        /// <summary>
        /// Обработчик изменения сортировки
        /// </summary>
        private void sortComboBox_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {
            UpdateUsers();
        }

        /// <summary>
        /// Обработчик включения фильтра по администраторам
        /// </summary>
        private void onlyAdminCheckBox_Checked(object sender,
RoutedEventArgs e)
        {
            UpdateUsers();
        }

        /// <summary>
        /// Обработчик выключения фильтра по администраторам
        /// </summary>
        private void onlyAdminCheckBox_Unchecked(object sender,
RoutedEventArgs e)
        {
            UpdateUsers();
        }

        /// <summary>
        /// Метод обновления списка пользователей с применением фильтров
        /// </summary>
        private void UpdateUsers()
        {
            if (!IsInitialized)
            {
                return;
            }

            try
            {
                List<User> currentUsers =
Entities.GetContext().Users.ToList();

                // Фильтрация по ФИО
                if (!string.IsNullOrEmpty(fioFilterTextBox.Text))
                {
                    currentUsers = currentUsers.Where(x =>
x.FIO.ToLower().Contains(fioFilterTextBox.Text.ToLower())).ToList();
                }

                // Фильтрация по роли
                if (onlyAdminCheckBox.IsChecked.Value)
                {
                    currentUsers = currentUsers.Where(x => x.Role ==
"Admin").ToList();
                }
            }
            catch { }
        }

```

```

    }

    // Сортировка по ФИО
    ListUser.ItemsSource = (sortComboBox.SelectedIndex == 0) ?
        currentUsers.OrderBy(x => x.FIO).ToList() :
        currentUsers.OrderByDescending(x => x.FIO).ToList();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при обновлении списка:
{ex.Message}",
            "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
    }
}
}
}

```

Листинг №28 - Код "UsersTabPage.xaml"

```

<Page x:Class="Djabarov_Payment.Pages.UsersTabPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Управление пользователями">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="60" />
        </Grid.RowDefinitions>

        <!-- Таблица пользователей -->
        <DataGrid Grid.Row="0" x:Name="DataGridUser"
            AutoGenerateColumns="False"
            IsReadOnly="True"
            SelectionMode="Extended">
            <DataGrid.Columns>
                <DataGridTextColumn Header="ID" Binding="{Binding ID}"
Width="50"/>
                <DataGridTextColumn Header="Логин" Binding="{Binding Login}"
Width="150"/>
                <DataGridTextColumn Header="Пароль (хеш)" Binding="{Binding
Password}" Width="250"/>
                <DataGridTextColumn Header="Роль" Binding="{Binding Role}"
Width="100"/>
                <DataGridTextColumn Header="ФИО" Binding="{Binding FIO}"
Width="250"/>
                <DataGridTextColumn Header="Фото" Binding="{Binding Photo}"
Width="150"/>

                <DataGridTemplateColumn Width="*" Header="Действия">
                    <DataGridTemplateColumn.CellTemplate>
                        <DataTemplate>
                            <Button x:Name="ButtonEdit"
                                Content="Редактировать"
                                ToolTip="Редактировать данные
пользователя"

                                Click="ButtonEdit_Click"
                                Margin="5"/>
                        </DataTemplate>
                    </DataGridTemplateColumn.CellTemplate>
                </DataGridTemplateColumn>
            </DataGrid.Columns>
        </DataGrid>
    </Grid>

```

```

        <!-- Панель кнопок -->
        <Grid Grid.Row="1" Background="#F0F0F0">
            <StackPanel Orientation="Horizontal"
                HorizontalAlignment="Center"
                VerticalAlignment="Center">
                <Button x:Name="ButtonAdd"
                    ToolTip="Добавить нового пользователя"
                    Content="✚ Добавить"
                    Width="150"
                    Click="ButtonAdd_Click"/>

                <Button x:Name="ButtonDel"
                    ToolTip="Удалить выбранных пользователей"
                    Content="🗑 Удалить"
                    Width="150"
                    Click="ButtonDel_Click"/>
            </StackPanel>
        </Grid>
    </Grid>
</Page>

```

Листинг №29 - Код "UsersTabPage.xaml.cs"

```

using System;
using System.Linq;
using System.Windows;
using System.Windows.Controls;

namespace Djabarov_Payment.Pages
{
    /// <summary>
    /// Страница управления таблицей пользователей
    /// </summary>
    public partial class UsersTabPage : Page
    {
        public UsersTabPage()
        {
            InitializeComponent();
            DataGridUser.ItemsSource = Entities.GetContext().Users.ToList();
            this.IsVisibleChanged += Page_IsVisibleChanged;
        }

        /// <summary>
        /// Обновление данных при возврате на страницу
        /// </summary>
        private void Page_IsVisibleChanged(object sender,
            DependencyPropertyChangedEventArgs e)
        {
            if (Visibility == Visibility.Visible)
            {
                Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(x =>
                    x.Reload());
                DataGridUser.ItemsSource =
                Entities.GetContext().Users.ToList();
            }
        }

        /// <summary>
        /// Добавление нового пользователя
        /// </summary>
    }
}

```

```

        private void ButtonAdd_Click(object sender, RoutedEventArgs e)
        {
            NavigationService?.Navigate(new AddUserPage(null));
        }

        /// <summary>
        /// Удаление выбранных пользователей
        /// </summary>
        private void ButtonDel_Click(object sender, RoutedEventArgs e)
        {
            var usersForRemoving =
DataGridUser.SelectedItems.Cast<User>().ToList();

            if (usersForRemoving.Count == 0)
            {
                MessageBox.Show("Выберите пользователей для удаления!",
                    "Предупреждение", MessageBoxButton.OK,
MessageBoxImage.Warning);
                return;
            }

            if (MessageBox.Show($"Вы точно хотите удалить записи в
количестве {usersForRemoving.Count()} элементов?",
                "Подтверждение", MessageBoxButton.YesNo,
MessageBoxImage.Question) == MessageBoxResult.Yes)
            {
                try
                {
Entities.GetContext().Users.RemoveRange(usersForRemoving);
                    Entities.GetContext().SaveChanges();
                    MessageBox.Show("Данные успешно удалены!",
                        "Успех", MessageBoxButton.OK,
MessageBoxImage.Information);
                    DataGridUser.ItemsSource =
Entities.GetContext().Users.ToList();
                }
                catch (Exception ex)
                {
                    MessageBox.Show($"Ошибка при удалении:\n{ex.Message}",
                        "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
                }
            }
        }

        /// <summary>
        /// Редактирование пользователя
        /// </summary>
        private void ButtonEdit_Click(object sender, RoutedEventArgs e)
        {
            NavigationService?.Navigate(new AddUserPage((sender as
Button).DataContext as User));
        }
    }
}

```