

**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.02 Осуществление интеграции программных модулей
(код) (наименование)

Междисциплинарный курс МДК.02.01 Технология разработки программного обеспечения
(код) (наименование)

Выполнил:

Студент 4 курса 4ИСИП-522 учебной группы
(номер) (номер)
Мед Д.В. Медведева
(подпись) (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	Анализ выбранной предметной области	5
2	Разработка и оформление технического задания на программное средство	5
3	Построение архитектуры программного средства	5
4	Анализ требований и построение необходимых диаграмм	5

Ход работы:

Создадим базу данных «Medevedeva_DB_UP2» состоящую из трех таблиц и заполним их, также создадим диаграмму (Рисунки 1 - 13).

	Column Name	Data Type	Allow Nulls
▼	AdID	int	<input type="checkbox"/>
	UserID	int	<input checked="" type="checkbox"/>
	Title	nvarchar(255)	<input type="checkbox"/>
	Description	nvarchar(MAX)	<input checked="" type="checkbox"/>
	PostDate	datetime	<input type="checkbox"/>
	CityID	int	<input checked="" type="checkbox"/>
	CategoryID	int	<input checked="" type="checkbox"/>
	AdTypeID	int	<input checked="" type="checkbox"/>
	StatusID	int	<input checked="" type="checkbox"/>
	Price	money	<input checked="" type="checkbox"/>
	IsActive	bit	<input checked="" type="checkbox"/>
	ImagePath	nvarchar(MAX)	<input checked="" type="checkbox"/>

Рисунок 1 – Таблица «Ads»

	Column Name	Data Type	Allow Nulls
▼	AdTypeID	int	<input type="checkbox"/>
	TypeDescription	nvarchar(50)	<input type="checkbox"/>
▶			<input type="checkbox"/>

Рисунок 2 – Таблица «AdTypes»

	Column Name	Data Type	Allow Nulls
▼	CategoryID	int	<input type="checkbox"/>
	CategoryName	nvarchar(50)	<input type="checkbox"/>

Рисунок 3 – Таблица «Categories»

	Column Name	Data Type	Allow Nulls
▼	CityID	int	<input type="checkbox"/>
	CityName	nvarchar(50)	<input type="checkbox"/>

Рисунок 4 – Таблица «Cities»

	Column Name	Data Type	Allow Nulls
▼	StatusID	int	<input type="checkbox"/>
	StatusName	nvarchar(50)	<input type="checkbox"/>

Рисунок 5– Таблица «Statuses»

	Column Name	Data Type	Allow Nulls
▼	UserID	int	<input type="checkbox"/>
	Login	nvarchar(50)	<input type="checkbox"/>
	PasswordHash	nvarchar(100)	<input type="checkbox"/>
	TotalProfit	money	<input checked="" type="checkbox"/>

Рисунок 6 – Таблица «Users»

	AdID	UserID	Title	Description	PostDate	CityID	CategoryID	AdTypeID	StatusID	Price	IsActive	ImagePath
1	1	10	Продажа дома	Участок 10 соток, все коммуникации	2025-05-04 00:00:00.000	4	2	2	1	21362,00	1	NULL
2	2	20	Уборка квартир	Генеральная уборка, моем окна	2025-04-26 00:00:00.000	3	4	3	2	87371,00	1	NULL
3	3	8	Уборка квартир	Генеральная уборка, моем окна	2025-03-09 00:00:00.000	2	4	1	2	85947,00	1	NULL
4	4	11	Продам iPhone 13	Состояние отличное, полный комплект	2025-04-19 00:00:00.000	3	5	1	1	47780,00	1	NULL
5	5	4	Продам ВАЗ 2107	Хорошее состояние, новая резина	2025-03-25 00:00:00.000	1	1	1	1	51527,00	1	NULL
6	6	16	Услуги электрика	Быстро и качественно. Гарантия.	2025-03-08 00:00:00.000	3	4	1	2	25184,00	1	NULL
7	7	12	Продам ВАЗ 2107	Хорошее состояние, новая резина	2025-03-09 00:00:00.000	5	1	1	1	89389,00	1	NULL
8	8	10	Сдам квартиру	2-комнатная, рядом с метро	2025-03-28 00:00:00.000	3	2	2	1	64047,00	1	NULL

Рисунок 7 – Данные таблицы «Ads»

	AdTypeID	TypeDescription
1	1	Аренда
2	2	Продажа
3	3	Услуга

Рисунок 8 – Данные таблицы «AdTypes»

	CategoryID	CategoryName
1	1	Авто
2	2	Недвижимость
3	3	Одежда
4	4	Услуги
5	5	Электроника

Рисунок 9 – Данные таблицы «Categories»

	CityID	CityName
1	1	Екатеринбург
2	2	Казань
3	3	Москва
4	4	Новосибирск
5	5	Санкт-Пете...

Рисунок 10 – Данные таблицы «Cities»

	StatusID	StatusName
1	1	Активно
2	2	Завершено

Рисунок 11 – Данные таблицы «Statuses»

	UserID	Login	PasswordHash	TotalProfit
1	1	user1	pass1	48884,00
2	2	user10	pass10	0,00
3	3	user11	pass11	0,00
4	4	user12	pass12	0,00
5	5	user13	pass13	0,00
6	6	user14	pass14	0,00
7	7	user15	pass15	0,00
8	8	user16	pass16	0,00

Рисунок 12 – Данные таблицы «Users»



Рисунок 13 – ER-диаграмма

Теперь создадим скрипт базы данных с заполненными таблицами. Продемонстрировано на листинге 1.

Листинг 1 – скрипт базы данных

```

GO
/***** Object: Table [dbo].[Ads]    Script Date: 17.11.2025 16:12:18 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Ads](
    [AdID] [int] IDENTITY(1,1) NOT NULL,
    [UserID] [int] NULL,

```

```

        [Title] [nvarchar](255) NOT NULL,
        [Description] [nvarchar](max) NULL,
        [PostDate] [datetime] NOT NULL,
        [CityID] [int] NULL,
        [CategoryID] [int] NULL,
        [AdTypeID] [int] NULL,
        [StatusID] [int] NULL,
        [Price] [money] NULL,
        [IsActive] [bit] NULL,
PRIMARY KEY CLUSTERED
(
    [AdID] 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] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[AdTypes]    Script Date: 17.11.2025 16:12:18 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AdTypes](
    [AdTypeID] [int] IDENTITY(1,1) NOT NULL,
    [TypeDescription] [nvarchar](50) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [AdTypeID] 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].[Categories]    Script Date: 17.11.2025 16:12:18 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Categories](
    [CategoryID] [int] IDENTITY(1,1) NOT NULL,
    [CategoryName] [nvarchar](50) NOT NULL,

```

```

PRIMARY KEY CLUSTERED
(
    [CategoryID] 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].[Cities]    Script Date: 17.11.2025 16:12:18 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Cities](
    [CityID] [int] IDENTITY(1,1) NOT NULL,
    [CityName] [nvarchar](50) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [CityID] 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].[Statuses]    Script Date: 17.11.2025 16:12:18 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Statuses](
    [StatusID] [int] IDENTITY(1,1) NOT NULL,
    [StatusName] [nvarchar](50) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [StatusID] 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: 17.11.2025 16:12:18 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Users](
    [UserID] [int] IDENTITY(1,1) NOT NULL,
    [Login] [nvarchar](50) NOT NULL,
    [PasswordHash] [nvarchar](100) NOT NULL,
    [TotalProfit] [money] NULL,
PRIMARY KEY CLUSTERED
(
    [UserID] 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
SET IDENTITY_INSERT [dbo].[Ads] ON

INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (1, 10, N'Продажа
дома', N'Участок 10 соток, все коммуникации', CAST(N'2025-05-04T00:00:00.000' AS
DateTime), 4, 2, 2, 1, 21362.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (2, 20, N'Уборка
квартир', N'Генеральная уборка, моем окна', CAST(N'2025-04-26T00:00:00.000' AS
DateTime), 3, 4, 3, 2, 87371.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (3, 8, N'Уборка
квартир', N'Генеральная уборка, моем окна', CAST(N'2025-03-09T00:00:00.000' AS
DateTime), 2, 4, 1, 2, 85947.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (4, 11, N'Продам
iPhone 13', N'Состояние отличное, полный комплект', CAST(N'2025-04-19T00:00:00.000'
AS DateTime), 3, 5, 1, 1, 47780.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (5, 4, N'Продам
BA3 2107', N'Хорошее состояние, новая резина', CAST(N'2025-03-25T00:00:00.000' AS
DateTime), 1, 1, 1, 1, 51527.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (6, 16, N'Услуги

```



```
электрика', N'Быстро и качественно. Гарантия.', CAST(N'2025-03-08T00:00:00.000' AS
DateTime), 3, 4, 1, 1, 25184.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (7, 12, N'Продам
BA3 2107', N'Хорошее состояние, новая резина', CAST(N'2025-03-09T00:00:00.000' AS
DateTime), 5, 1, 1, 1, 89389.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (8, 10, N'Сдам
квартиру', N'2-комнатная, рядом с метро', CAST(N'2025-03-28T00:00:00.000' AS
DateTime), 3, 2, 2, 1, 64047.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (9, 7, N'Продам
куртку', N'Новая зимняя куртка, размер М', CAST(N'2025-03-19T00:00:00.000' AS
DateTime), 1, 3, 1, 1, 39483.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (10, 6, N'Аренда
авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-03-18T00:00:00.000' AS
DateTime), 4, 1, 1, 1, 41496.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (11, 15, N'Аренда
авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-04-09T00:00:00.000' AS
DateTime), 5, 1, 1, 2, 28813.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (12, 16, N'Услуги
электрика', N'Быстро и качественно. Гарантия.', CAST(N'2025-03-23T00:00:00.000' AS
DateTime), 2, 4, 3, 1, 1808.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (13, 1, N'Продам
iPhone 13', N'Состояние отличное, полный комплект', CAST(N'2025-04-30T00:00:00.000'
AS DateTime), 5, 5, 2, 2, 26117.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (14, 8, N'Услуги
электрика', N'Быстро и качественно. Гарантия.', CAST(N'2025-04-10T00:00:00.000' AS
DateTime), 2, 4, 3, 1, 73353.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (15, 11, N'Услуги
электрика', N'Быстро и качественно. Гарантия.', CAST(N'2025-03-20T00:00:00.000' AS
DateTime), 3, 4, 3, 2, 20177.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (16, 9, N'Аренда
```

```
авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-03-17T00:00:00.000' AS
DateTime), 4, 1, 1, 1, 27648.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (17, 10, N'Продам
куртку', N'Новая зимняя куртка, размер М', CAST(N'2025-04-30T00:00:00.000' AS
DateTime), 1, 3, 1, 1, 10129.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (18, 10, N'Ноутбук
Acer', N'Core i7, SSD 512 ГБ, как новый', CAST(N'2025-04-14T00:00:00.000' AS
DateTime), 4, 5, 2, 1, 29522.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (19, 17, N'Аренда
авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-03-15T00:00:00.000' AS
DateTime), 3, 1, 3, 1, 40477.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (20, 17, N'Джинсы
мужские', N'Состояние хорошее, размер 32', CAST(N'2025-03-21T00:00:00.000' AS
DateTime), 3, 3, 3, 2, 50531.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (21, 7, N'Аренда
авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-04-15T00:00:00.000' AS
DateTime), 4, 1, 2, 1, 60821.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (22, 15, N'Продажа
дома', N'Участок 10 соток, все коммуникации', CAST(N'2025-05-03T00:00:00.000' AS
DateTime), 4, 2, 1, 1, 34058.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (23, 17, N'Продажа
дома', N'Участок 10 соток, все коммуникации', CAST(N'2025-04-30T00:00:00.000' AS
DateTime), 2, 2, 1, 2, 31878.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (24, 9, N'Продам
iPhone 13', N'Состояние отличное, полный комплект', CAST(N'2025-04-07T00:00:00.000'
AS DateTime), 3, 5, 1, 2, 61358.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (25, 4, N'Продам
BA3 2107', N'Хорошее состояние, новая резина', CAST(N'2025-03-07T00:00:00.000' AS
DateTime), 4, 1, 2, 2, 81792.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (26, 11, N'Услуги
```

```
электрика', N'Быстро и качественно. Гарантия.', CAST(N'2025-04-24T00:00:00.000' AS  
DateTime), 2, 4, 3, 2, 11113.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (27, 18, N'Аренда  
авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-03-07T00:00:00.000' AS  
DateTime), 1, 1, 1, 1, 70726.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (28, 13, N'Услуги  
электрика', N'Быстро и качественно. Гарантия.', CAST(N'2025-03-19T00:00:00.000' AS  
DateTime), 3, 4, 2, 1, 65931.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (29, 16, N'Уборка  
квартир', N'Генеральная уборка, моем окна', CAST(N'2025-03-20T00:00:00.000' AS  
DateTime), 3, 4, 2, 2, 57170.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (30, 8, N'Услуги  
электрика', N'Быстро и качественно. Гарантия.', CAST(N'2025-04-02T00:00:00.000' AS  
DateTime), 2, 4, 3, 1, 29335.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (31, 17, N'Аренда  
авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-04-26T00:00:00.000' AS  
DateTime), 4, 1, 3, 2, 15509.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (32, 1, N'Продам  
куртку', N'Новая зимняя куртка, размер М', CAST(N'2025-03-11T00:00:00.000' AS  
DateTime), 5, 3, 3, 1, 22767.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (33, 16, N'Джинсы  
мужские', N'Состояние хорошее, размер 32', CAST(N'2025-03-08T00:00:00.000' AS  
DateTime), 1, 3, 2, 1, 1484.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (34, 18, N'Уборка  
квартир', N'Генеральная уборка, моем окна', CAST(N'2025-04-29T00:00:00.000' AS  
DateTime), 1, 4, 2, 2, 77129.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (35, 14, N'Ноутбук  
Acer', N'Core i7, SSD 512 ГБ, как новый', CAST(N'2025-05-01T00:00:00.000' AS  
DateTime), 2, 5, 1, 1, 51598.0000, 1)  
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],  
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (36, 16, N'Продам
```

```
iPhone 13', N'Состояние отличное, полный комплект', CAST(N'2025-03-30T00:00:00.000' AS DateTime), 2, 5, 2, 1, 33265.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (37, 5, N'Уборка квартир', N'Генеральная уборка, моем окна', CAST(N'2025-03-12T00:00:00.000' AS DateTime), 1, 4, 2, 2, 95719.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (38, 2, N'Аренда авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-04-30T00:00:00.000' AS DateTime), 4, 1, 2, 1, 88532.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (39, 5, N'Сдам квартиру', N'2-комнатная, рядом с метро', CAST(N'2025-03-20T00:00:00.000' AS DateTime), 3, 2, 2, 2, 82786.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (40, 5, N'Уборка квартир', N'Генеральная уборка, моем окна', CAST(N'2025-03-08T00:00:00.000' AS DateTime), 5, 4, 3, 2, 22411.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (41, 3, N'Аренда авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-04-10T00:00:00.000' AS DateTime), 4, 1, 1, 1, 13740.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (42, 12, N'Аренда авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-04-28T00:00:00.000' AS DateTime), 1, 1, 3, 1, 2132.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (43, 13, N'Услуги электрика', N'Быстро и качественно. Гарантия.', CAST(N'2025-04-18T00:00:00.000' AS DateTime), 4, 4, 3, 2, 13627.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (44, 10, N'Уборка квартир', N'Генеральная уборка, моем окна', CAST(N'2025-04-13T00:00:00.000' AS DateTime), 4, 4, 3, 1, 79256.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (45, 9, N'Продам ВАЗ 2107', N'Хорошее состояние, новая резина', CAST(N'2025-03-29T00:00:00.000' AS DateTime), 5, 1, 3, 2, 33644.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID], [CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (46, 14, N'Продажа
```

```

дома', N'Участок 10 соток, все коммуникации', CAST(N'2025-03-25T00:00:00.000' AS
DateTime), 2, 2, 1, 1, 28414.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (47, 12, N'Ноутбук
Acer', N'Core i7, SSD 512 ГБ, как новый', CAST(N'2025-04-16T00:00:00.000' AS
DateTime), 3, 5, 1, 2, 6426.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (48, 19, N'Джинсы
мужские', N'Состояние хорошее, размер 32', CAST(N'2025-04-01T00:00:00.000' AS
DateTime), 4, 3, 1, 1, 10988.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (49, 12, N'Аренда
авто', N'Почасовая аренда, документы в порядке', CAST(N'2025-04-03T00:00:00.000' AS
DateTime), 3, 1, 3, 1, 11283.0000, 1)
INSERT [dbo].[Ads] ([AdID], [UserID], [Title], [Description], [PostDate], [CityID],
[CategoryID], [AdTypeID], [StatusID], [Price], [IsActive]) VALUES (50, 6, N'Сдам
квартиру', N'2-комнатная, рядом с метро', CAST(N'2025-04-11T00:00:00.000' AS
DateTime), 4, 2, 1, 2, 96721.0000, 1)
SET IDENTITY_INSERT [dbo].[Ads] OFF
GO
SET IDENTITY_INSERT [dbo].[AdTypes] ON

INSERT [dbo].[AdTypes] ([AdTypeID], [TypeDescription]) VALUES (1, N'Аренда')
INSERT [dbo].[AdTypes] ([AdTypeID], [TypeDescription]) VALUES (2, N'Продажа')
INSERT [dbo].[AdTypes] ([AdTypeID], [TypeDescription]) VALUES (3, N'Услуга')
SET IDENTITY_INSERT [dbo].[AdTypes] OFF
GO
SET IDENTITY_INSERT [dbo].[Categories] ON

INSERT [dbo].[Categories] ([CategoryID], [CategoryName]) VALUES (1, N'Авто')
INSERT [dbo].[Categories] ([CategoryID], [CategoryName]) VALUES (2, N'Недвижимость')
INSERT [dbo].[Categories] ([CategoryID], [CategoryName]) VALUES (3, N'Одежда')
INSERT [dbo].[Categories] ([CategoryID], [CategoryName]) VALUES (4, N'Услуги')
INSERT [dbo].[Categories] ([CategoryID], [CategoryName]) VALUES (5, N'Электроника')
SET IDENTITY_INSERT [dbo].[Categories] OFF
GO
SET IDENTITY_INSERT [dbo].[Cities] ON

INSERT [dbo].[Cities] ([CityID], [CityName]) VALUES (1, N'Екатеринбург')
INSERT [dbo].[Cities] ([CityID], [CityName]) VALUES (2, N'Казань')
INSERT [dbo].[Cities] ([CityID], [CityName]) VALUES (3, N'Москва')

```

```
INSERT [dbo].[Cities] ([CityID], [CityName]) VALUES (4, N'Новосибирск')
INSERT [dbo].[Cities] ([CityID], [CityName]) VALUES (5, N'Санкт-Петербург')
SET IDENTITY_INSERT [dbo].[Cities] OFF
GO
SET IDENTITY_INSERT [dbo].[Statuses] ON

INSERT [dbo].[Statuses] ([StatusID], [StatusName]) VALUES (1, N'Активно')
INSERT [dbo].[Statuses] ([StatusID], [StatusName]) VALUES (2, N'Завершено')
SET IDENTITY_INSERT [dbo].[Statuses] OFF
GO
SET IDENTITY_INSERT [dbo].[Users] ON

INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (1,
N'user1', N'pass1', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (2,
N'user10', N'pass10', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (3,
N'user11', N'pass11', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (4,
N'user12', N'pass12', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (5,
N'user13', N'pass13', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (6,
N'user14', N'pass14', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (7,
N'user15', N'pass15', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (8,
N'user16', N'pass16', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (9,
N'user17', N'pass17', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (10,
N'user18', N'pass18', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (11,
N'user19', N'pass19', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (12,
N'user2', N'pass2', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (13,
N'user20', N'pass20', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (14,
N'user3', N'pass3', 0.0000)
```

```

INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (15,
N'user4', N'pass4', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (16,
N'user5', N'pass5', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (17,
N'user6', N'pass6', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (18,
N'user7', N'pass7', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (19,
N'user8', N'pass8', 0.0000)
INSERT [dbo].[Users] ([UserID], [Login], [PasswordHash], [TotalProfit]) VALUES (20,
N'user9', N'pass9', 0.0000)
SET IDENTITY_INSERT [dbo].[Users] OFF
GO
SET ANSI_PADDING ON
GO
/***** Object:      Index [UQ__Users__5E55825BD9D61570]      Script Date: 17.11.2025
16:12:18 *****/
ALTER TABLE [dbo].[Users] ADD UNIQUE NONCLUSTERED
(
    [Login] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Ads] ADD  DEFAULT ((1)) FOR [IsActive]
GO
ALTER TABLE [dbo].[Users] ADD  DEFAULT ((0)) FOR [TotalProfit]
GO
ALTER TABLE [dbo].[Ads]  WITH CHECK ADD FOREIGN KEY([AdTypeID])
REFERENCES [dbo].[AdTypes] ([AdTypeID])
GO
ALTER TABLE [dbo].[Ads]  WITH CHECK ADD FOREIGN KEY([CategoryID])
REFERENCES [dbo].[Categories] ([CategoryID])
GO
ALTER TABLE [dbo].[Ads]  WITH CHECK ADD FOREIGN KEY([CityID])
REFERENCES [dbo].[Cities] ([CityID])
GO
ALTER TABLE [dbo].[Ads]  WITH CHECK ADD FOREIGN KEY([StatusID])
REFERENCES [dbo].[Statuses] ([StatusID])
GO

```

```
ALTER TABLE [dbo].[Ads] WITH CHECK ADD FOREIGN KEY([UserID])
REFERENCES [dbo].[Users] ([UserID])
GO
```

Далее представим две блок-схемы алгоритм подсистемы работы с объявлениями и алгоритм функции подсчета прибыли от продажи товаров при изменении статуса объявления на «Завершено» (Рисунки 14 - 15).

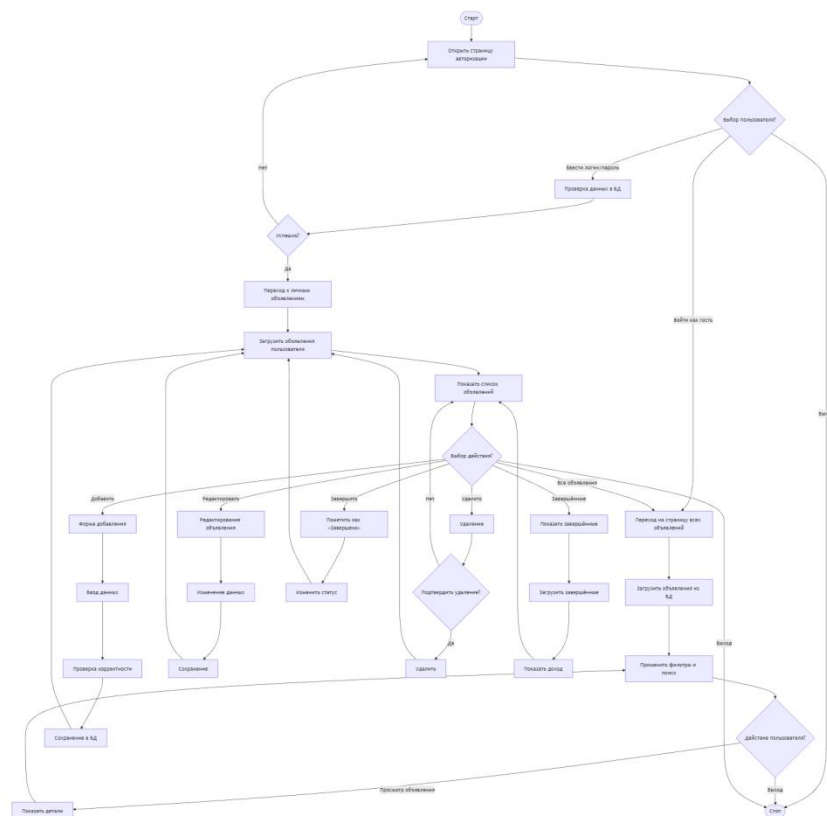


Рисунок 14 – Алгоритм подсистемы для работы с объявлениями

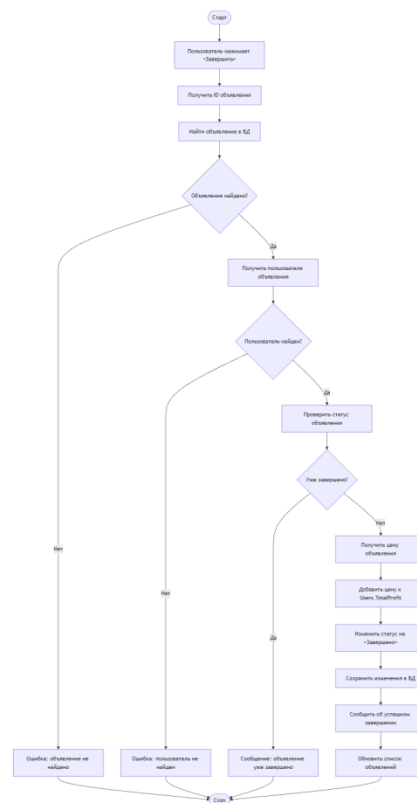


Рисунок 15 – Алгоритм функции подсчета прибыли от продажи товаров при изменении статуса объявления на «Завершено».

По завершении разработки базы данных приступим к этапу разработки интерфейса в Microsoft Visual Studio.

Процесс разработки приложения начинается с подключения базы данных к проекту. Подключение происходит как добавление нового элемента в проект, где происходит настройка подключаемых данных. Процесс подключения показан на Рисунках 16 и 17.

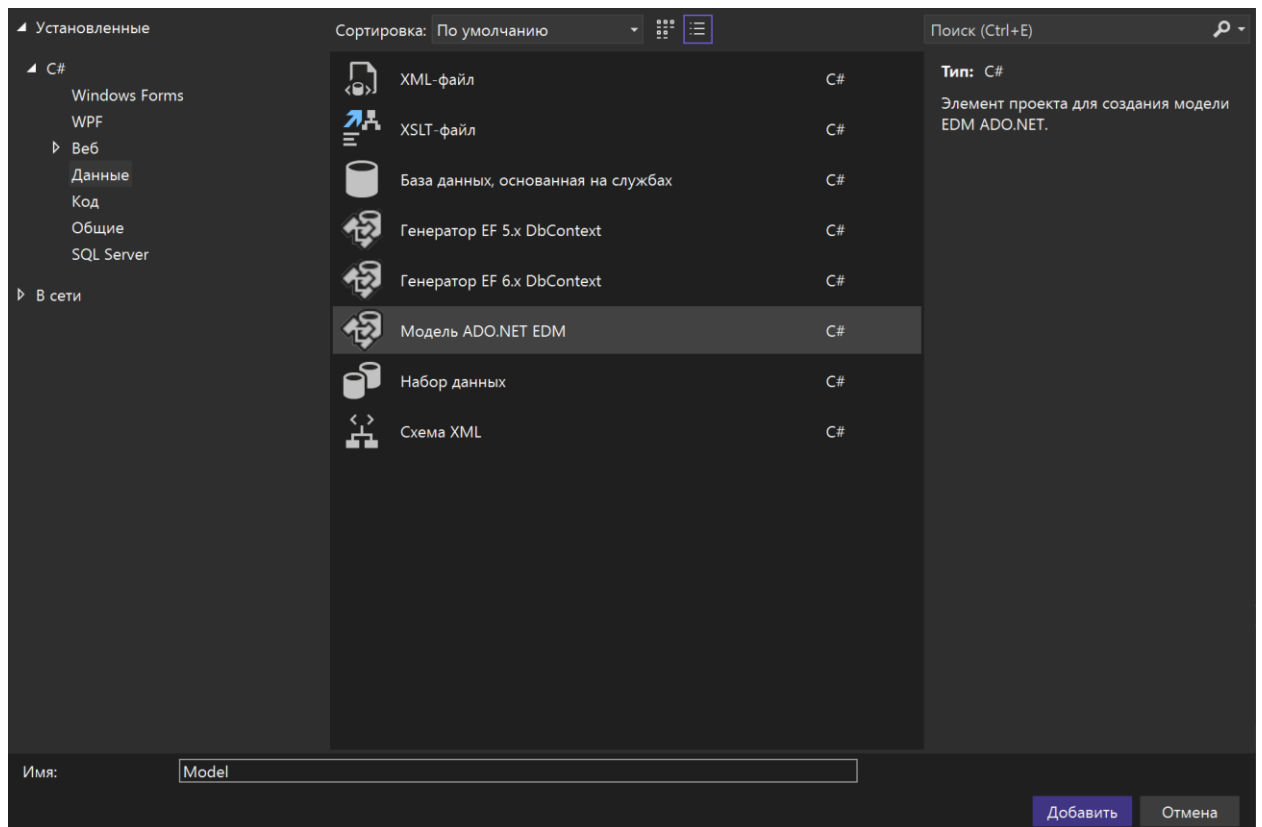


Рисунок 16 – Подключение базы данных к проекту

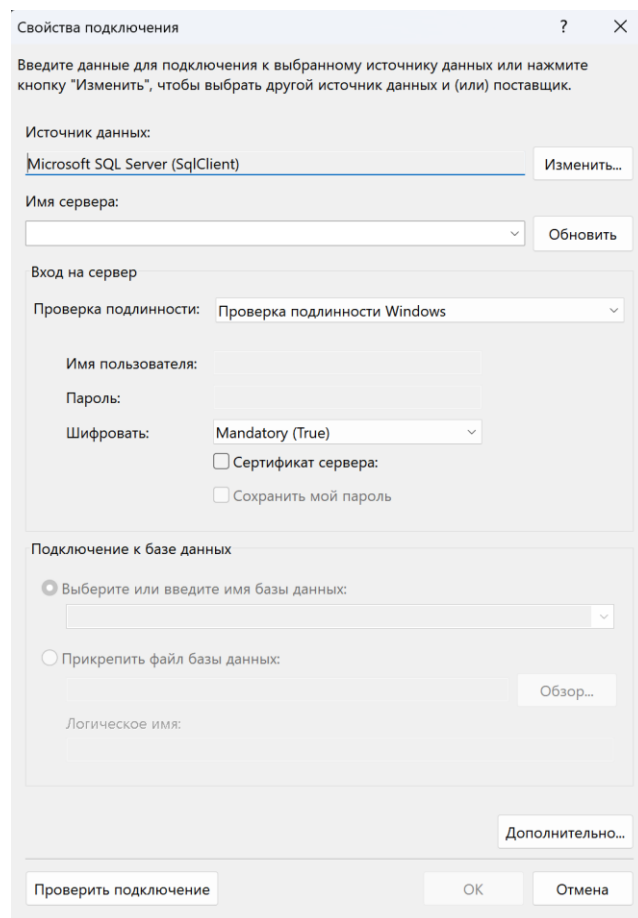


Рисунок 17 – Создание соединения с базой данных

Для удобства работы с данными в автоматически созданном классе реализуем метод доступа. Код этого метода показан в Листинге 2.

Листинг 2 – Фрагмент кода с методом для получения доступа к данным

```
private static Entities _context;  
public static Entities GetContext()  
{  
    if (_context == null) _context = new Entities();  
    return _context;  
}
```

На рисунках 18 - 23 представлены страницы созданного приложения.

Частные объявления

Назад

Доска объявлений

Логин

Пароль

Войти

Войти как гость

Рисунок 18 – Страница авторизации

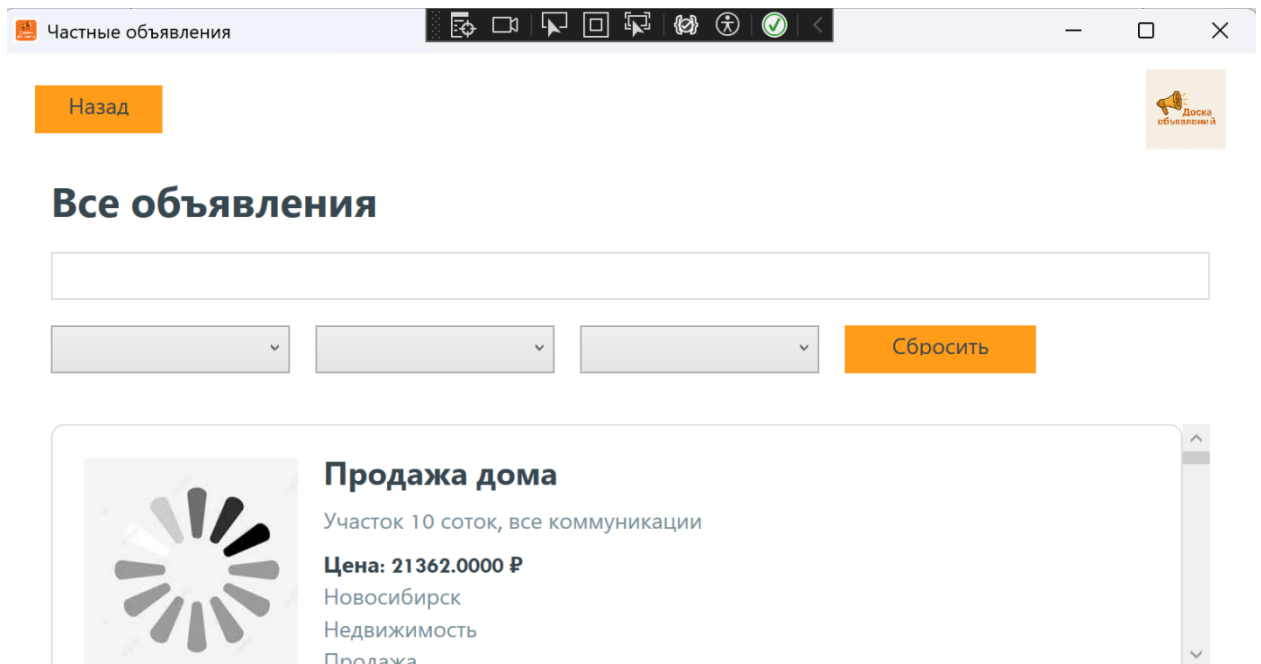


Рисунок 19 – Страница всех объявлений гостя

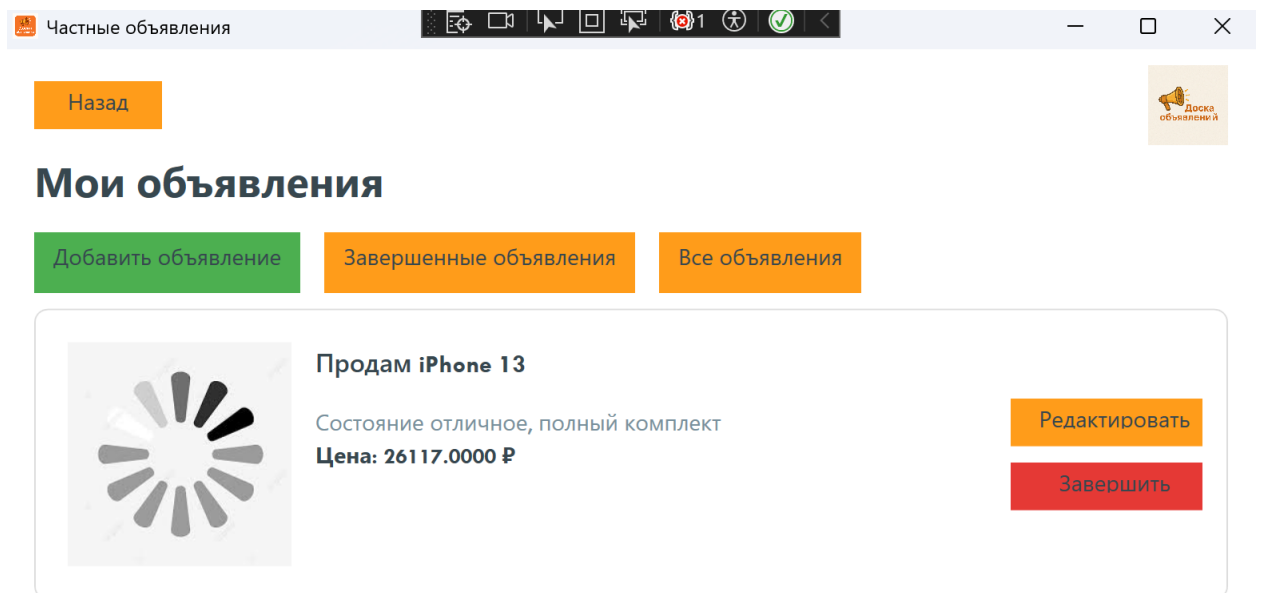


Рисунок 20 – Страница объявлений пользователя

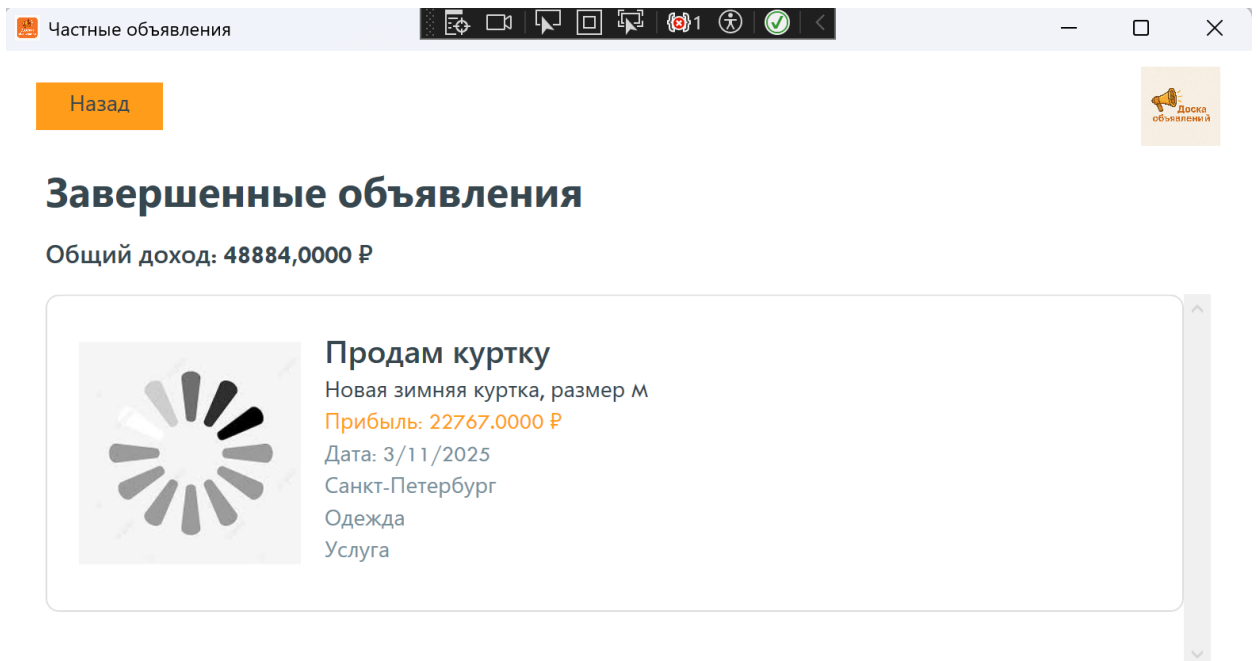


Рисунок 21 – Страница завершенных объявлений

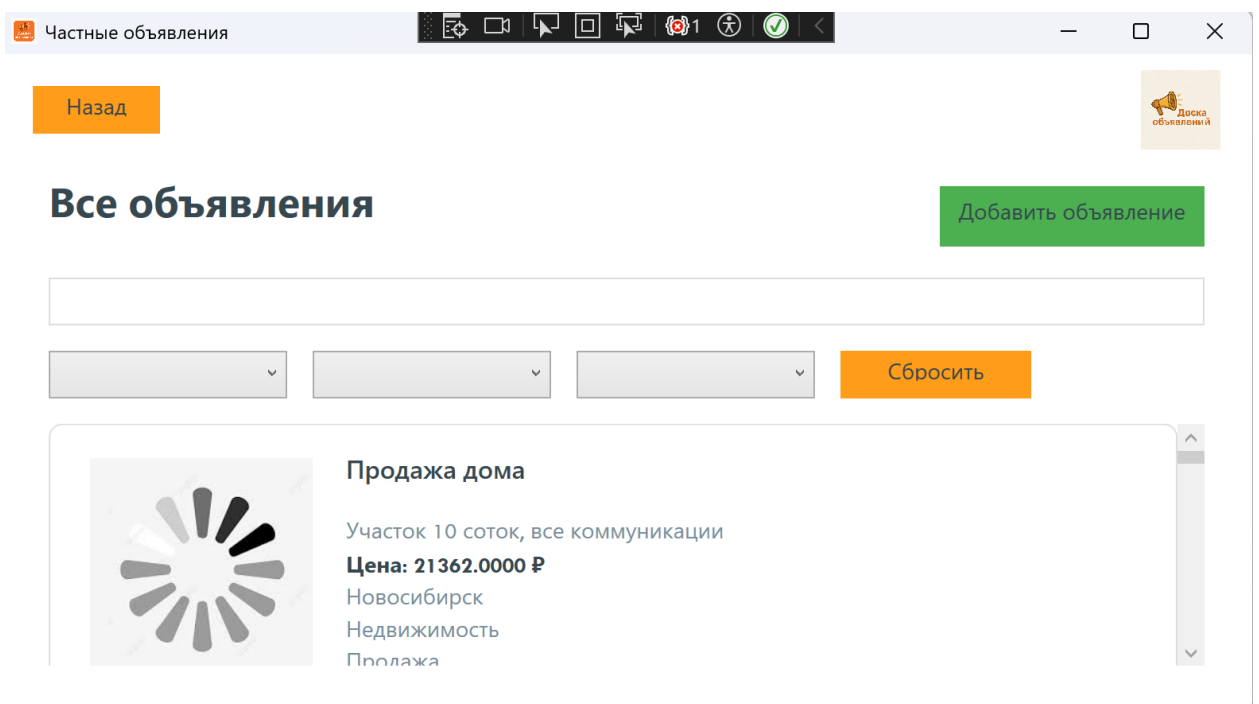


Рисунок 22 – Страница всех объявлений авторизованного пользователя

Частные объявления

Назад

Доска объявлений

Название

Описание

Цена

Город Категория Тип Статус объявления

Выбрать фото

Сохранить

Рисунок 23 – Страница добавления/редактирования объявлений

На листингах 3 - 17 представлены коды страниц.

Листинг 3 – Окно MainWindow.xaml

```
<Window x:Class="Medvedeva_PrivateAdertisementService.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:Medvedeva_PrivateAdertisementService"
mc:Ignorable="d"
Title="Частные объявления" Height="450" Width="800"
WindowStartupLocation="CenterScreen" Closing="Window_Closing">
<Grid Margin="10">

    <Grid.RowDefinitions>
        <RowDefinition Height="50"/>
        <RowDefinition Height="*" />
    </Grid.RowDefinitions>

    <Grid Grid.Row="0">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="100"/>
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>
```

```

        <Button x:Name="BackButton"
            Content="Назад"
            Width="80"
            Height="30"
            HorizontalContentAlignment="Center"
            Click="BackButton_Click"/>

        <StackPanel Grid.Column="1" Orientation="Horizontal"
            VerticalAlignment="Center" HorizontalAlignment="Right"
            Margin="0,0,10,0">
            <Image Source="\Resources\ЛТор.png"/>
        </StackPanel>
    </Grid>

    <Frame Grid.Row="1"
        x:Name="MainFrame"
        NavigationUIVisibility="Hidden"/>
</Grid>
</Window>

```

Листинг 4 – Реализация функций MainWindow.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Medvedeva_PrivateAdvertisementService
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public string currentUserRole = null;

        public MainWindow()
        {
            InitializeComponent();
            Loaded += (s, e) =>
            {
                MainFrame.Visibility = Visibility.Visible;
                MainFrame.Navigate(new AuthPage());
            };
        }

        private void BackButton_Click(object sender, RoutedEventArgs e)
        {
            if (MainFrame.CanGoBack)
            {
                MainFrame.GoBack();
            }
        }

        private void Window_Closing(object sender, System.ComponentModel.CancelEventArgs e)
        {

```

```

var result = MessageBox.Show("Вы уверены, что хотите закрыть окно?", "Подтверждение",
    MessageBoxButton.YesNo, MessageBoxImage.Question);
if (result == MessageBoxResult.No)
{
    e.Cancel = true;
}
}
}
}

```

Листинг 5 – Реализация функций UserAdsPage.xaml

```

<Page x:Class="Medvedeva_PrivateAdvertisementService.UserAdsPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:Medvedeva_PrivateAdvertisementService"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="Мои объявления">

    <Grid>
        <ScrollViewer VerticalScrollBarVisibility="Auto">
            <StackPanel Orientation="Vertical" Margin="10">

                <TextBlock Text="Мои объявления"
                    Style="{StaticResource PageTitle}"/>

                <StackPanel Orientation="Horizontal" HorizontalAlignment="Left">
                    <Button Content="Добавить объявление"
                        Style="{StaticResource SuccessButton}"
                        Click="Add_Click"
                        Height="38"
                        Margin="0,0,15,10"/>
                    <Button Content="Завершенные объявления"
                        Click="ShowCompletedAds_Click"
                        Margin="0,0,15,10"/>
                    <Button Content="Все объявления"
                        Click="AdsPage_Click"
                        Margin="0,0,0,10"/>
                </StackPanel>

                <ItemsControl x:Name="UserAdsList">
                    <ItemsControl.ItemTemplate>
                        <DataTemplate>
                            <Border Style="{StaticResource AdCard}">

                                <Grid Margin="5">
                                    <Grid.ColumnDefinitions>
                                        <ColumnDefinition Width="150"/>
                                        <ColumnDefinition Width="*"/>
                                        <ColumnDefinition Width="120"/>
                                    </Grid.ColumnDefinitions>

                                    <Image Source="{Binding ImagePath}"
                                        Width="140" Height="140"
                                        Stretch="UniformToFill"
                                        Margin="5"/>

                                    <StackPanel Grid.Column="1" Margin="10">
                                        <TextBlock Text="{Binding Title}"
                                            Style="{StaticResource SubTitle}"/>
                                        <TextBlock Text="{Binding Description}"
                                            TextWrapping="Wrap"
                                            Margin="0 5 0 5"
                                            Foreground="{StaticResource SecondaryText}"/>
                                    </StackPanel>
                                </Grid>
                            </Border>
                        </DataTemplate>
                    </ItemsControl.ItemTemplate>
                </ItemsControl>
            </StackPanel>
        </ScrollViewer>
    </Grid>

```



```

        <TextBlock Text="{Binding Price, StringFormat='Цена: {0} P'}"
            FontWeight="Bold"
            Foreground="{StaticResource PrimaryText}"/>
        <TextBlock Text="{Binding Status, StringFormat='Статус: {0}'}"
            Margin="0 5 0 0"
            Foreground="{StaticResource SecondaryText}"/>
    </StackPanel>

    <StackPanel Grid.Column="2"
        VerticalAlignment="Center">
        <Button Content="Редактировать"
            Margin="0 5"
            Width="130"
            Height="30"
            Tag="{Binding AdID}"
            Click="Edit_Click"/>
        <Button Content="Завершить"
            Width="130"
            Height="30"
            Margin="0 5"
            Style="{StaticResource DangerButton}"
            Tag="{Binding AdID}"
            Click="Finish_Click"/>
    </StackPanel>

</Grid>
</Border>
</DataTemplate>
</ItemsControl.ItemTemplate>
</ItemsControl>

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

```

Листинг 6 – Реализация функций UserAdsPage.xaml.cs

```

using Medvedeva_PrivateAdvertisementService.Pages;
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Medvedeva_PrivateAdvertisementService
{
    /// <summary>
    /// Логика взаимодействия для UserAdsPage.xaml
    /// </summary>
    public partial class UserAdsPage : Page
    {
        private int currentUserId;

        public UserAdsPage(int userId)
        {

```

```

InitializeComponent();
currentUserId = userId;
LoadUserAds();
}

public void LoadUserAds()
{
    try
    {
        using (var db = new Entities())
        {
            var ads = db.Ads
                .Where(a => a.UserID == currentUserId && a.StatusID == 1)
                .AsNoTracking()
                .Select(a => new
                {
                    a.AdID,
                    a.Title,
                    a.Description,
                    a.Price,

                    ImagePath = string.IsNullOrEmpty(a.ImagePath)
                        ? "/Resources/images.jpg"
                        : a.ImagePath
                })
                .ToList();

            UserAdsList.ItemsSource = ads;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка загрузки объявлений: {ex.Message}");
    }
}

private void Add_Click(object sender, RoutedEventArgs e)
{
    (Application.Current.MainWindow as MainWindow)
        ?.MainFrame.Navigate(new AddEditPage(currentUserId));
}

private void Edit_Click(object sender, RoutedEventArgs e)
{
    int adId = Convert.ToInt32((sender as Button).Tag);
    (Application.Current.MainWindow as MainWindow)
        ?.MainFrame.Navigate(new AddEditPage(currentUserId, adId));
}

private void ShowCompletedAds_Click(object sender, RoutedEventArgs e)
{
    (Application.Current.MainWindow as MainWindow)
        ?.MainFrame.Navigate(new CompletedAdsPage(currentUserId));
}

private void Finish_Click(object sender, RoutedEventArgs e)
{
    int adId = Convert.ToInt32((sender as Button).Tag);

    try
    {
        using (var db = new Entities())
        {

```

```

var ad = db.Ads.FirstOrDefault(a => a.AdID == adId);
var user = db.Users.FirstOrDefault(u => u.UserID == currentUserId);

if (ad == null || user == null)
{
    MessageBox.Show("Ошибка: объявление или пользователь не найден!");
    return;
}

ad.StatusID = 2; // Завершено

user.TotalProfit += ad.Price;

db.SaveChanges();
}

MessageBox.Show("Объявление завершено!");
LoadUserAds();
}
catch (Exception ex)
{
    MessageBox.Show("Ошибка: " + ex.Message);
}
}

private void AdsPage_Click(object sender, RoutedEventArgs e)
{
    (Application.Current.MainWindow as MainWindow)
        ?.MainFrame.Navigate(new AdsPage(currentUserId));
}
}
}

```

Листинг 7 – Реализация функций GuestViewPage.xaml

```

<Page x:Class="Medvedeva_PrivateAdvertisementService.GuestViewPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Просмотр объявлений">

    <Grid Margin="20">

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

        <StackPanel Grid.Row="0">

            <TextBlock Text="Все объявления"
                Style="{StaticResource PageTitle}"/>

            <TextBox x:Name="SearchBox"
                Height="30"
                Margin="0 0 0 16"
                ToolTip="Поиск по названию или описанию"
                TextChanged="FilterChanged"/>

            <StackPanel Orientation="Horizontal" Margin="0 0 0 16">

                <ComboBox x:Name="CityFilter"
                    Width="150"
                    Margin="0 0 16 0"
                    Height="30"
                    ToolTip="Город"
                    SelectionChanged="FilterChanged"/>
            </StackPanel>
        </StackPanel>
    </Grid>

```

```

<ComboBox x:Name="CategoryFilter"
    Width="150"
    Margin="0 0 16 0"
    Height="30"
    ToolTip="Категория"
    SelectionChanged="FilterChanged"/>

<ComboBox x:Name="TypeFilter"
    Width="150"
    Height="30"
    Margin="0 0 16 0"
    ToolTip="Тип"
    SelectionChanged="FilterChanged"/>

<Button Content="Сбросить"
    Width="120"
    Height="30"
    Click="ResetFilters_Click"/>

</StackPanel>

</StackPanel>

<ScrollView Grid.Row="1" Margin="0 16 0 0">
    <ItemsControl x:Name="AdsList">
        <ItemsControl.ItemTemplate>
            <DataTemplate>
                <Border Style="{StaticResource AdCard}">

                    <Grid Margin="10">
                        <Grid.ColumnDefinitions>
                            <ColumnDefinition Width="150"/>
                            <ColumnDefinition Width="*/>
                        </Grid.ColumnDefinitions>

                        <Image Source="{Binding ImagePath}"
                            Width="140"
                            Height="140"
                            Stretch="UniformToFill"
                            Margin="0 0 16 0"/>

                        <StackPanel Grid.Column="1">

                            <TextBlock Text="{Binding Title}"
                                FontSize="20"
                                FontWeight="Bold"
                                Foreground="{StaticResource PrimaryText}"/>

                            <TextBlock Text="{Binding Description}"
                                Margin="0 6 0 6"
                                TextWrapping="Wrap"
                                Foreground="{StaticResource SecondaryText}"/>

                            <StackPanel Orientation="Horizontal" Margin="0 6 0 0">
                                <TextBlock Text="{Binding Price, StringFormat='Цена: {0} P'}"
                                    FontWeight="Bold"
                                    Foreground="{StaticResource PrimaryText}"/>
                            </StackPanel>

                            <TextBlock Text="{Binding CityName}" Foreground="{StaticResource SecondaryText}"/>
                            <TextBlock Text="{Binding CategoryName}" Foreground="{StaticResource
SecondaryText}"/>
                            <TextBlock Text="{Binding AdTypeName}" Foreground="{StaticResource SecondaryText}"/>

```

```

        </StackPanel>
    </Grid>
</Border>
</DataTemplate>
</ItemsControl.ItemTemplate>
</ItemsControl>
</ScrollViewer>

</Grid>
</Page>

```

Листинг 8 – Реализация функций GuestViewPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows;
using System.Data.Entity;
using System.Windows.Controls;

namespace Medvedeva_PrivateAdvertisementService
{
    /// <summary>
    /// Логика взаимодействия для GuestViewPage.xaml
    /// </summary>
    public partial class GuestViewPage : Page
    {
        public GuestViewPage()
        {
            InitializeComponent();
            LoadAds();
        }

        private void LoadAds()
        {
            try
            {
                using (var db = new Entities())
                {
                    var ads = db.Ads
                        .Where(a => a.StatusID == 1)
                        .Select(a => new
                        {
                            a.AdID,
                            a.Title,
                            a.Description,
                            a.Price,

                            CityID = a.CityID,
                            CategoryID = a.CategoryID,
                            TypeID = a.AdTypeID,

                            CityName = a.Cities.CityName,
                            CategoryName = a.Categories.CategoryName,
                            AdTypeName = a.AdTypes.TypeDescription,

                            ImagePath = string.IsNullOrEmpty(a.ImagePath)
                                ? "/Resources/images.jpg"
                                : a.ImagePath
                        })
                        .ToList();

                    CityFilter.ItemsSource = db.Cities.ToList();
                    CityFilter.DisplayMemberPath = "CityName";
                    CityFilter.SelectedValuePath = "CityID";

                    CategoryFilter.ItemsSource = db.Categories.ToList();

```

```

        CategoryFilter.DisplayMemberPath = "CategoryName";
        CategoryFilter.SelectedValuePath = "CategoryID";

        TypeFilter.ItemsSource = db.AdTypes.ToList();
        TypeFilter.DisplayMemberPath = "TypeDescription";
        TypeFilter.SelectedValuePath = "AdTypeID";

        AdsList.ItemsSource = ads;
    }
}
catch
{
    MessageBox.Show(
        "Ошибка загрузки данных",
        "Ошибка",
        MessageBoxButton.OK,
        MessageBoxImage.Error);
}

private void ResetFilters_Click(object sender, RoutedEventArgs e)
{
    SearchBox.Text = "";
    CityFilter.SelectedIndex = -1;
    CategoryFilter.SelectedIndex = -1;
    TypeFilter.SelectedIndex = -1;
}

private void FilterChanged(object sender, EventArgs e)
{
    UpdateAds();
}

private void UpdateAds()
{
    if (!IsInitialized) return;

    try
    {
        using (var db = new Entities())
        {
            var ads = db.Ads
                .Where(a => a.StatusID == 1)
                .Select(a => new
                {
                    a.AdID,
                    a.Title,
                    a.Description,
                    a.Price,

                    CityID = a.CityID,
                    CategoryID = a.CategoryID,
                    TypeID = a.AdTypeID,

                    CityName = a.Cities.CityName,
                    CategoryName = a.Categories.CategoryName,
                    AdTypeName = a.AdTypes.TypeDescription,

                    ImagePath = string.IsNullOrEmpty(a.ImagePath)
                        ? "/Resources/images.jpg"
                        : a.ImagePath
                })
                .ToList();

            if (!string.IsNullOrEmpty(SearchBox.Text))
            {

```

```

        string search = SearchBox.Text.ToLower();

        ads = ads.Where(a =>
            (a.Title != null && a.Title.ToLower().Contains(search)) ||
            (a.Description != null && a.Description.ToLower().Contains(search))
        ).ToList();
    }

    if (CityFilter.SelectedItem != null)
    {
        int selectedCity = (int)CityFilter.SelectedValue;
        ads = ads.Where(a => a.CityID == selectedCity).ToList();
    }

    if (CategoryFilter.SelectedItem != null)
    {
        int selectedCategory = (int)CategoryFilter.SelectedValue;
        ads = ads.Where(a => a.CategoryID == selectedCategory).ToList();
    }

    if (TypeFilter.SelectedItem != null)
    {
        int selectedType = (int)TypeFilter.SelectedValue;
        ads = ads.Where(a => a.TypeID == selectedType).ToList();
    }

    AdsList.ItemsSource = ads;
}
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при фильтрации: {ex.Message}");
}
}
}
}

```

Листинг 9 – Реализация функций CompletedAdsPage.xaml

```

<Page x:Class="Medvedeva_PrivateAdvertisementService.Pages.CompletedAdsPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:Medvedeva_PrivateAdvertisementService.Pages"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="Завершенные объявления">

    <Grid Margin="16">

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

        <TextBlock Text="Завершенные объявления"
            Style="{StaticResource PageTitle}"/>

        <TextBlock x:Name="TotalProfitText"
            Style="{StaticResource SubTitle}"
            Grid.Row="1"/>

        <ScrollView Grid.Row="2">
            <ItemsControl x:Name="CompletedAdsList">

```

```

<ItemsControl.ItemTemplate>
  <DataTemplate>
    <Border Style="{StaticResource AdCard}">
      <Grid Margin="5">
        <Grid.ColumnDefinitions>
          <ColumnDefinition Width="150"/>
          <ColumnDefinition Width="*/>
        </Grid.ColumnDefinitions>

        <Image Source="{Binding ImagePath}"
          Width="140" Height="140"
          Stretch="UniformToFill"/>

        <StackPanel Grid.Column="1" Margin="10">
          <TextBlock Text="{Binding Title}"
            FontSize="20"
            FontWeight="SemiBold"/>

          <TextBlock Text="{Binding Description}" />

          <TextBlock Text="{Binding Price,
            StringFormat='Прибыль: {0} P'}"
            Foreground="{StaticResource AccentBrush}"/>

          <TextBlock Text="{Binding PostDate,
            StringFormat='Дата: {0:d}'}"
            Foreground="{StaticResource SecondaryText}"/>

          <TextBlock Text="{Binding CityName}"
            Foreground="{StaticResource SecondaryText}"/>
          <TextBlock Text="{Binding CategoryName}"
            Foreground="{StaticResource SecondaryText}"/>
          <TextBlock Text="{Binding AdTypeName}"
            Foreground="{StaticResource SecondaryText}"/>
        </StackPanel>
      </Grid>
    </Border>
  </DataTemplate>
</ItemsControl.ItemTemplate>
</ItemsControl>
</ScrollView>

</Grid>
</Page>

```

Листинг 10 – Реализация функций CompletedAdsPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Medvedeva_PrivateAdvertisementService.Pages
{
    /// <summary>
    /// Логика взаимодействия для CompletedAdsPage.xaml

```



```

/// </summary>
public partial class CompletedAdsPage : Page
{
    private int currentUserId;

    public CompletedAdsPage(int userId)
    {
        InitializeComponent();
        currentUserId = userId;
        LoadCompletedAds();
        LoadTotalProfit();
    }

    public void LoadCompletedAds()
    {
        try
        {
            using (var db = new Entities())
            {
                var ads = db.Ads
                    .Where(a => a.UserID == currentUserId && a.StatusID == 2) // Статус "Завершено"
                    .Select(a => new
                    {
                        AdID,
                        Title,
                        Description,
                        Price = a.Price,
                        PostDate = a.PostDate,
                        CityName = a.Cities.CityName,
                        CategoryName = a.Categories.CategoryName,
                        AdTypeName = a.AdTypes.TypeDescription,
                        ImagePath = string.IsNullOrEmpty(a.ImagePath)
                            ? "/Resources/images.jpg"
                            : a.ImagePath
                    })
                    .ToList();

                CompletedAdsList.ItemsSource = ads;
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Ошибка загрузки завершенных объявлений: {ex.Message}",
                "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }

    private void LoadTotalProfit()
    {
        try
        {
            using (var db = new Entities())
            {
                var user = db.Users.FirstOrDefault(u => u.UserID == currentUserId);
                if (user != null)
                {
                    TotalProfitText.Text = $"Общий доход: {user.TotalProfit} Р";
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Ошибка загрузки дохода: {ex.Message}",
                "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }
}

```

```
}
```

Листинг 11 – Реализация функций AuthPage.xaml

```
<Page x:Class="Medvedeva_PrivateAdvertisementService.AuthPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:Medvedeva_PrivateAdvertisementService"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="Авторизация">
    <Grid Margin="50">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>
        <StackPanel HorizontalAlignment="Center" VerticalAlignment="Top" Width="500" >
            <TextBlock Text="Логин"
                Margin="0 0 0 5"/>
            <TextBox x:Name="LoginBox"
                Height="30"
                Margin="0 0 0 5"/>

            <TextBlock Text="Пароль"
                Margin="0 0 0 5"/>
            <PasswordBox x:Name="PasswordBox"
                Height="30"
                Margin="0 0 0 15"/>
            <StackPanel HorizontalAlignment="Center" >

                <Button Grid.Row="2"
                    Content="Войти"
                    Width="200"
                    Height="30"
                    Margin="0 0 0 15"
                    Click="Login_Click"/>

                <Button Grid.Row="3"
                    Content="Войти как гость"
                    Width="200"
                    Height="30"
                    Click="Guest_Click"/>
            </StackPanel>
        </StackPanel>
    </Grid>
</Page>
```

Листинг 12 – Реализация функций AuthPage.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
```

```

using System.Windows.Shapes;

namespace Medvedeva_PrivateAdvertisementService
{
    /// <summary>
    /// Логика взаимодействия для AuthPage.xaml
    /// </summary>
    public partial class AuthPage : Page
    {
        public AuthPage()
        {
            InitializeComponent();
        }
        private Users currentUser;
        private void Login_Click(object sender, RoutedEventArgs e)
        {
            if (string.IsNullOrEmpty(LoginBox.Text) || string.IsNullOrEmpty>PasswordBox.Password))
            {
                MessageBox.Show("Введите логин и пароль");
                return;
            }
            try
            {
                using (var db = new Entities())
                {
                    var user = db.Users
                        .AsNoTracking()
                        .FirstOrDefault(u => u.Login == LoginBox.Text && u.PasswordHash == PasswordBox.Password);

                    if (user == null)
                    {
                        MessageBox.Show("Пользователь с такими данными не найден!");
                    }
                    else
                    {
                        MessageBox.Show("Пользователь успешно найден!");
                        (Application.Current.MainWindow as MainWindow)?.MainFrame.Navigate(new
UserAdsPage(user.UserID));
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Ошибка доступа к базе данных: {ex.Message}");
            }
        }

        private void Guest_Click(object sender, RoutedEventArgs e)
        {
            (Application.Current.MainWindow as MainWindow)?.MainFrame.Navigate(new GuestViewPage());
        }
    }
}

```

Листинг 13 – Реализация функций AdsPage.xaml

```

<Page x:Class="Medvedeva_PrivateAdvertisementService.AdsPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:Medvedeva_PrivateAdvertisementService"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="Все объявления">

```

```
<Grid Margin="20">
<Grid.RowDefinitions>
  <RowDefinition Height="Auto"/>
  <RowDefinition Height="*/>
</Grid.RowDefinitions>
<StackPanel Grid.Row="0">
  <DockPanel Margin="0 0 0 16">
    <TextBlock Text="Все объявления"
      Style="{StaticResource PageTitle}"
      DockPanel.Dock="Left"/>
    <Button Content="Добавить объявление"
      Style="{StaticResource SuccessButton}"
      Height="38"
      HorizontalAlignment="Right"
      Click="AddAd_Click"
      DockPanel.Dock="Right"/>
  </DockPanel>
  <TextBox x:Name="SearchBox"
    Height="30"
    ToolTip="Поиск по названию или описанию..."
    Margin="0 0 0 16"
    TextChanged="FilterChanged"/>
  <StackPanel Orientation="Horizontal" Margin="0 0 0 16">
    <ComboBox x:Name="CityFilter"
      SelectionChanged="FilterChanged"
      ToolTip="Город"/>
    <ComboBox x:Name="CategoryFilter"
      SelectionChanged="FilterChanged"
      ToolTip="Категория"/>
    <ComboBox x:Name="TypeFilter"
      SelectionChanged="FilterChanged"
      ToolTip="Тип"/>
    <Button Content="Сбросить"
      Width="120"
      Height="30"
      Click="ResetFilters_Click"/>
  </StackPanel>
</StackPanel>
<ScrollView Grid.Row="1">
  <ItemsControl x:Name="AdsList">
    <ItemsControl.ItemTemplate>
      <DataTemplate>
        <Border Style="{StaticResource AdCard}">
          <Grid Margin="10">
            <Grid.ColumnDefinitions>
              <ColumnDefinition Width="150"/>
              <ColumnDefinition Width="*/>
              <ColumnDefinition Width="120"/>
            </Grid.ColumnDefinitions>

```

```

        <Image Source="{Binding ImagePath}"
            Width="140"
            Height="140"
            Stretch="UniformToFill"/>

        <StackPanel Grid.Column="1" Margin="16 0 0 0">

            <TextBlock Text="{Binding Title}"
                Style="{StaticResource SubTitle}"/>

            <TextBlock Text="{Binding Description}"
                Margin="0 6 0 6"
                Foreground="{StaticResource SecondaryText}"/>

            <TextBlock Text="{Binding Price, StringFormat='Цена: {0} P'}"
                FontWeight="Bold"
                Foreground="{StaticResource PrimaryText}"/>

            <TextBlock Text="{Binding CityName}"
                Foreground="{StaticResource SecondaryText}"/>
            <TextBlock Text="{Binding CategoryName}"
                Foreground="{StaticResource SecondaryText}"/>
            <TextBlock Text="{Binding AdTypeName}"
                Foreground="{StaticResource SecondaryText}"/>

        </StackPanel>

        <StackPanel Grid.Column="2" VerticalAlignment="Top">

            <Button Content="Редактировать"
                Tag="{Binding AdID}"
                Margin="0 0 0 8"
                Click="EditAd_Click"
                Visibility="{Binding IsOwner, Converter={StaticResource BoolToVis}}"/>

            <Button Content="Удалить"
                Style="{StaticResource DangerButton}"
                Padding="5"
                FontSize="12"
                Tag="{Binding AdID}"
                Click="DeleteAd_Click"
                Visibility="{Binding IsOwner, Converter={StaticResource BoolToVis}}"/>

        </StackPanel>
    </Grid>

</Border>
</DataTemplate>
</ItemsControl.ItemTemplate>
</ItemsControl>
</ScrollView>

</Grid>
</Page>

```

Листинг 14 – Реализация функций AdsPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;

```

```

using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Medvedeva_PrivateAdvertisementService
{
    /// <summary>
    /// Логика взаимодействия для AdsPage.xaml
    /// </summary>
    public partial class AdsPage : Page
    {
        private int currentUserId;

        public AdsPage(int userId)
        {
            InitializeComponent();
            currentUserId = userId;
            LoadAds();
        }

        private void LoadAds()
        {
            try
            {
                using (var db = new Entities())
                {
                    var ads = db.Ads
                        .Select(a => new
                        {
                            a.AdID,
                            a.Title,
                            a.Description,
                            a.Price,
                            a.StatusID,

                            CityName = a.Cities.CityName,
                            CategoryName = a.Categories.CategoryName,
                            AdTypeName = a.AdTypes.TypeDescription,

                            IsOwner = (a.UserID == currentUserId),

                            ImagePath = string.IsNullOrEmpty(a.ImagePath)
                                ? "/Resources/images.jpg"
                                : a.ImagePath
                        })
                        .ToList();
                    CityFilter.ItemsSource = db.Cities.ToList();
                    CityFilter.DisplayMemberPath = "CityName";
                    CityFilter.SelectedValuePath = "CityID";

                    CategoryFilter.ItemsSource = db.Categories.ToList();
                    CategoryFilter.DisplayMemberPath = "CategoryName";
                    CategoryFilter.SelectedValuePath = "CategoryID";

                    TypeFilter.ItemsSource = db.AdTypes.ToList();
                    TypeFilter.DisplayMemberPath = "TypeDescription";
                    TypeFilter.SelectedValuePath = "AdTypeID";

                    AdsList.ItemsSource = ads;
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Ошибка загрузки объявлений:\n" + ex.Message,
                    "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
            }
        }
    }
}

```

```

    }

    private void AddAd_Click(object sender, RoutedEventArgs e)
    {
        (Application.Current.MainWindow as MainWindow)
            ?.MainFrame.Navigate(new AddEditPage(currentUserId));
    }

    private void EditAd_Click(object sender, RoutedEventArgs e)
    {
        int adId = Convert.ToInt32((sender as Button).Tag);

        (Application.Current.MainWindow as MainWindow)
            ?.MainFrame.Navigate(new AddEditPage(currentUserId, adId));
    }

    private void DeleteAd_Click(object sender, RoutedEventArgs e)
    {
        int adId = Convert.ToInt32((sender as Button).Tag);

        if (MessageBox.Show("Удалить объявление?",
            "Подтверждение", MessageBoxButton.YesNo, MessageBoxImage.Warning)
            != MessageBoxResult.Yes) return;

        try
        {
            using (var db = new Entities())
            {
                var ad = db.Ads.FirstOrDefault(a => a.AdID == adId);
                if (ad == null) return;

                db.Ads.Remove(ad);
                db.SaveChanges();
            }

            MessageBox.Show("Объявление удалено.",
                "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

            LoadAds();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Ошибка удаления:\n" + ex.Message,
                "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }

    private void ResetFilters_Click(object sender, RoutedEventArgs e)
    {
        SearchBox.Text = "";
        CityFilter.SelectedIndex = -1;
        CategoryFilter.SelectedIndex = -1;
        TypeFilter.SelectedIndex = -1;
    }

    private void FilterChanged(object sender, EventArgs e)
    {
        UpdateAds();
    }

    private void UpdateAds()
    {
        if (!IsInitialized) return;

        try
        {
            using (var db = new Entities())

```

```

{
    var ads = db.Ads
        .Where(a => a.StatusID == 1)
        .Select(a => new
        {
            a.AdID,
            a.Title,
            a.Description,
            a.Price,

            CityID = a.CityID,
            CategoryID = a.CategoryID,
            TypeID = a.AdTypeID,

            CityName = a.Cities.CityName,
            CategoryName = a.Categories.CategoryName,
            AdTypeName = a.AdTypes.TypeDescription,
            IsOwner = (a.UserID == currentUserID),

            ImagePath = string.IsNullOrEmpty(a.ImagePath)
                ? "/Resources/images.jpg"
                : a.ImagePath
        })
        .ToList();

    if (!string.IsNullOrEmpty(SearchBox.Text))
    {
        string search = SearchBox.Text.ToLower();

        ads = ads.Where(a =>
            (a.Title != null && a.Title.ToLower().Contains(search)) ||
            (a.Description != null && a.Description.ToLower().Contains(search))
        ).ToList();
    }

    if (CityFilter.SelectedItem != null)
    {
        int selectedCity = (int)CityFilter.SelectedValue;
        ads = ads.Where(a => a.CityID == selectedCity).ToList();
    }

    if (CategoryFilter.SelectedItem != null)
    {
        int selectedCategory = (int)CategoryFilter.SelectedValue;
        ads = ads.Where(a => a.CategoryID == selectedCategory).ToList();
    }

    if (TypeFilter.SelectedItem != null)
    {
        int selectedType = (int)TypeFilter.SelectedValue;
        ads = ads.Where(a => a.TypeID == selectedType).ToList();
    }

    AdsList.ItemsSource = ads;
}
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при фильтрации: {ex.Message}");
}
}
}

```

Листинг 15 – Реализация функций AddEditPage.xaml


```
<Page x:Class="Medvedeva_PrivateAdvertisementService.AddEditPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:local="clr-namespace:Medvedeva_PrivateAdvertisementService"
mc:Ignorable="d"
d:DesignHeight="450" d:DesignWidth="800"
Title="Добавление / редактирование объявления">

<ScrollViewer VerticalScrollBarVisibility="Auto">

    <StackPanel Margin="15">

        <TextBlock Text="Название"/>
        <TextBox x:Name="TitleBox"
            Height="30"
            Margin="0 8 0 16"/>

        <TextBlock Text="Описание"/>
        <TextBox x:Name="DescriptionBox"
            Height="80"
            AcceptsReturn="True"
            TextWrapping="Wrap"
            Margin="0 8 0 16"/>

        <TextBlock Text="Цена"/>
        <TextBox x:Name="PriceBox"
            Height="30"
            Margin="0 8 0 16"/>

        <StackPanel Orientation="Horizontal" Margin="0 0 0 16">
            <StackPanel Width="170">
                <TextBlock Text="Город"/>
                <ComboBox x:Name="CityBox" Height="30"/>
            </StackPanel>

            <StackPanel Width="170" Margin="16 0 0 0">
                <TextBlock Text="Категория"/>
                <ComboBox x:Name="CategoryBox" Height="30"/>
            </StackPanel>

            <StackPanel Width="170" Margin="16 0 0 0">
                <TextBlock Text="Тип"/>
                <ComboBox x:Name="TypeBox" Height="30"/>
            </StackPanel>

            <StackPanel Width="170" Margin="16 0 0 0">
                <TextBlock Text="Статус объявления"/>
                <ComboBox x:Name="StatusBox" Height="30"/>
            </StackPanel>
        </StackPanel>

        <Image x:Name="AdImage"
            Width="200"
            Height="150"
            Stretch="UniformToFill"/>

        <Button Content="Выбрать фото"
            Width="150"
            Click="ChooseImage_Click"
            Margin="0 0 0 15"/>

        <Button x:Name="SaveButton"
            Content="Сохранить"/>
```

```

        Margin="0 20 0 0"
        Click="Save_Click"/>

</StackPanel>

</ScrollViewer>
</Page>

```

Листинг 16 – Реализация функций AddEditPage.xaml.cs

```

using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace Medvedeva_PrivateAdvertisementService
{
    /// <summary>
    /// Логика взаимодействия для AddEditPage.xaml
    /// </summary>
    public partial class AddEditPage : Page
    {
        private int userId;
        private int? adId;
        private string selectedImagePath = null;

        public AddEditPage(int userId, int? adId = null)
        {
            InitializeComponent();
            this.userId = userId;
            this.adId = adId;

            LoadComboboxes();

            if (adId != null)
                LoadAdData();
        }

        private void LoadComboboxes()
        {
            using (var db = new Entities())
            {
                CityBox.ItemsSource = db.Cities.ToList();
                CityBox.DisplayMemberPath = "CityName";
                CityBox.SelectedValuePath = "CityID";

                CategoryBox.ItemsSource = db.Categories.ToList();
                CategoryBox.DisplayMemberPath = "CategoryName";
                CategoryBox.SelectedValuePath = "CategoryID";

                TypeBox.ItemsSource = db.AdTypes.ToList();
                TypeBox.DisplayMemberPath = "TypeDescription";
                TypeBox.SelectedValuePath = "AdTypeID";

                StatusBox.ItemsSource = db.Statuses.ToList();
                StatusBox.DisplayMemberPath = "StatusName";
            }
        }
    }
}

```

```

        StatusBox.SelectedValuePath = "StatusID";
    }
}

private void LoadAdData()
{
    using (var db = new Entities())
    {
        var ad = db.Ads.FirstOrDefault(a => a.AdID == adId);
        if (ad == null) return;

        TitleBox.Text = ad.Title;
        DescriptionBox.Text = ad.Description;
        PriceBox.Text = ad.Price.ToString();

        CityBox.SelectedValue = ad.CityID;
        CategoryBox.SelectedValue = ad.CategoryID;
        TypeBox.SelectedValue = ad.AdTypeID;
        StatusBox.SelectedValue = ad.StatusID;

        if (!string.IsNullOrEmpty(ad.ImagePath))
            AdImage.Source = new BitmapImage(new Uri(ad.ImagePath, UriKind.RelativeOrAbsolute));
    }
}

private void ChooseImage_Click(object sender, RoutedEventArgs e)
{
    OpenFileDialog dlg = new OpenFileDialog();
    dlg.Filter = "Изображения|*.jpg;*.png;*.jpeg";

    if (dlg.ShowDialog() == true)
    {
        selectedImagePath = dlg.FileName;
        AdImage.Source = new BitmapImage(new Uri(selectedImagePath, UriKind.Absolute));
    }
}

private void Save_Click(object sender, RoutedEventArgs e)
{
    if (string.IsNullOrEmpty(TitleBox.Text) ||
        string.IsNullOrEmpty(PriceBox.Text))
    {
        MessageBox.Show("Заполните обязательные поля.\nНазвание и цена обязательны.",
            "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        return;
    }

    if (!decimal.TryParse(PriceBox.Text, out decimal price))
    {
        MessageBox.Show("Цена должна быть числом!", "Ошибка",
            MessageBoxButton.OK, MessageBoxImage.Error);
        return;
    }

    try
    {
        using (var db = new Entities())
        {
            Ads ad;

            if (adId == null)
            {
                ad = new Ads
                {
                    UserID = userId,
                    PostDate = DateTime.Now
                }
            }
            else
            {
                ad = db.Ads.FirstOrDefault(a => a.AdID == adId);
                ad.Title = TitleBox.Text;
                ad.Description = DescriptionBox.Text;
                ad.Price = price;
                ad.CategoryID = CategoryBox.SelectedValue;
                ad.AdTypeID = TypeBox.SelectedValue;
                ad.StatusID = StatusBox.SelectedValue;
                ad.ImagePath = selectedImagePath;
            }

            db.SaveChanges();
        }
    }
    catch { }
}

```

```

    };

    db.Ads.Add(ad);
}
else
{
    ad = db.Ads.FirstOrDefault(a => a.AdID == adId);
}

ad.Title = TitleBox.Text;
ad.Description = DescriptionBox.Text;
ad.Price = price;

ad.CityID = (int)CityBox.SelectedValue;
ad.CategoryID = (int)CategoryBox.SelectedValue;
ad.AdTypeID = (int)TypeBox.SelectedValue;
ad.StatusID = (int)StatusBox.SelectedValue;

if (!string.IsNullOrEmpty(selectedImagePath))
    ad.ImagePath = selectedImagePath;

db.SaveChanges();
}

MessageBox.Show("Объявление сохранено!",
    "Информация", MessageBoxButton.OK, MessageBoxImage.Information);

if (NavigationService.CanGoBack)
{
    NavigationService.GoBack();

    if (NavigationService.Content is UserAdsPage userAdsPage)
    {
        userAdsPage.LoadUserAds();
    }
}
}
catch (Exception ex)
{
    MessageBox.Show("Ошибка сохранения:\n" + ex.Message,
        "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
}
}
}
}

```

Листинг 17 – Словарь ресурсов Dictionary.xaml

```

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

    <Color x:Key="PrimaryTextColor">#37474F</Color>
    <Color x:Key="SecondaryTextColor">#78909C</Color>
    <Color x:Key="AccentColor">#FF9C1A</Color>
    <Color x:Key="DangerColor">#E53935</Color>
    <Color x:Key="SuccessColor">#4CAF50</Color>
    <Color x:Key="BorderGray">#DDDDDD</Color>

    <SolidColorBrush x:Key="PrimaryText" Color="{StaticResource PrimaryTextColor}" />
    <SolidColorBrush x:Key="SecondaryText" Color="{StaticResource SecondaryTextColor}" />
    <SolidColorBrush x:Key="AccentBrush" Color="{StaticResource AccentColor}" />
    <SolidColorBrush x:Key="DangerBrush" Color="{StaticResource DangerColor}" />
    <SolidColorBrush x:Key="SuccessBrush" Color="{StaticResource SuccessColor}" />
    <SolidColorBrush x:Key="BorderGrayBrush" Color="{StaticResource BorderGray}" />

    <Style TargetType="TextBlock">
        <Setter Property="FontFamily" Value="Tw Cen MT"/>
    </Style>
</ResourceDictionary>

```

```

        <Setter Property="Foreground" Value="{StaticResource PrimaryText}"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="Margin" Value="0 0 0 5"/>
        <Setter Property="TextWrapping" Value="Wrap"/>
    </Style>

    <Style x:Key="PageTitle" TargetType="TextBlock">
        <Setter Property="FontFamily" Value="Tw Cen MT"/>
        <Setter Property="Foreground" Value="{StaticResource PrimaryText}"/>
        <Setter Property="FontSize" Value="26"/>
        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="Margin" Value="0 0 0 16"/>
    </Style>

    <Style x:Key="SubTitle" TargetType="TextBlock">
        <Setter Property="FontFamily" Value="Tw Cen MT"/>
        <Setter Property="FontSize" Value="16"/>
        <Setter Property="FontWeight" Value="SemiBold"/>
        <Setter Property="Foreground" Value="{StaticResource PrimaryText}"/>
        <Setter Property="Margin" Value="0 0 0 16"/>
    </Style>

    <Style TargetType="Button">
        <Setter Property="FontFamily" Value="Tw Cen MT"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="Padding" Value="12 6"/>
        <Setter Property="Background" Value="{StaticResource AccentBrush}"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderThickness" Value="0"/>
        <Setter Property="Cursor" Value="Hand"/>
        <Setter Property="HorizontalContentAlignment" Value="Center"/>
    </Style>

    <Style x:Key="SuccessButton" TargetType="Button" BasedOn="{StaticResource {x:Type Button}}">
        <Setter Property="Background" Value="{StaticResource SuccessBrush}"/>
    </Style>

    <Style x:Key="DangerButton" TargetType="Button" BasedOn="{StaticResource {x:Type Button}}">
        <Setter Property="Background" Value="{StaticResource DangerBrush}"/>
    </Style>

    <Style TargetType="TextBox">
        <Setter Property="FontFamily" Value="Tw Cen MT"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="Padding" Value="6"/>
        <Setter Property="Margin" Value="0 0 0 10"/>
        <Setter Property="BorderBrush" Value="{StaticResource BorderGrayBrush}"/>
        <Setter Property="BorderThickness" Value="1"/>
    </Style>

    <Style TargetType="PasswordBox">
        <Setter Property="FontFamily" Value="Tw Cen MT"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="Padding" Value="6"/>
        <Setter Property="Margin" Value="0 0 0 10"/>
        <Setter Property="BorderBrush" Value="{StaticResource BorderGrayBrush}"/>
        <Setter Property="BorderThickness" Value="1"/>
    </Style>

    <Style TargetType="ComboBox">
        <Setter Property="FontFamily" Value="Tw Cen MT"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="Padding" Value="4"/>
        <Setter Property="Margin" Value="0 0 16 0"/>
        <Setter Property="BorderBrush" Value="{StaticResource BorderGrayBrush}"/>
        <Setter Property="BorderThickness" Value="1"/>
    </Style>

```

```

<Setter Property="Height" Value="30"/>
<Setter Property="Width" Value="150"/>
</Style>

<Style x:Key="AdCard" TargetType="Border">
  <Setter Property="CornerRadius" Value="8"/>
  <Setter Property="BorderBrush" Value="{ StaticResource BorderGrayBrush }"/>
  <Setter Property="BorderThickness" Value="1"/>
  <Setter Property="Padding" Value="10"/>
  <Setter Property="Margin" Value="0 0 0 16"/>
  <Setter Property="Background" Value="White"/>
</Style>

</ResourceDictionary>

```

Далее на рисунке 24 представлена диаграмма прецедентов (диаграмма вариантов использования).

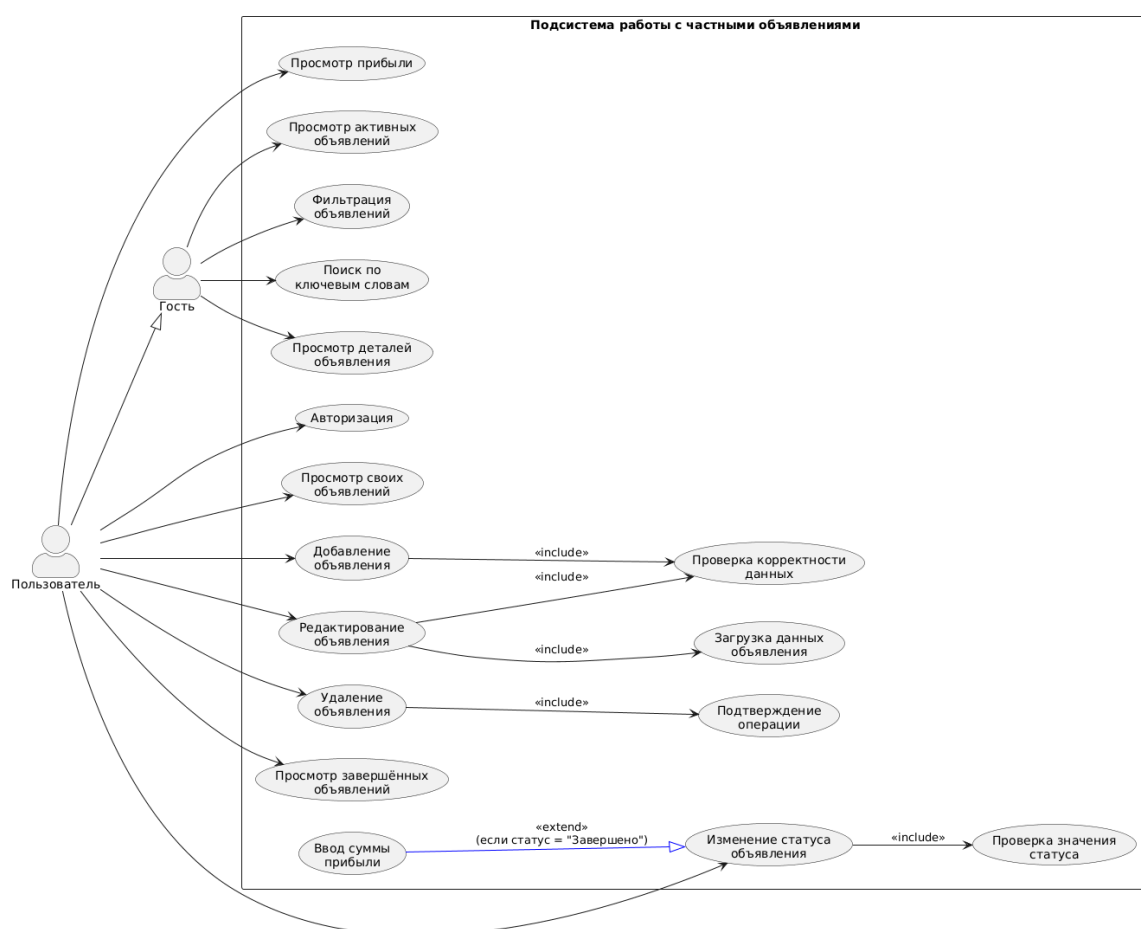


Рисунок 24 – Диаграмма прецедентов

Теперь проведем тестирование.

Аннотация теста

Название проекта

УП 02

Рабочая версия	1.0
Имя тестирующего	Медведева Диана
Дата(ы) теста	12.12.2025

Расшифровка тестовых информационных полей:

Поле	Описание
Название проекта	Название тестируемого проекта
Рабочая версия	Версия проекта/программного обеспечения (первый тест считается 1.0).
Имя тестирующего	Имя того, кто проводил тесты
Дата(ы) теста	Дата(ы) проведения тестов – это один или несколько дней. Если тесты проводились в более протяженный период времени, нужно отметить отдельную дату для каждого теста.
Тестовый пример #	Уникальный ID для каждого тестового примера. Следуйте некоторым конвенциям, чтобы указать типы тестов. Например, 'ТС_UI_1' означает 'user interface test case #1' (ТС_ПИ_1: тестовый случай пользовательского интерфейса#1)
Приоритет тестирования (Низкий/Средний/Высокий)	Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет.
Заголовок/название теста	Название тестового случая. Например, Подтвердите страницу авторизации с действительным именем пользователя и паролем.
Краткое изложение теста	Описание того, что должен достичь тест.
Этапы теста	Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея.
Тестовые данные	Перечислите/опишите все тестовые данные, используемые для данного тестового случая. Так, фактические используемые входные данные можно отслеживать по результатам тестирования. Например, Имя пользователя и пароль для подтверждения входа.
Ожидаемый результат	Каким должен быть вывод системы после выполнения теста? Подробно опишите ожидаемый результат, включая все сообщения/ошибки, которые должны отображаться на экране.

Фактический результат	Каким должен быть фактический результат после выполнения теста? Опишите любое релевантное поведение системы после выполнения теста.
Предварительное условие	Любые предварительные условия, которые должны быть выполнены до выполнения теста. Перечислите все предварительные условия для выполнения этого тестового случая.
Постусловие	Каким должно быть состояние системы после выполнения теста?
Статус (<i>Зачет/Незачет</i>)	Если фактический результат не соответствует ожидаемому результату, отметьте тест как неудачный. В ином случае обновление пройдено.
Примечания/комментарии	Используйте эту область для любых дополнительных замечаний/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами).

Тестовый пример #1:

Тестовый пример #	TC_FUNC_1
Приоритет тестирования	Высокий
Заголовок/название теста	Просмотр списка активных объявлений
Краткое изложение теста	Проверить, отображаются ли все активные объявления на главной странице без авторизации
Этапы теста	<ol style="list-style-type: none"> 1. Открыть приложение 2. Перейти на главную форму просмотра объявлений 3. Убедиться, что загружен список объявлений 4. Проверить, что статус отображаемых объявлений — «Активно»
Тестовые данные	нет
Ожидаемый результат	Все активные объявления отображаются корректно; не отображаются завершённые объявления
Фактический результат	Представлен на рисунке 25
Статус	Зачет
Предварительное условие	Система запущена. Есть данные об активных объявлениях в базе.
Постусловие	Состояние системы не изменяется
Примечания/комментарии	-

Тестовый пример #2:

Тестовый пример #	TC_FUNC_2
Приоритет тестирования	Средний
Заголовок/название теста	Поиск объявлений по ключевому слову
Краткое изложение теста	Проверить корректность работы поиска объявлений без авторизации
Этапы теста	<ol style="list-style-type: none"> 1. Открыть главную форму просмотра объявлений 2. Ввести в поле поиска ключевое слово: «куртка» 3. Проверить отображаемые результаты
Тестовые данные	Ключевое слово: «куртка»
Ожидаемый результат	Отображаются только те активные объявления, в названии или описании которых присутствует ключевое слово «куртка».
Фактический результат	Представлен на рисунке 26
Статус	Зачет

Предварительное условие	Есть активные объявления, содержащие ключевое слово
Постусловие	Система остаётся в прежнем состоянии
Примечания/комментарии	-

Тестовый пример #3:

Тестовый пример #	TC_AUTH_1
Приоритет тестирования	Высокий
Заголовок/название теста	Авторизация пользователя с корректными данными
Краткое изложение теста	Проверить возможность входа в систему при правильном логине и пароле
Этапы теста	<ol style="list-style-type: none"> 1. Открыть окно авторизации 2. Ввести корректный логин 3. Ввести корректный пароль 4. Нажать кнопку «Войти»
Тестовые данные	Логин: user1 Пароль: pass1
Ожидаемый результат	Пользователь успешно входит в систему; отображается главная форма или личные объявления
Фактический результат	Представлен на рисунке 27
Статус	Зачет
Предварительное условие	Учётная запись пользователя существует в базе данных
Постусловие	Пользователь авторизован, сессия активна
Примечания/комментарии	-

Тестовый пример #4:

Тестовый пример #	TC_AD_1
Приоритет тестирования	Высокий
Заголовок/название теста	Добавление нового объявления
Краткое изложение теста	Проверить корректность работы функции добавления объявления
Этапы теста	<ol style="list-style-type: none"> 1. Авторизоваться как пользователь 2. Открыть форму «Мои объявления» 3. Нажать кнопку «Добавить объявление» 4. Заполнить поля: название, описание, категория, город, тип, стоимость 5. Нажать «Сохранить» 6. Проверить, появилось ли объявление в списке пользователя

Тестовые данные	Название: "Продам телефон" Описание: "Новый, отличное состояние" Категория: "Электроника" Тип: "Продажа" Город: "Москва" Стоимость: 10000
Ожидаемый результат	Новое объявление сохраняется в базе и отображается в списке объявлений пользователя со статусом «Активно»
Фактический результат	Представлен на рисунках 28 - 30
Статус	Зачет
Предварительное условие	Пользователь авторизован
Постусловие	В базе данных появляется новая запись
Примечания/комментарии	-

Тестовый пример #5:

Test Case #	TC_AD_2
Приоритет тестирования	Высокий
Заголовок/название теста	Завершение объявления и ввод прибыли
Краткое изложение теста	Проверить работу изменения статуса объявления на «Завершено» и корректный ввод прибыли
Этапы теста	<ol style="list-style-type: none"> 1. Авторизоваться 2. Открыть список собственных объявлений 3. Выбрать активное объявление 4. Нажать «Редактировать» 5. Изменить статус на «Завершено» 6. В появившемся окне ввести сумму прибыли 7. Нажать «Сохранить» 8. Проверить, что объявление отображается в списке завершённых объявлений
Тестовые данные	Стоимость продажи: 15000 Прибыль: 5000
Ожидаемый результат	Статус объявления изменён на «Завершено»; прибыль сохранена; объявление отображается в списке завершённых
Фактический результат	Представлен на рисунках 31 - 32
Статус	Зачет
Предварительное условие	Минимум одно активное объявление существует в базе
Постусловие	В базе данных обновлены статус и прибыль объявления
Примечания/комментарии	-

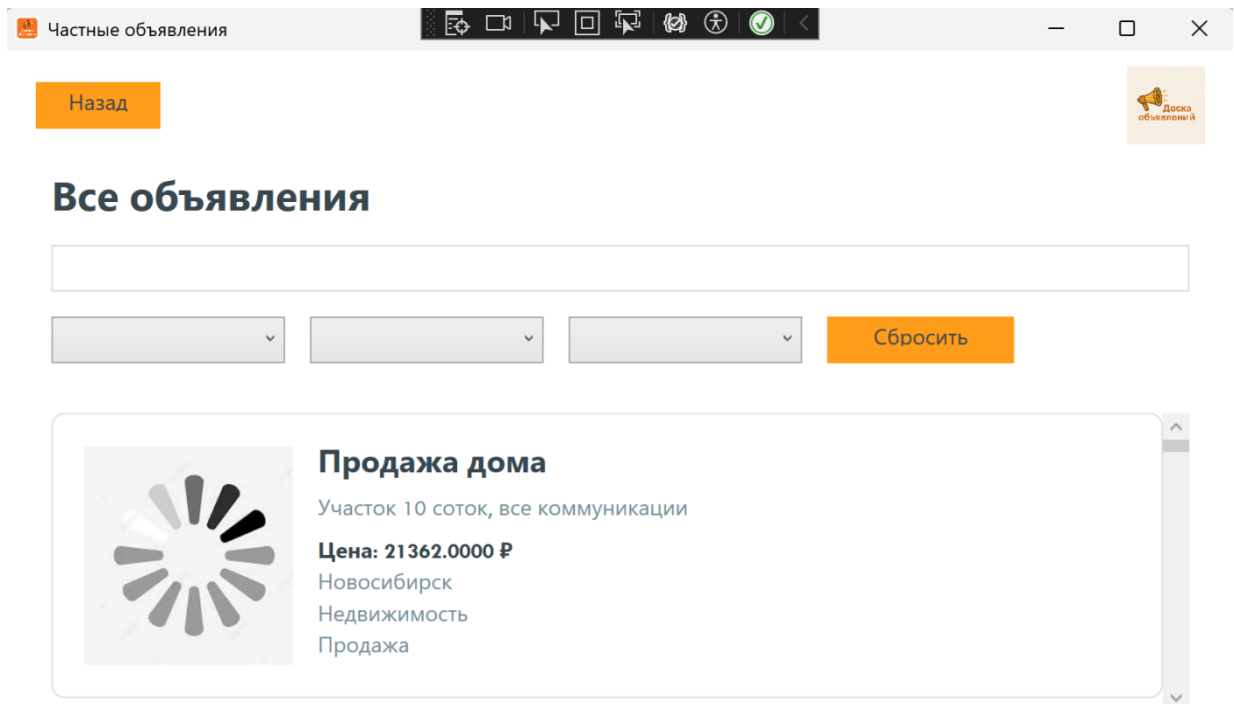


Рисунок 25 – Список активных объявлений

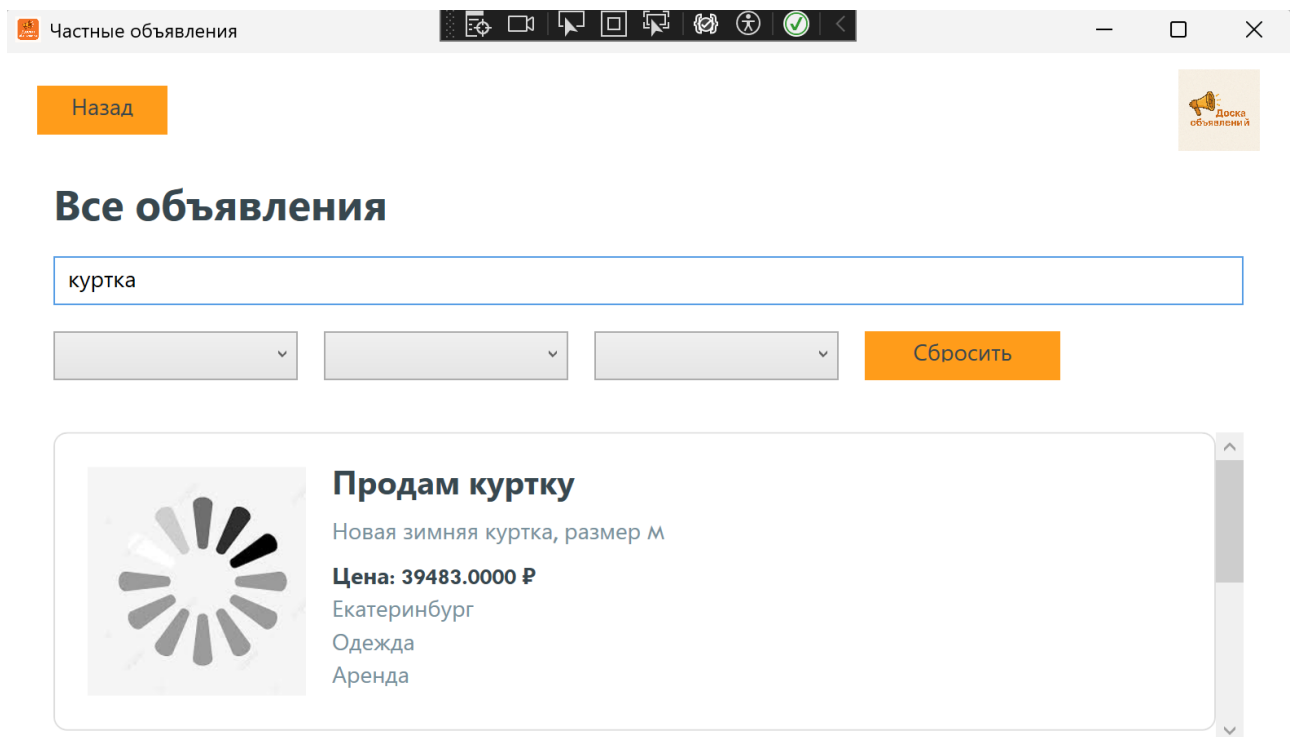


Рисунок 26 – Поиск по ключевому слову

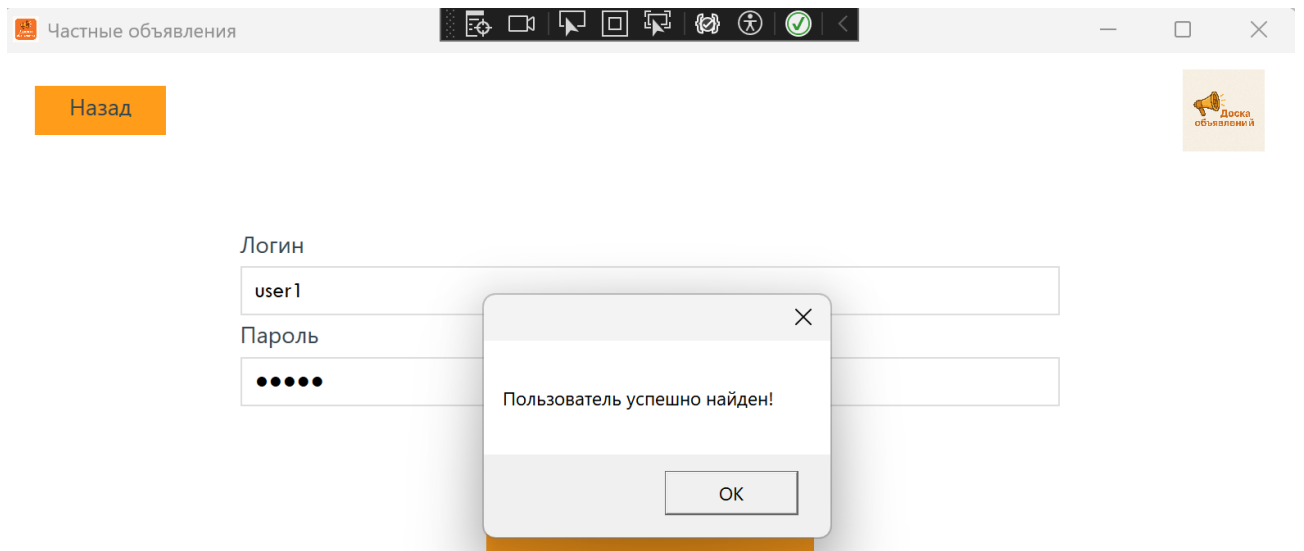


Рисунок 27 – Авторизация пользователя

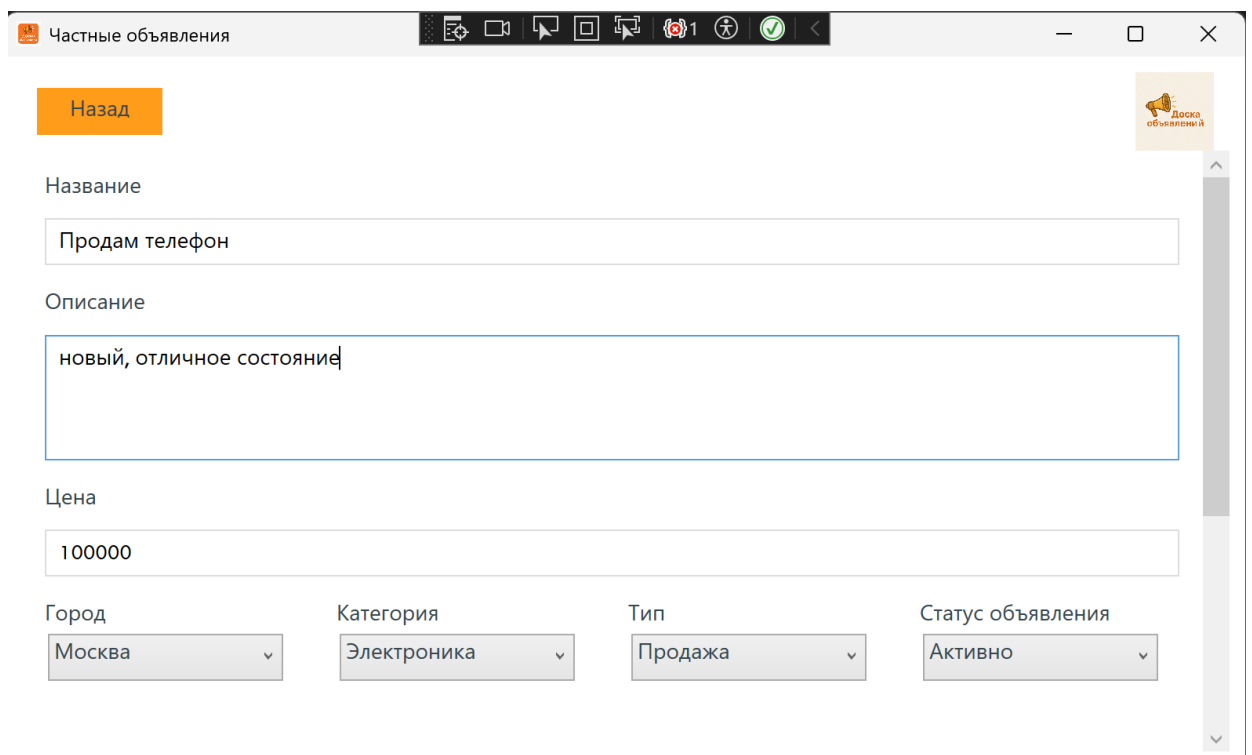


Рисунок 28 – Добавление объявления

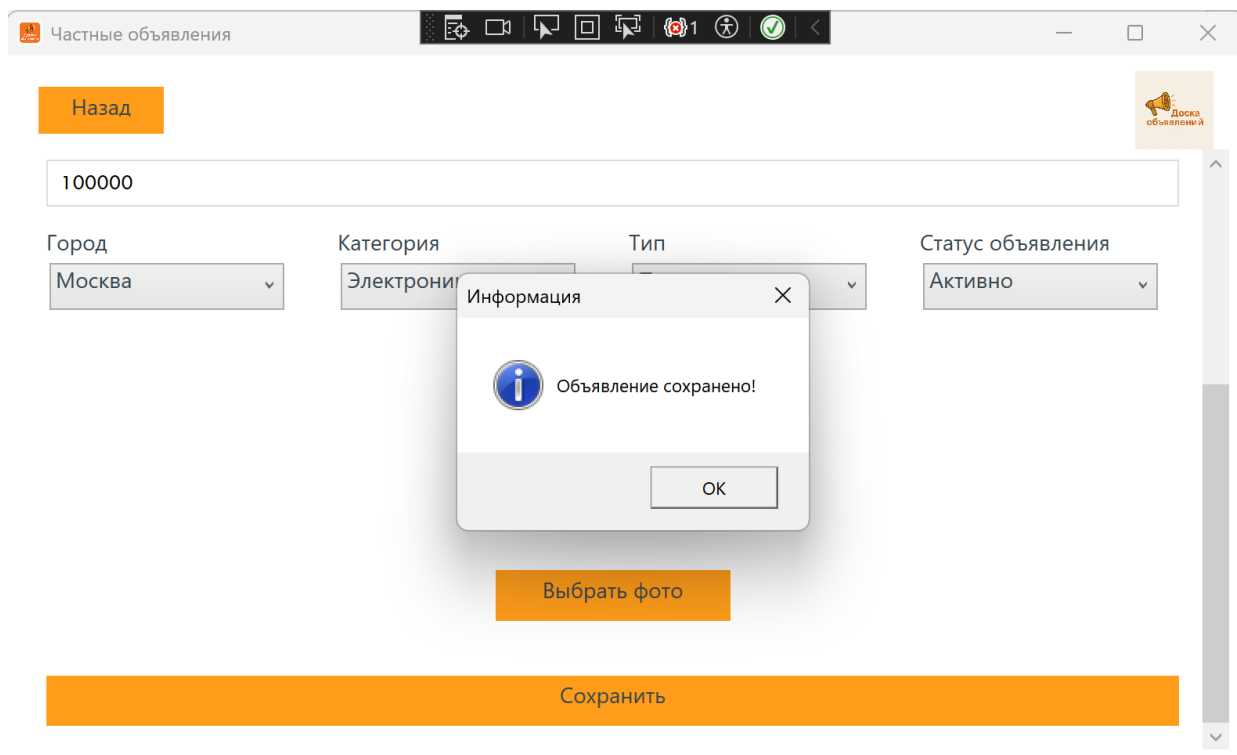


Рисунок 29 – Сообщение о успешном добавлении объявления

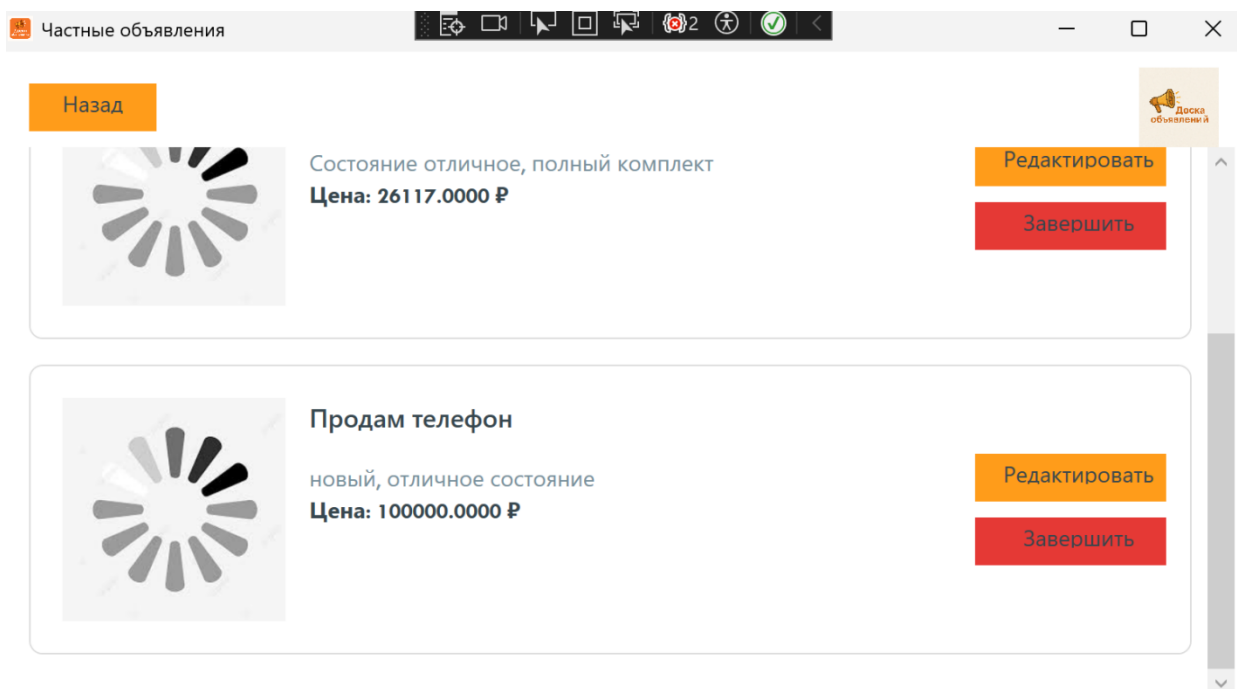


Рисунок 30 – Добавленное объявление на странице пользователя

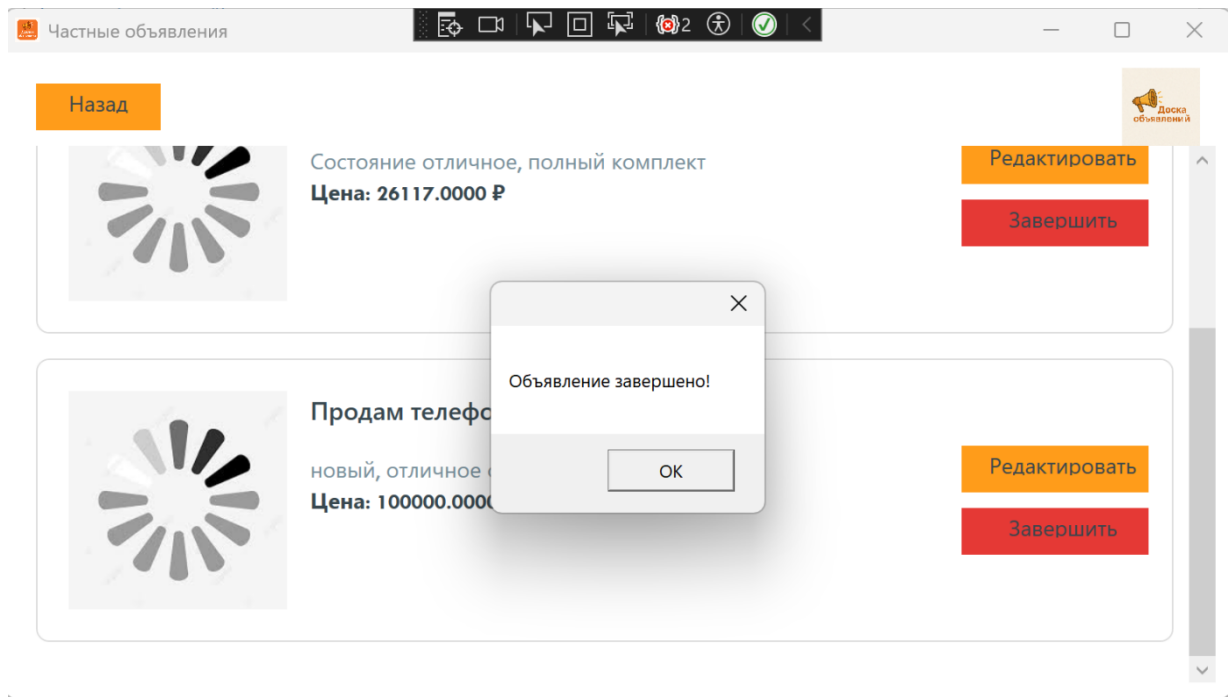


Рисунок 31 – Сообщение о завершении объявления

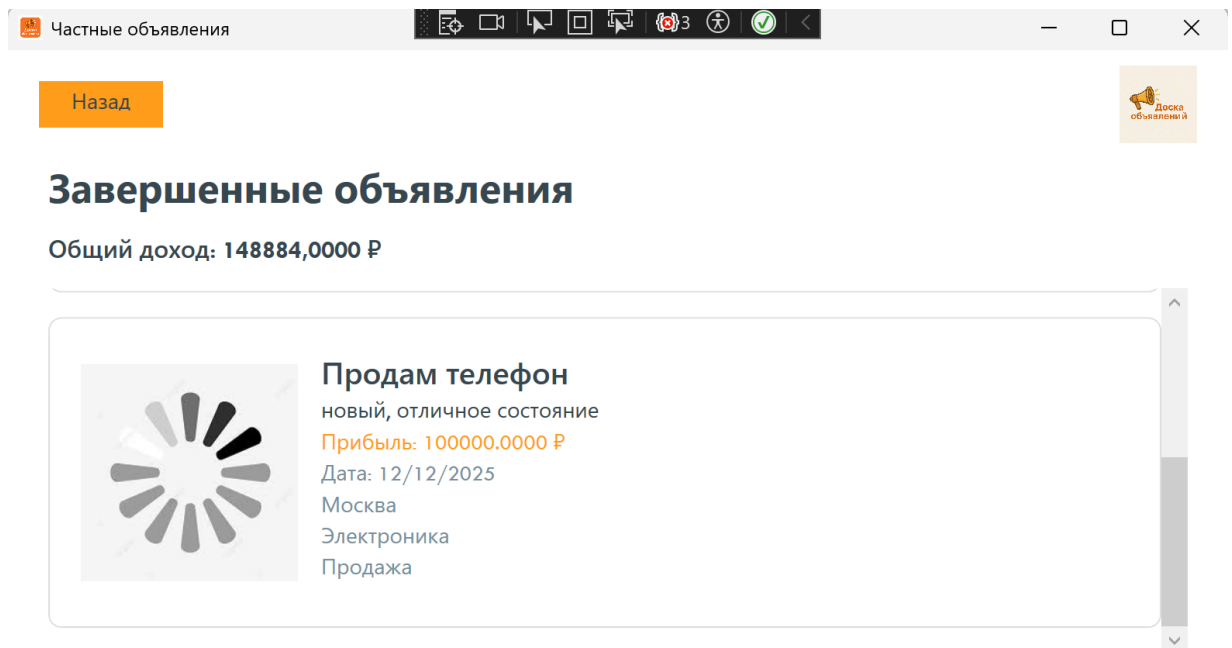


Рисунок 32 – Появляется в завершенных объявлениях