

**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.01.04 Системное программирование
(код) (наименование)

Выполнил:

Студент 4 курса 4ИСИП-321 учебной группы
(номер) (номер)


(подпись)

Д.О. Слуданов
(инициалы, фамилия)

Проверил:

Руководитель практики от Колледжа информатики и
программирования

Преподаватель 1КК, к.т.н., доцент Е.Л. Альшакова
(квалификационная категория или звание, должность) (инициалы, фамилия)

Перечень работ, выполненных в ходе учебной практики

№ п/п	Виды работ	Оценка
1	Проектирование базы данных и разработка объектов базы данных	Отлично
2	Разработка приложения с использованием WPF	Отлично

Содержание

Учебная практика 1	5
Проектирование базы данных и разработка объектов базы данных.....	5
Учебная практика 1	23
Разработка приложения с использованием WPF	23

Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)
Колледж информатики и программирования

ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 1

Тема: Проектирование базы данных и разработка объектов базы данных

Студент: Слуданов Д. О.

Группа: 4ИСИП-321

Преподаватель: Альшакова Е.Л.

Дата: 10.11.2024

Учебная практика 1

Проектирование базы данных и разработка объектов базы данных

Цель работы: разработать базу данных в соответствии с заданием, наполнить ее сущностями и атрибутами, реализовать ER-диаграмму.

Проанализируем предметную область. Основная цель учета заявок на ремонт оборудования – эффективное и оперативное осуществление ремонтных работ с минимизацией простоев и удовлетворением запросов клиентов или сотрудников. Эта предметная область широко используется в различных сферах деятельности, таких как сервисные услуги, производство, информационные технологии и другие. Предметная область учета заявок на ремонт оборудования касается процесса подачи, обработки и учёта заявок на ремонт различного оборудования.

Исходя из предметной области реализуем базу данных. На рисунке 1 представлена ER-диаграмма базы данных.

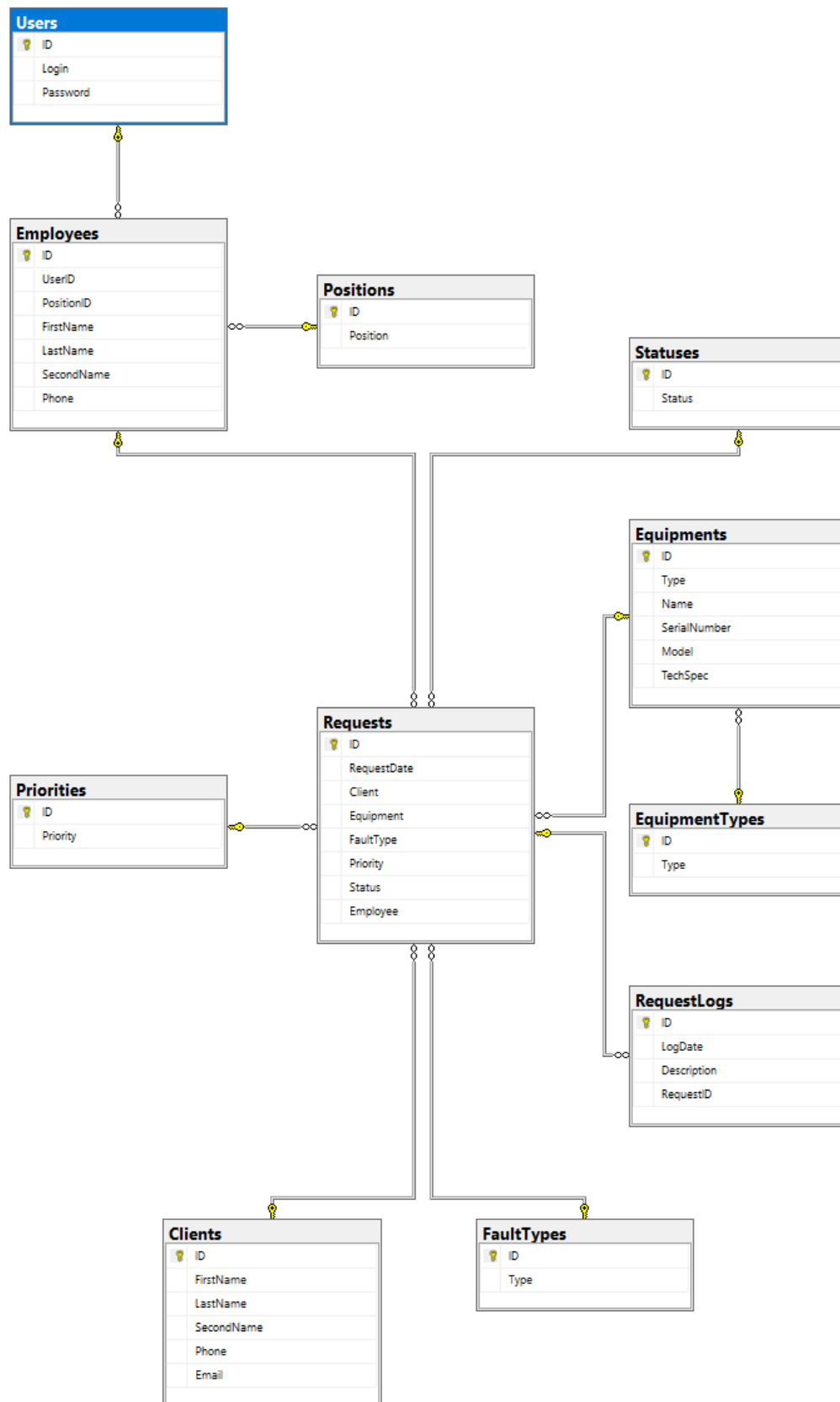


Рисунок 1 – ER-диаграмма базы данных

После того, как мы спроектировали базу данных, нам требуется заполнить ее начальными данными, чтобы в ходе разработки WPF приложения, мы могли понимать, как мы взаимодействуем с данными.

На рисунках 2-12 представлены данные для каждой сущности.

	ID	FirstName	LastName	SecondName	Phone	Email
1	2	Алексей	Петров	NULL	89001234568	petrov@example.com
2	3	Мария	Сидорова	NULL	89001234569	sidorova@example.com
3	4	Ольга	Кузнецова	NULL	89001234570	kuznetsova@example.com
4	6	Петров	Петр	Петрович	+78621231212	petrovich@yandex.ru
5	7	Иванов	Иван	Иванович	+79299172902	divanov@gmail.com

Рисунок 2 – Таблица «Clients»

	ID	UserID	PositionID	FirstName	LastName	SecondName	Phone
1	1	1	1	Иван	Иванов	NULL	89001234567
2	2	2	2	Алексей	Петров	NULL	89001234568
3	3	3	3	Мария	Сидорова	NULL	89001234569
4	4	4	4	Ольга	Кузнецова	NULL	89001234570
5	6	6	1	Петров	Петр	Петрович	+76584321212
6	7	7	4	Петров	Петр	Петрович	+79261577683
7	8	8	3	Петров	Петр	Петрович	+79299172902
8	9	9	1	Петров	Петр	Петрович	+78888888888

Рисунок 3 – Таблица «Employees»

	ID	Type	Name	SerialNumber	Model	TechSpec
1	2	2	Принтер 1	SN002	Model P1	Черно-белый, лазерный
2	3	3	Сканер 1	SN003	Model S1	1200dpi, цветной
3	4	4	Маршрутизатор 5	SN005	Model R1	Wi-Fi, 5GHz

Рисунок 4 – Таблица «Equipments»

Results		Messages
	ID	Type
1	6	Клавиатура
2	4	Маршрутизатор
3	5	Монитор
4	2	Принтер
5	3	Сканер
6	7	тест

Рисунок 5 – Таблица «EquipmentTypes»

	ID	Типе
1	1	Аппаратная ошибка
2	5	Механическая ошибка
3	4	Ошибка питания
4	2	Программная ошибка
5	3	Сетевая ошибка

Рисунок 6 – Таблица «FaultTypes»

	ID	Position
1	3	Инженер
2	1	Менеджер
3	4	Менеджер по продажам
4	5	Сотрудник поддержки
5	2	Техник

Рисунок 7 – Таблица «Positions»

	ID	Priority
1	3	Высокий
2	5	Критический
3	1	Низкий
4	2	Средний
5	4	Срочный

Рисунок 8 – Таблица «Priorities»

	ID	LogDate	Description	RequestID
1	2	2024-12-02 00:00:00.000	Запрос принят	2
2	3	2024-12-03 00:00:00.000	Запрос принят	3
3	6	2024-12-06 07:26:46.487	Создана новая заявка с ID 9.	9

Рисунок 9 – Таблица «RequestLogs»

	ID	RequestDate	Client	Equipment	FaultType	Priority	Status	Employee
1	2	2024-12-02 00:00:00.000	2	2	2	2	2	3
2	3	2024-12-03 00:00:00.000	3	3	3	1	3	4
3	8	2024-12-06 07:18:19.577	7	3	5	3	1	3
4	9	2024-12-06 07:26:46.323	4	2	5	3	1	1

Рисунок 10 – Таблица «Requests»

	ID	Status
1	5	Архивировано
2	1	В ожидании
3	2	В процессе
4	3	Завершено
5	4	Отменено

Рисунок 11 – Таблица «Statuses»

	ID	Login	Password
1	1	ivanov	password1
2	2	petrov	password2
3	3	sidorova	password3
4	4	kuznetsova	password4
5	5	mikhailov	password5
6	6	testuser1	E38AD214943DAAD1D64C102FAEC29DE4AFE9DA3D
7	7	testuser2	2AA60A8FF7FCD473D321E0146AFD9E26DF395147
8	8	testovich	7C4A8D09CA3762AF61E59520943DC26494F8941B
9	9	testuser011	7C4A8D09CA3762AF61E59520943DC26494F8941B

Рисунок 12 – Таблица «Users»

Также, был создан скрипт базы данных, который представлен ниже в листинге 1.

Листинг 1 – Скрипт базы данных

```
USE [TechnoService]
GO
/***** Object:  Table [dbo].[Clients]      Script Date: 19.12.24 14:58:56
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Clients](
```

```

        [ID] [int] IDENTITY(1,1) NOT NULL,
        [FirstName] [nvarchar](100) NOT NULL,
        [LastName] [nvarchar](100) NOT NULL,
        [SecondName] [nvarchar](100) NULL,
        [Phone] [nvarchar](15) NOT NULL,
        [Email] [nvarchar](75) 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].[Employees]      Script Date: 19.12.24 14:58:56
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Employees](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [UserID] [int] NOT NULL,
    [PositionID] [int] NOT NULL,
    [FirstName] [nvarchar](70) NOT NULL,
    [LastName] [nvarchar](70) NOT NULL,
    [SecondName] [nvarchar](70) NULL,
    [Phone] [nvarchar](15) 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].[Equipments]      Script Date: 19.12.24 14:58:56
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Equipments](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Type] [int] NOT NULL,

```

```

        [Name] [nvarchar](50) NOT NULL,
        [SerialNumber] [nvarchar](50) NOT NULL,
        [Model] [nvarchar](100) NULL,
        [TechSpec] [nvarchar](255) 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].[EquipmentTypes]      Script Date: 19.12.24
14:58:56 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[EquipmentTypes](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Type] [nvarchar](50) 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].[FaultTypes]      Script Date: 19.12.24 14:58:56
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[FaultTypes](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Type] [nvarchar](50) 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].[Positions]      Script Date: 19.12.24 14:58:56
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Positions](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Position] [nvarchar](50) 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].[Priorities]      Script Date: 19.12.24 14:58:56
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Priorities](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Priority] [nvarchar](20) 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].[RequestLogs]      Script Date: 19.12.24 14:58:56
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[RequestLogs](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [LogDate] [datetime] NOT NULL,
    [Description] [nvarchar](max) NULL,

```

```

        [RequestID] [int] 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] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object:  Table [dbo].[Requests]      Script Date: 19.12.24 14:58:56
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Requests] (
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [RequestDate] [datetime] NOT NULL,
    [Client] [int] NOT NULL,
    [Equipment] [int] NOT NULL,
    [FaultType] [int] NOT NULL,
    [Priority] [int] NOT NULL,
    [Status] [int] NOT NULL,
    [Employee] [int] 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].[Statuses]      Script Date: 19.12.24 14:58:56
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Statuses] (
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Status] [nvarchar] (50) 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: 19.12.24 14:58:56 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Users](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Login] [varchar](50) NOT NULL,
    [Password] [varchar](64) 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
SET IDENTITY_INSERT [dbo].[Clients] ON
GO
INSERT [dbo].[Clients] ([ID], [FirstName], [LastName], [SecondName], [Phone],
[Email]) VALUES (2, N'Алексей', N'Петров', NULL, N'89001234568',
N'petrov@example.com')
GO
INSERT [dbo].[Clients] ([ID], [FirstName], [LastName], [SecondName], [Phone],
[Email]) VALUES (3, N'Мария', N'Сидорова', NULL, N'89001234569',
N'sidorova@example.com')
GO
INSERT [dbo].[Clients] ([ID], [FirstName], [LastName], [SecondName], [Phone],
[Email]) VALUES (4, N'Ольга', N'Кузнецова', NULL, N'89001234570',
N'kuznetsova@example.com')
GO
INSERT [dbo].[Clients] ([ID], [FirstName], [LastName], [SecondName], [Phone],
[Email]) VALUES (6, N'Петров', N'Петр', N'Петрович', N'+78621231212',
N'petrovich@yandex.ru')
GO
INSERT [dbo].[Clients] ([ID], [FirstName], [LastName], [SecondName], [Phone],
[Email]) VALUES (7, N'Иванов', N'Иван', N'Иванович', N'+79299172902',
N'divanov@gmail.com')
GO
SET IDENTITY_INSERT [dbo].[Clients] OFF

```

```

GO
SET IDENTITY_INSERT [dbo].[Employees] ON
GO
INSERT [dbo].[Employees] ([ID], [UserID], [PositionID], [FirstName],
[LastName], [SecondName], [Phone]) VALUES (1, 1, 1, N'Иван', N'Иванов', NULL,
N'89001234567')
GO
INSERT [dbo].[Employees] ([ID], [UserID], [PositionID], [FirstName],
[LastName], [SecondName], [Phone]) VALUES (2, 2, 2, N'Алексей', N'Петров',
NULL, N'89001234568')
GO
INSERT [dbo].[Employees] ([ID], [UserID], [PositionID], [FirstName],
[LastName], [SecondName], [Phone]) VALUES (3, 3, 3, N'Мария', N'Сидорова',
NULL, N'89001234569')
GO
INSERT [dbo].[Employees] ([ID], [UserID], [PositionID], [FirstName],
[LastName], [SecondName], [Phone]) VALUES (4, 4, 4, N'Ольга', N'Кузнецова',
NULL, N'89001234570')
GO
INSERT [dbo].[Employees] ([ID], [UserID], [PositionID], [FirstName],
[LastName], [SecondName], [Phone]) VALUES (6, 6, 1, N'Петров', N'Петр',
N'Петрович', N'+76584321212')
GO
INSERT [dbo].[Employees] ([ID], [UserID], [PositionID], [FirstName],
[LastName], [SecondName], [Phone]) VALUES (7, 7, 4, N'Петров', N'Петр',
N'Петрович', N'+79261577683')
GO
INSERT [dbo].[Employees] ([ID], [UserID], [PositionID], [FirstName],
[LastName], [SecondName], [Phone]) VALUES (8, 8, 3, N'Петров', N'Петр',
N'Петрович', N'+79299172902')
GO
INSERT [dbo].[Employees] ([ID], [UserID], [PositionID], [FirstName],
[LastName], [SecondName], [Phone]) VALUES (9, 9, 1, N'Петров', N'Петр',
N'Петрович', N'+78888888888')
GO
SET IDENTITY_INSERT [dbo].[Employees] OFF
GO
SET IDENTITY_INSERT [dbo].[Equipments] ON
GO
INSERT [dbo].[Equipments] ([ID], [Type], [Name], [SerialNumber], [Model],
[TechSpec]) VALUES (2, 2, N'Принтер 1', N'SN002', N'Model P1', N'Черно-белый,
лазерный')
GO

```

```
INSERT [dbo].[Equipments] ([ID], [Type], [Name], [SerialNumber], [Model],  
[TechSpec]) VALUES (3, 3, N'Сканер 1', N'SN003', N'Model S1', N'1200dpi,  
цветной')  
GO  
INSERT [dbo].[Equipments] ([ID], [Type], [Name], [SerialNumber], [Model],  
[TechSpec]) VALUES (4, 4, N'Маршрутизатор 5', N'SN005', N'Model R1', N'Wi-Fi,  
5GHz')  
GO  
SET IDENTITY_INSERT [dbo].[Equipments] OFF  
GO  
SET IDENTITY_INSERT [dbo].[EquipmentTypes] ON  
GO  
INSERT [dbo].[EquipmentTypes] ([ID], [Type]) VALUES (6, N'Клавиатура')  
GO  
INSERT [dbo].[EquipmentTypes] ([ID], [Type]) VALUES (4, N'Маршрутизатор')  
GO  
INSERT [dbo].[EquipmentTypes] ([ID], [Type]) VALUES (5, N'Монитор')  
GO  
INSERT [dbo].[EquipmentTypes] ([ID], [Type]) VALUES (2, N'Принтер')  
GO  
INSERT [dbo].[EquipmentTypes] ([ID], [Type]) VALUES (3, N'Сканер')  
GO  
INSERT [dbo].[EquipmentTypes] ([ID], [Type]) VALUES (7, N'тест')  
GO  
SET IDENTITY_INSERT [dbo].[EquipmentTypes] OFF  
GO  
SET IDENTITY_INSERT [dbo].[FaultTypes] ON  
GO  
INSERT [dbo].[FaultTypes] ([ID], [Type]) VALUES (1, N'Аппаратная ошибка')  
GO  
INSERT [dbo].[FaultTypes] ([ID], [Type]) VALUES (5, N'Механическая ошибка')  
GO  
INSERT [dbo].[FaultTypes] ([ID], [Type]) VALUES (4, N'Ошибка питания')  
GO  
INSERT [dbo].[FaultTypes] ([ID], [Type]) VALUES (2, N'Программная ошибка')  
GO  
INSERT [dbo].[FaultTypes] ([ID], [Type]) VALUES (3, N'Сетевая ошибка')  
GO  
SET IDENTITY_INSERT [dbo].[FaultTypes] OFF  
GO  
SET IDENTITY_INSERT [dbo].[Positions] ON  
GO  
INSERT [dbo].[Positions] ([ID], [Position]) VALUES (3, N'Инженер')  
GO  
INSERT [dbo].[Positions] ([ID], [Position]) VALUES (1, N'Менеджер')
```



```
GO
INSERT [dbo].[Positions] ([ID], [Position]) VALUES (4, N'Менеджер по
продажам')
GO
INSERT [dbo].[Positions] ([ID], [Position]) VALUES (5, N'Сотрудник поддержки')
GO
INSERT [dbo].[Positions] ([ID], [Position]) VALUES (2, N'Техник')
GO
SET IDENTITY_INSERT [dbo].[Positions] OFF
GO
SET IDENTITY_INSERT [dbo].[Priorities] ON
GO
INSERT [dbo].[Priorities] ([ID], [Priority]) VALUES (3, N'Высокий')
GO
INSERT [dbo].[Priorities] ([ID], [Priority]) VALUES (5, N'Критический')
GO
INSERT [dbo].[Priorities] ([ID], [Priority]) VALUES (1, N'Низкий')
GO
INSERT [dbo].[Priorities] ([ID], [Priority]) VALUES (2, N'Средний')
GO
INSERT [dbo].[Priorities] ([ID], [Priority]) VALUES (4, N'Срочный')
GO
SET IDENTITY_INSERT [dbo].[Priorities] OFF
GO
SET IDENTITY_INSERT [dbo].[RequestLogs] ON
GO
INSERT [dbo].[RequestLogs] ([ID], [LogDate], [Description], [RequestID])
VALUES (2, CAST(N'2024-12-02T00:00:00.000' AS DateTime), N'Запрос принят', 2)
GO
INSERT [dbo].[RequestLogs] ([ID], [LogDate], [Description], [RequestID])
VALUES (3, CAST(N'2024-12-03T00:00:00.000' AS DateTime), N'Запрос принят', 3)
GO
INSERT [dbo].[RequestLogs] ([ID], [LogDate], [Description], [RequestID])
VALUES (6, CAST(N'2024-12-06T07:26:46.487' AS DateTime), N'Создана новая
заявка с ID 9.', 9)
GO
SET IDENTITY_INSERT [dbo].[RequestLogs] OFF
GO
SET IDENTITY_INSERT [dbo].[Requests] ON
GO
INSERT [dbo].[Requests] ([ID], [RequestDate], [Client], [Equipment],
[FaultType], [Priority], [Status], [Employee]) VALUES (2, CAST(N'2024-12-
02T00:00:00.000' AS DateTime), 2, 2, 2, 2, 2, 3)
GO
```

```
INSERT [dbo].[Requests] ([ID], [RequestDate], [Client], [Equipment],
[FaultType], [Priority], [Status], [Employee]) VALUES (3, CAST(N'2024-12-
03T00:00:00.000' AS DateTime), 3, 3, 3, 1, 3, 4)
GO
INSERT [dbo].[Requests] ([ID], [RequestDate], [Client], [Equipment],
[FaultType], [Priority], [Status], [Employee]) VALUES (8, CAST(N'2024-12-
06T07:18:19.577' AS DateTime), 7, 3, 5, 3, 1, 3)
GO
INSERT [dbo].[Requests] ([ID], [RequestDate], [Client], [Equipment],
[FaultType], [Priority], [Status], [Employee]) VALUES (9, CAST(N'2024-12-
06T07:26:46.323' AS DateTime), 4, 2, 5, 3, 1, 1)
GO
SET IDENTITY_INSERT [dbo].[Requests] OFF
GO
SET IDENTITY_INSERT [dbo].[Statuses] ON
GO
INSERT [dbo].[Statuses] ([ID], [Status]) VALUES (5, N'Архивировано')
GO
INSERT [dbo].[Statuses] ([ID], [Status]) VALUES (1, N'В ожидании')
GO
INSERT [dbo].[Statuses] ([ID], [Status]) VALUES (2, N'В процессе')
GO
INSERT [dbo].[Statuses] ([ID], [Status]) VALUES (3, N'Завершено')
GO
INSERT [dbo].[Statuses] ([ID], [Status]) VALUES (4, N'Отменено')
GO
SET IDENTITY_INSERT [dbo].[Statuses] OFF
GO
SET IDENTITY_INSERT [dbo].[Users] ON
GO
INSERT [dbo].[Users] ([ID], [Login], [Password]) VALUES (1, N'ivanov',
N'password1')
GO
INSERT [dbo].[Users] ([ID], [Login], [Password]) VALUES (2, N'petrov',
N'password2')
GO
INSERT [dbo].[Users] ([ID], [Login], [Password]) VALUES (3, N'sidorova',
N'password3')
GO
INSERT [dbo].[Users] ([ID], [Login], [Password]) VALUES (4, N'kuznetsova',
N'password4')
GO
INSERT [dbo].[Users] ([ID], [Login], [Password]) VALUES (5, N'mikhailov',
N'password5')
GO
```

```

INSERT [dbo].[Users] ([ID], [Login], [Password]) VALUES (6, N'testuser1',
N'E38AD214943DAAD1D64C102FAEC29DE4AFE9DA3D')
GO
INSERT [dbo].[Users] ([ID], [Login], [Password]) VALUES (7, N'testuser2',
N'2AA60A8FF7FCD473D321E0146AFD9E26DF395147')
GO
INSERT [dbo].[Users] ([ID], [Login], [Password]) VALUES (8, N'testovich',
N'7C4A8D09CA3762AF61E59520943DC26494F8941B')
GO
INSERT [dbo].[Users] ([ID], [Login], [Password]) VALUES (9, N'testuser011',
N'7C4A8D09CA3762AF61E59520943DC26494F8941B')
GO
SET IDENTITY_INSERT [dbo].[Users] OFF
GO
SET ANSI_PADDING ON
GO
/***** Object:  Index [UQ__Equipmen__048A00087C1FF1D1]      Script Date:
19.12.24 14:58:56 *****/
ALTER TABLE [dbo].[Equipments] ADD UNIQUE NONCLUSTERED
(
    [SerialNumber] 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
SET ANSI_PADDING ON
GO
/***** Object:  Index [UQ__Equipmen__F9B8A48B11CD8A0B]      Script Date:
19.12.24 14:58:56 *****/
ALTER TABLE [dbo].[EquipmentTypes] ADD UNIQUE NONCLUSTERED
(
    [Type] 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
SET ANSI_PADDING ON
GO
/***** Object:  Index [UQ__FaultTyp__F9B8A48BD3477207]      Script Date:
19.12.24 14:58:56 *****/
ALTER TABLE [dbo].[FaultTypes] ADD UNIQUE NONCLUSTERED
(
    [Type] 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
SET ANSI_PADDING ON
GO
/***** Object:  Index [UQ__Position__5A8B58B84D9A2B9B]      Script Date:
19.12.24 14:58:56 *****/
ALTER TABLE [dbo].[Positions] ADD UNIQUE NONCLUSTERED
(
    [Position] 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
SET ANSI_PADDING ON
GO
/***** Object:  Index [UQ__Prioriti__534DF97BBED279A6]      Script Date:
19.12.24 14:58:56 *****/
ALTER TABLE [dbo].[Priorities] ADD UNIQUE NONCLUSTERED
(
    [Priority] 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
SET ANSI_PADDING ON
GO
/***** Object:  Index [UQ__Statuses__3A15923F4DBACE62]      Script Date:
19.12.24 14:58:56 *****/
ALTER TABLE [dbo].[Statuses] ADD UNIQUE NONCLUSTERED
(
    [Status] 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].[RequestLogs] ADD DEFAULT (getdate()) FOR [LogDate]
GO
ALTER TABLE [dbo].[Requests] ADD DEFAULT (getdate()) FOR [RequestDate]
GO
ALTER TABLE [dbo].[Employees] WITH CHECK ADD FOREIGN KEY([PositionID])
REFERENCES [dbo].[Positions] ([ID])
ON DELETE CASCADE
GO

```

```
ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [FK_Employees_Users]
FOREIGN KEY([UserID])
REFERENCES [dbo].[Users] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Employees] CHECK CONSTRAINT [FK_Employees_Users]
GO
ALTER TABLE [dbo].[Equipments] WITH CHECK ADD FOREIGN KEY([Type])
REFERENCES [dbo].[EquipmentTypes] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[RequestLogs] WITH CHECK ADD FOREIGN KEY([RequestID])
REFERENCES [dbo].[Requests] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Requests] WITH CHECK ADD FOREIGN KEY([Client])
REFERENCES [dbo].[Clients] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Requests] WITH CHECK ADD FOREIGN KEY([Employee])
REFERENCES [dbo].[Employees] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Requests] WITH CHECK ADD FOREIGN KEY([Equipment])
REFERENCES [dbo].[Equipments] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Requests] WITH CHECK ADD FOREIGN KEY([FaultType])
REFERENCES [dbo].[FaultTypes] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Requests] WITH CHECK ADD FOREIGN KEY([Priority])
REFERENCES [dbo].[Priorities] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Requests] WITH CHECK ADD FOREIGN KEY([Status])
REFERENCES [dbo].[Statuses] ([ID])
ON DELETE CASCADE
GO
```

Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)
Колледж информатики и программирования

ОТЧЕТ О ВЫПОЛНЕНИИ ЗАДАНИЯ № 2

Тема: Разработка кода программного продукта

Студент: Слуданов Д. О.

Группа: 4ИСИП-321

Преподаватель: Альшакова Е.Л.

Дата: 03.12.2024

Учебная практика 1

Разработка приложения с использованием WPF

Цель работы: разработать WPF-приложение в соответствии с заданием, наполнить ее сущностями и атрибутами, написать листинг программы.

Было реализовано WPF-приложение под требования заказчика. При запуске данного приложения, у нас открывается окно с авторизацией в приложения. В случае, если у нас нет пользователя в системе, то мы можем зарегистрироваться. На рисунках 13-14 представлены окна авторизации и регистрации. А также, продемонстрирован код xaml для этих страниц в листингах 2-3 и логика работы данных окон в листингах 4-5.

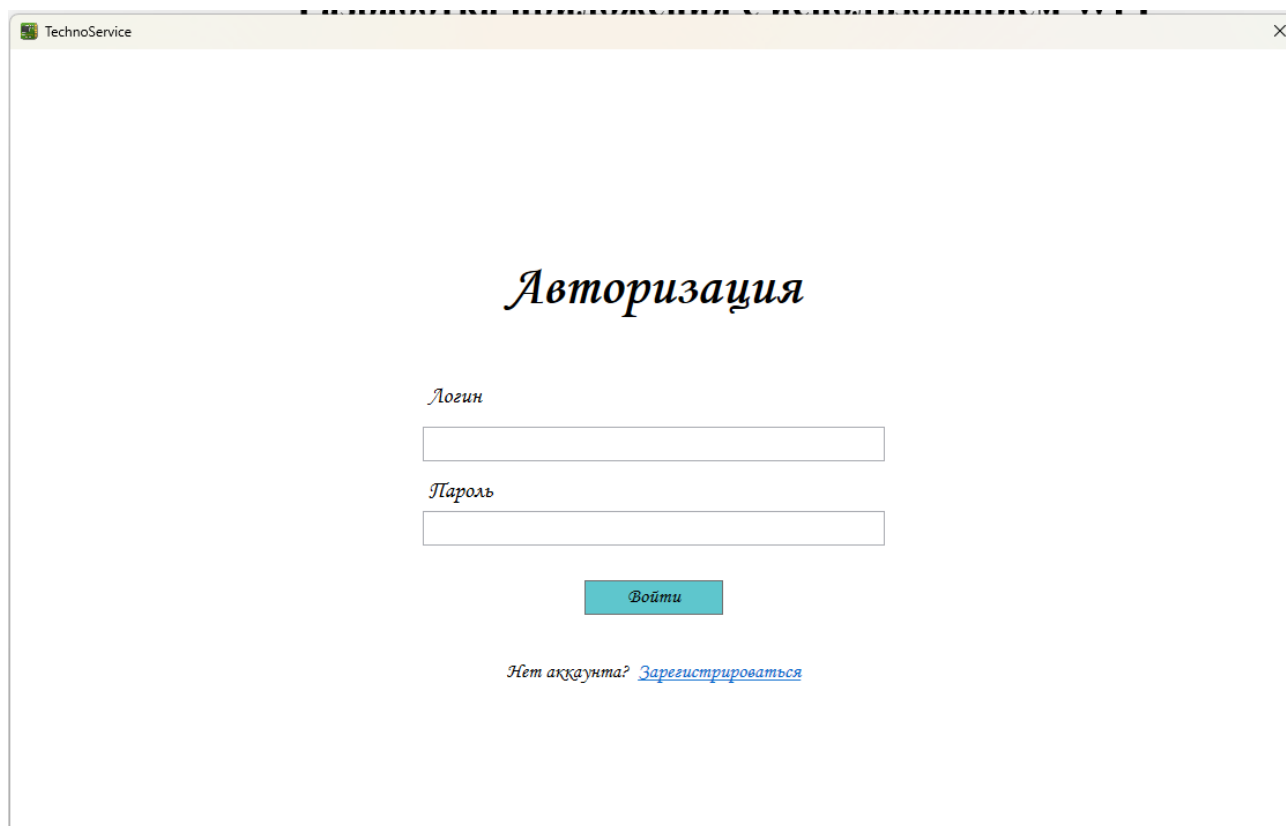


Рисунок 13 – Окно авторизации

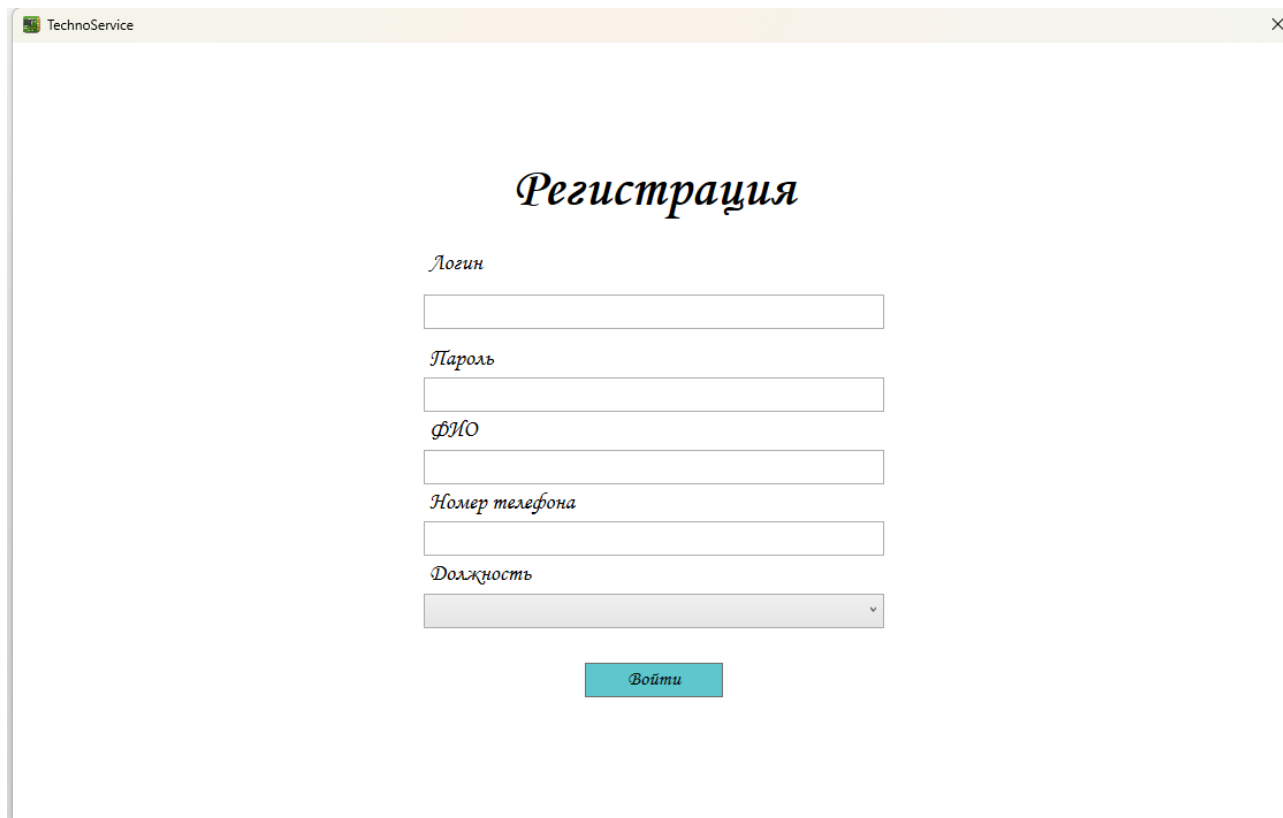


Рисунок 14 – Окно регистрации

Листинг 2 – Код xaml для окна авторизации

```
<Window x:Class="TechnoService.Auth"
        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:TechnoService"
        mc:Ignorable="d"
        Title="TechnoService" Height="720" Width="1130"
        WindowStartupLocation="CenterScreen" Closing="Window_Closing"
        ResizeMode="NoResize">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions>

        <StackPanel Grid.Row="1" VerticalAlignment="Center"
        HorizontalAlignment="Center" Width="400">
            <TextBlock Grid.Row="0" Text="Авторизация" FontSize="48"
            FontWeight="Bold" HorizontalAlignment="Center" Margin="0,50"
            Style="{StaticResource TextBlock}"/>
```



```

        <Label Content="Логин" FontSize="20" HorizontalAlignment="Left"
Style="{StaticResource Label}"/>

        <TextBox Name="LoginTextBox" FontSize="18" Height="30"
HorizontalAlignment="Stretch" Margin="0,10" VerticalContentAlignment="Center"
FontFamily="Monotype Corsiva"/>

        <Label Content="Пароль" FontSize="20" HorizontalAlignment="Left"
Style="{StaticResource Label}"/>

        <PasswordBox Name="PasswordBox" FontSize="18" Height="30"
HorizontalAlignment="Stretch" VerticalContentAlignment="Center"/>

        <Button x:Name="btnLogin" Content="Войти"
HorizontalAlignment="Center" Margin="0,30" Height="30" Width="120"
FontSize="18" Style="{StaticResource Button}" Click="btnLogin_Click"/>

        <TextBlock HorizontalAlignment="Center" Style="{StaticResource
TextBlock}" FontSize="18" Margin="0 10 0 0">
            <Run Text="Нет аккаунта? "/>
            <Hyperlink
Click="Hyperlink_Click">Зарегистрироваться</Hyperlink>
        </TextBlock>
    </StackPanel>
</Grid>
</Window>

```

Листинг 3 – Код xaml для окна регистрации

```

<Window x:Class="TechnoService.Reg"
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:TechnoService"
mc:Ignorable="d"
Title="TechnoService" Height="720" Width="1130"
WindowStartupLocation="CenterScreen" Closing="Window_Closing"
ResizeMode="NoResize">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions>

        <StackPanel Grid.Row="1" VerticalAlignment="Center"
HorizontalAlignment="Center" Width="400">

```

```

        <TextBlock      Grid.Row="0"      Text="Регистрация"      FontSize="48"
FontWeight="Bold"      HorizontalAlignment="Center"      Margin="0,20"
Style="{StaticResource TextBlock}"/>

        <Label      Content="Логин"      FontSize="20"      HorizontalAlignment="Left"
Style="{StaticResource Label}"/>
        <TextBox      Name="Login"      FontSize="18"      Height="30"
HorizontalAlignment="Stretch"      Margin="0,10"      VerticalContentAlignment="Center"
FontFamily="Monotype Corsiva"/>

        <Label      Content="Пароль"      FontSize="20"      HorizontalAlignment="Left"
Style="{StaticResource Label}"/>
        <PasswordBox      Name="Password"      FontSize="18"      Height="30"
HorizontalAlignment="Stretch"      VerticalContentAlignment="Center"/>

        <Label      Content="ФИО"      FontSize="20"      HorizontalAlignment="Left"
Style="{StaticResource Label}"/>
        <TextBox      Name="FIObox"      FontSize="18"      Height="30"
HorizontalAlignment="Stretch"      VerticalContentAlignment="Center"
FontFamily="Monotype Corsiva"/>

        <Label      Content="Номер      телефона"      FontSize="20"
HorizontalAlignment="Left"      Style="{StaticResource Label}"/>
        <TextBox      Name="PhoneNumber"      FontSize="18"      Height="30"
HorizontalAlignment="Stretch"      VerticalContentAlignment="Center"
FontFamily="Monotype Corsiva"/>

        <Label      Content="Должность"      FontSize="20"      HorizontalAlignment="Left"
Style="{StaticResource Label}"/>
        <ComboBox      Name="Positions"      FontSize="18"      Height="30"
HorizontalAlignment="Stretch"      VerticalContentAlignment="Center"
FontFamily="Monotype Corsiva"/>

        <Button      x:Name="btnRegister"      Content="Войти"
HorizontalAlignment="Center"      Margin="0,30"      Height="30"      Width="120"
FontSize="18"      Style="{StaticResource Button}"      Click="btnRegister_Click"/>
    </StackPanel>
</Grid>
</Window>

```

Листинг 4 – Код логики работы окна авторизации

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

namespace TechnoService
{
    public partial class Auth : Window
    {
        private static readonly Regex LoginRegex = new Regex(@"^[a-zA-Z0-9]{8,}$");
        private static readonly Regex PasswordRegex = new Regex(@"^(?=.*\d)[a-zA-Z0-9]{8,}$");
        private bool isNavToReg = false;
        private readonly Entities db;
        public Auth()
        {
            InitializeComponent();
            db = new Entities();
        }

        private void Window_Closing(object sender,
            System.ComponentModel.CancelEventArgs e)
        {
            if (isNavToReg)
            {
                return;
            }

            var result = MessageBox.Show("Вы действительно хотите выйти из программы?", "Выход", MessageBoxButton.YesNo, MessageBoxImage.Question);

            if (result == MessageBoxResult.Yes)
            {
                e.Cancel = false;
            }
            else
            {
                e.Cancel = true;
            }
        }
    }
}
```

```

    }

    private void Hyperlink_Click(object sender, RoutedEventArgs e)
    {

        isNavToReg = true;
        Reg reg = new Reg();
        reg.Show();
        this.Close();
    }

    private void btnLogin_Click(object sender, RoutedEventArgs e)
    {
        string login = LoginTextBox.Text;
        string password = Hash_Password>PasswordBox.Password);

        try
        {
            if (!LoginRegex.IsMatch(login))
            {
                MessageBox.Show("Логин должен содержать только английские буквы и цифры, и быть длиной не менее 8 символов.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);
                return;
            }
            if (!PasswordRegex.IsMatch(password))
            {
                MessageBox.Show("Пароль должен содержать только английские буквы и цифры, включать минимум одну цифру, и быть длиной не менее 8 символов.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);
                return;
            }
            var user = db.Users.AsNoTracking().FirstOrDefault(u => u.Login == login && u.Password == password);

            if (user == null)
            {
                MessageBox.Show("Пользователь с такими данными не найден в системе!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);
                return;
            }

            MessageBox.Show("Вы авторизовались!");

```

```

        var employee = db.Employees.FirstOrDefault(u => u.UserID ==
user.ID);

        if (employee != null)
        {
            ((App)Application.Current).CurrentUserPositionID =
employee.PositionID;
        }
        isNavToReg = true;
        ((App)Application.Current).CurrentUserID = user.ID;
        MainWindow mainWindow = new MainWindow();
        mainWindow.Show();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

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

```

Листинг 5 – Код логики работы окна регистрации

```

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

namespace TechnoService
{
    public partial class Reg : Window

```

```

    {
        private static readonly Regex LoginRegex = new Regex(@"^[a-zA-Z0-9]{8,}$");
        private static readonly Regex PasswordRegex = new Regex(@"^(?=.*\d)[a-zA-Z0-9]{8,}$");
        private static readonly Regex FIOregex = new Regex(@"^[А-ЯЁ][а-яё]+$");
        private static readonly Regex PhoneNumberRegex = new Regex(@"^\+?[1-9]\d{9,14}$");

        private bool isNavToAuth = false;
        private readonly Entities db;
        public Reg()
        {
            InitializeComponent();
            db = new Entities();
            LoadPositions();
        }

        private void LoadPositions()
        {
            var positions = db.Positions.ToList();
            Positions.ItemsSource = positions;
            Positions.DisplayMemberPath = "Position1";
            Positions.SelectedValuePath = "ID";
        }

        private void btnRegister_Click(object sender, RoutedEventArgs e)
        {
            string login = Login.Text;
            string password = Hash_Password(Password.Password);
            string FIO = FIObox.Text;
            string phoneNumber = PhoneNumber.Text;

            try
            {
                if (!LoginRegex.IsMatch(login))
                {
                    MessageBox.Show("Логин должен содержать только английские буквы и цифры, и быть длиной не менее 8 символов.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);
                    return;
                }

                if (db.Users.Any(u => u.Login == login))

```

```

        {
            MessageBox.Show("Пользователь с таким логином уже
зарегистрирован в системе!", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Warning);
            Login.Text = "";
            return;
        }

        if (!PasswordRegex.IsMatch(password))
        {
            MessageBox.Show("Пароль должен содержать только английские
буквы и цифры, включать минимум одну цифру, и быть длиной не менее 8 символов.",
"Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);
            return;
        }

        if (!FIOcheck(FIO))
        {
            MessageBox.Show("ФИО введено некорректно.", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Warning);
            return;
        }

        if (db.Employees.Any(u => u.Phone == phoneNumber))
        {
            MessageBox.Show("Этот номер телефона уже зарегистрирован в
системе.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Warning);
            return;
        }

        if (!PhoneNumberRegex.IsMatch(phoneNumber))
        {
            MessageBox.Show("Введите корректный международный номер
телефона (от 10 до 15 символов, допускается знак '+').", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Warning);
            return;
        }

        if (Positions.SelectedItem == null)
        {
            MessageBox.Show("Выберите должность.", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Warning);
            return;
        }
    }

```

```

        var selectedPositionID = (int)Positions.SelectedValue;

        var newUser = new User
        {
            Login = login,
            Password = password
        };

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

        var splitFIO = FIO.Split(' ');

        var newEmployee = new Employee
        {
            UserID = newUser.ID,
            PositionID = selectedPositionID,
            FirstName = splitFIO[0],
            LastName = splitFIO[1],
            SecondName = splitFIO.Length > 2 ? splitFIO[2] :
string.Empty,
            Phone = phoneNumber
        };
        db.Employees.Add(newEmployee);
        db.SaveChanges();

        MessageBox.Show("Регистрация прошла успешно!");
        isNavToAuth = true;
        Auth auth = new Auth();
        auth.Show();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Window_Closing(object sender,
System.ComponentModel.CancelEventArgs e)
{
    if (isNavToAuth)
    {
        return;
    }
}

```



```

    }

    var result = MessageBox.Show("Вы действительно хотите выйти из
программы?", "Выход", MessageBoxButton.YesNo, MessageBoxImage.Question);
    if (result == DialogResult.Yes)
    {
        e.Cancel = false;
    }
    else
    {
        e.Cancel = true;
    }
}

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

private bool FIOcheck(string FIO)
{
    var splitFIO = FIO.Split(new[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);

    if (splitFIO.Length < 2 || splitFIO.Length > 3)
    {
        return false;
    }

    foreach (var part in splitFIO)
    {
        if (string.IsNullOrEmpty(part) || !FIOregex.IsMatch(part)
|| part.Length > 50)
        {
            return false;
        }
    }
    return true;
}
}
}

```

Имеются все проверки по заполнению данных пользователем, также была реализован небольшой модуль безопасности, где пароль шифруется алгоритмом SHA-1.

После того, как сотрудник авторизовался, его встречает следующее окно, где у него доступно меню. В этом меню хранятся компоненты программы. Пользователь может посмотреть список сотрудников, редактировать, удалять их.

В клиентах, у нас выводится список клиентов, где пользователь может удалять, добавлять, редактировать информацию о сотрудниках. В технике функционал аналогичный, но также есть переход на страницу «Тип техники», где можно добавить или удалить тип техники.

Тоже самое, пользователь может делать с заявками, просматривать логи, и формировать отчеты.

Важно отметить, если у пользователя имеется роль «Менеджер», то в меню будут доступны компоненты: клиенты, заявки, логи, отзыв.

Компонент отзыв в себе содержит QR-код, который перенаправляет на платформу «Google Forms», где пользователь может оставить свой отзыв о сервисе.

Для того, чтобы мы могли загрузить данные с базы данных, мы воспользуемся фреймворком «Entity Framework». И подключим нашу базу данных к приложению. По результату подключения, мы увидим в «Обозревателе решений» то, что у нас появился объект «model.edmx», что говорит о том, что база данных была подключена. После мы пропишем в нашу модель экземпляра, создание экземпляра в случае, если он равен null. На рисунке 15 представлен пример подключения базы данных, а также в листинге 6 код для создания экземпляра.

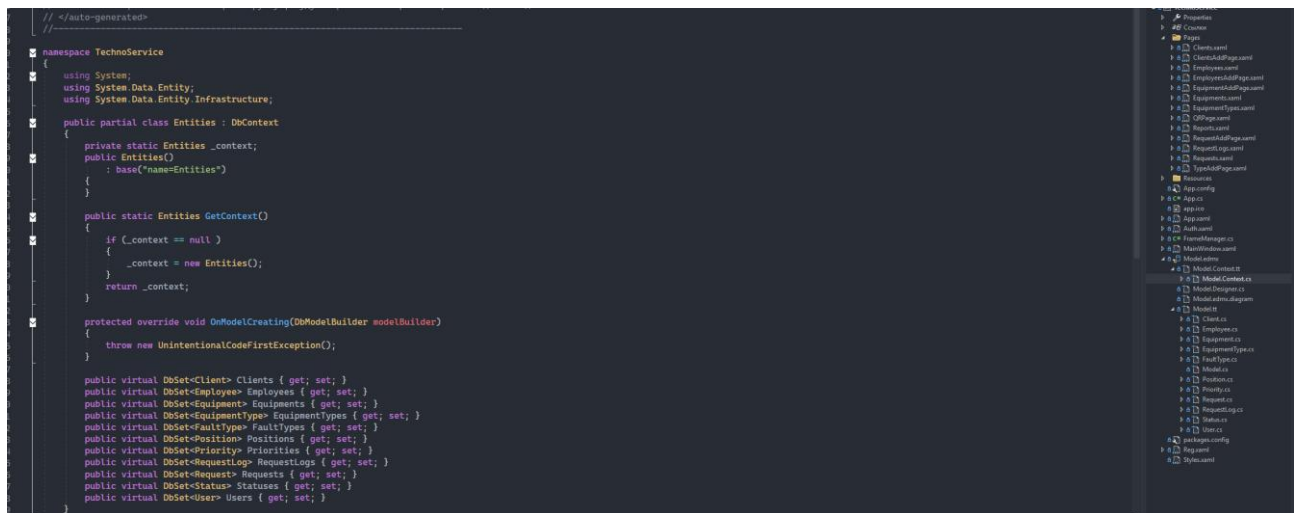


Рисунок 15 – Результат подключения базы данных

Листинг 6 – Код для создания экземпляра БД.

```
//-----
--
// <auto-generated>
//     Этот код создан по шаблону.
//
//     Изменения, вносимые в этот файл вручную, могут привести к непредвиденной
//     работе приложения.
//     Изменения, вносимые в этот файл вручную, будут перезаписаны при повторном
//     создании кода.
// </auto-generated>
//-----
--

namespace TechnoService
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;

    public partial class Entities : DbContext
    {
        private static Entities _context;
        public Entities()
            : base("name=Entities")
        {
        }

        public static Entities GetContext()
        {
            if (_context == null )

```

```

        {
            _context = new Entities();
        }
        return _context;
    }

    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        throw new UnintentionalCodeFirstException();
    }

    public virtual DbSet<Client> Clients { get; set; }
    public virtual DbSet<Employee> Employees { get; set; }
    public virtual DbSet<Equipment> Equipments { get; set; }
    public virtual DbSet<EquipmentType> EquipmentTypes { get; set; }
    public virtual DbSet<FaultType> FaultTypes { get; set; }
    public virtual DbSet<Position> Positions { get; set; }
    public virtual DbSet<Priority> Priorities { get; set; }
    public virtual DbSet<RequestLog> RequestLogs { get; set; }
    public virtual DbSet<Request> Requests { get; set; }
    public virtual DbSet<Status> Statuses { get; set; }
    public virtual DbSet<User> Users { get; set; }
}
}

```

Теперь перейдем к самим страницам, как они были реализованы. Страница «Клиенты». На рисунке 16 представлена страница «Клиенты». На рисунке 17 представлена страница для добавления клиента. На рисунке 18 представлен пример, как работает удаление пользователя. В листинге 7 представлен html код страницы, в листинге 8 код логики работы данной страницы, в листинге 9 html код для добавление клиента, а в листинге 10 код логики работы страницы добавления.

TechnoService - Клиенты

Имя	Фамилия	Отчество	Номер телефона	Электронная почта	
Алексей	Петров		89001234568	petrov@example.com	Редактировать
Мария	Сидорова		89001234569	sidorova@example.com	Редактировать
Ольга	Кузнецова		89001234570	kuznetsova@example.com	Редактировать
Петров	Петр	Петрович	+78621231212	petrovich@yandex.ru	Редактировать
Иванов	Иван	Иванович	+79299172902	ivanov@gmail.com	Редактировать

Добавить

Удалить

Сотрудники

Клиенты

Техника

Заявки

Логи

Отчеты

Рисунок 16 – Страница «Клиенты»

Клиент

ФИО

Номер телефона

Электронная почта

Рисунок 17 – Добавление клиента

TechnoService - Клиент

Клиент

ФИО

Алексей Петров

Номер телефона

89001234568

Электронная почта

petrov@example.com

Сохранить

Назад

Сотрудники

Клиенты

Штатная

Заявки

Логи

Отчеты

Рисунок 17.1 – Страница добавления клиента, которая работает как и редактирование данных о клиенте

TechnoService - Клиенты

Имя	Фамилия	Отчество	Номер телефона	Электронная почта	
Алексей	Петров		89001234568	petrov@example.com	Редактировать
Мария	Сидорова		89001234569	sidorova@example.com	Редактировать
Ольга	Кузнецова		89001234570	kuznetsova@example.com	Редактировать
Петров	Петр	Петрович	+78621231212	petrovich@yandex.ru	Редактировать
Иванов	Иван	Иванович	+79299172902	divanov@gmail.com	Редактировать

Предупреждение

Вы действительно хотите удалить клиента с базы данных?

Да

Нет

Добавить

Удалить

Сотрудники

Клиенты

Штатная

Заявки

Логи

Отчеты

Рисунок 18 – Всплывающее окно с удалением клиентов из системы

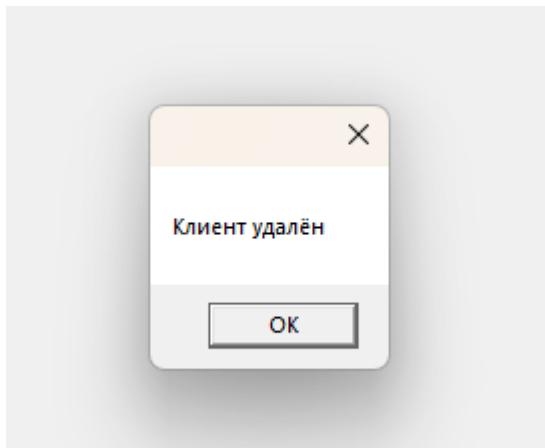


Рисунок 18.1 – Успешное удаление клиента из системы

Листинг 7 – код xaml страницы с клиентами

```
<Page x:Class="TechnoService.Pages.Clients"
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:TechnoService.Pages"
mc:Ignorable="d"
d:DesignHeight="720" d:DesignWidth="1130"
Title="Клиенты">

<Grid Background="White">
<Grid.RowDefinitions>
<RowDefinition Height="377*"/>
<RowDefinition Height="73*"/>
</Grid.RowDefinitions>
<DataGrid Grid.Row="0" x:Name="DataGridClients" AutoGenerateColumns="False"
IsReadOnly="True" FontFamily="Monotype Corsiva" FontSize="14">
<DataGrid.Columns>
<DataGridTextColumn Header="Имя" Width="*" Binding="{Binding FirstName}"/>
<DataGridTextColumn Header="Фамилия" Width="*" Binding="{Binding LastName}"/>
<DataGridTextColumn Header="Отчество" Width="*" Binding="{Binding
SecondName}"/>
<DataGridTextColumn Header="Номер телефона" Width="*" Binding="{Binding
Phone}"/>
<DataGridTextColumn Header="Электронная почта" Width="*" Binding="{Binding
Email}"/>

<DataGridTemplateColumn Width="100">
<DataGridTemplateColumn.CellTemplate>
<DataTemplate>
```

```

<Button x:Name="BtnEdit" Content="Редактировать" ToolTip="Нажав на кнопку, Вы
можете редактировать запись" Width="100" FontSize="14" Click="BtnEdit_Click"
Style="{StaticResource Button}"/>
</DataTemplate>
</DataGridTemplateColumn.CellTemplate>
</DataGridTemplateColumn>
</DataGrid.Columns>
</DataGrid>
<Button Grid.Row="1" x:Name="BtnAdd" Content="Добавить"
HorizontalAlignment="Left" Margin="15" Height="40" ToolTip="Нажав на кнопку,
Вы можете добавить запись." VerticalAlignment="Center" Click="BtnAdd_Click"
Style="{StaticResource Button}"/>
<Button Grid.Row="1" x:Name="BtnDelete" Content="Удалить"
HorizontalAlignment="Right" Margin="15" Height="40" ToolTip="Нажав на кнопку,
Вы можете удалить запись." VerticalAlignment="Center" Click="BtnDelete_Click"
Style="{StaticResource Button}"/>
</Grid>
</Page>

```

Листинг 8 – код логики работы страницы

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.CompilerServices;
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 TechnoService.Pages
{
    public partial class Clients : Page
    {
        private readonly Entities db;

        public Clients()

```



```

    {
        InitializeComponent();
        db = new Entities();
        DataGridClients.ItemsSource = db.Clients.ToList();
    }

    private void BtnEdit_Click(object sender, RoutedEventArgs e)
    {
        NavigationService.Navigate(new Pages.ClientsAddPage((sender as
Button).DataContext as Client));
    }

    private void BtnDelete_Click(object sender, RoutedEventArgs e)
    {
        if (DataGridClients.SelectedItem != null)
        {
            Client selectedClient = DataGridClients.SelectedItem as
Client;

            var result = MessageBox.Show("Вы действительно хотите удалить
клиента с базы данных?", "Предупреждение", MessageBoxButton.YesNo,
MessageBoxImage.Question);

            if (result == MessageBoxResult.Yes)
            {
                try
                {
                    db.Clients.Attach(selectedClient);
                    db.Clients.Remove(selectedClient);

                    db.SaveChanges();
                    MessageBox.Show("Клиент удалён");
                    DataGridClients.ItemsSource = db.Clients.ToList();
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Ошибка при удалении клиента.\n" +
"Скорее всего, данный клиент где-то
используется.", ex.Message);
                }
            }
            else
            {
                MessageBox.Show("Клиент не был удален из базы данных");
            }
        }
    }
}

```

```

        else
        {
            MessageBox.Show("Выберите запись для удаления", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }

    private void BtnAdd_Click(object sender, RoutedEventArgs e)
    {
        NavigationService.Navigate(new ClientsAddPage(null));
    }
}
}

```

Листинг 9 – код xaml страницы добавления клиента

```

<Page x:Class="TechnoService.Pages.ClientsAddPage"
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:TechnoService.Pages"
mc:Ignorable="d"
d:DesignHeight="450" d:DesignWidth="800"
Title="Клиент" Background="White">

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

        <StackPanel VerticalAlignment="Center" HorizontalAlignment="Center"
Grid.Row="1" Width="400">
            <TextBlock Text="Клиент" FontSize="26" FontWeight="Bold"
TextAlignment="Center" Margin="0,0,0,20" Style="{StaticResource TextBlock}"/>

            <TextBlock Text="ФИО" FontSize="16" Style="{StaticResource
TextBlock}"/>

            <TextBox Name="FullNameTextBox" Width="400" Height="40"
FontSize="16" VerticalContentAlignment="Center" Margin="0 0 0 10"
FontFamily="Monotype Corsiva"/>

```

```

        <TextBlock      Text="Homep      телефона"      FontSize="16"
Style="{StaticResource TextBlock}"/>

        <TextBox      Name="PhoneNumberTextBox"      Width="400"      Height="40"
FontSize="16"      VerticalContentAlignment="Center"      Margin="0      0      0      10"
FontFamily="Monotype Corsiva"/>

        <TextBlock      Text="Электронная      почта"      FontSize="16"
Style="{StaticResource TextBlock}"/>

        <TextBox Name="EmailTextBox" Width="400" Height="40" FontSize="16"
VerticalContentAlignment="Center" FontFamily="Monotype Corsiva" />
    </StackPanel>

    <StackPanel Orientation="Horizontal" HorizontalAlignment="Center"
Grid.Row="2" Margin="0,30,0,0">

        <Button x:Name="SaveButton" Content="Сохранить" Width="120"
Height="30" FontSize="16" Click="SaveButton_Click" Style="{StaticResource
Button}" Margin="0 0 10 10"/>

        <Button x:Name="GoBackButton" Content="Назад" Width="120"
Height="30" FontSize="16" Click="GoBackButton_Click" Style="{StaticResource
Button}" Margin="0 0 10 10"/>

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

```

Листинг 10 – Код логики работы страницы добавления клиента

```

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

namespace TechnoService.Pages
{
    public partial class ClientsAddPage : Page
    {
        private static readonly Regex FIOregex = new Regex(@"^[А-ЯЁ] [а-
яё]+$");
        private static readonly Regex PhoneNumberRegex = new Regex(@"^\+?[1-
9]\d{9,14}$");

        private readonly Entities db;
    }
}

```

```

private readonly Client currentClient;

public ClientsAddPage(Client selectedClient = null)
{
    InitializeComponent();
    db = new Entities();

    currentClient = selectedClient;

    if (currentClient != null)
    {
        FullNameTextBox.Text = $"{currentClient.FirstName}
{currentClient.LastName} {currentClient.SecondName}".Trim();
        PhoneNumberTextBox.Text = currentClient.Phone;
        EmailTextBox.Text = currentClient.Email;
    }
}

private void GoBackButton_Click(object sender, RoutedEventArgs e)
{
    FrameManager.MainFrame.Navigate(new Uri("Pages/Clients.xaml",
UriKind.Relative));
}

private void SaveButton_Click(object sender, RoutedEventArgs e)
{
    string FIO = FullNameTextBox.Text.Trim();
    string Phone = PhoneNumberTextBox.Text.Trim();
    string Email = EmailTextBox.Text.Trim();

    try
    {
        if (!FIOcheck(FIO))
        {
            MessageBox.Show("ФИО введено некорректно. Убедитесь, что
указаны от 2 до 3 слов с заглавной буквы.", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Information);
            return;
        }
        if (!PhoneNumberRegex.IsMatch(Phone))
        {

```

```

        MessageBox.Show("Введите корректный международный номер
        телефона (от 10 до 15 символов, допускается знак '+').", "Ошибка",
        MessageBoxButton.OK, MessageBoxImage.Warning);
        return;
    }

    var splitFIO = FIO.Split(' ');

    if (currentClient == null)
    {

        var newClient = new Client
        {
            FirstName = splitFIO[0],
            LastName = splitFIO.Length > 1 ? splitFIO[1] :
string.Empty,
            SecondName = splitFIO.Length > 2 ? splitFIO[2] :
string.Empty,
            Phone = Phone,
            Email = Email
        };

        db.Clients.Add(newClient);
        db.SaveChanges();
        MessageBox.Show("Новый клиент успешно добавлен.", "Успех",
        MessageBoxButton.OK, MessageBoxImage.Information);
    }
    else
    {
        var clientToUpdate = db.Clients.SingleOrDefault(c => c.ID
== currentClient.ID);

        if (clientToUpdate != null)
        {

            bool isChanged = clientToUpdate.FirstName !=
splitFIO[0] ||
                                clientToUpdate.LastName !=
(splitFIO.Length > 1 ? splitFIO[1] : string.Empty) ||
                                clientToUpdate.SecondName !=
(splitFIO.Length > 2 ? splitFIO[2] : string.Empty) ||
                                clientToUpdate.Phone != Phone ||
                                clientToUpdate.Email != Email;

            if (!isChanged)

```

```

        {
            MessageBox.Show("Вы не внесли никаких изменений.
Пожалуйста, измените хотя бы одно поле.", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Warning);

            return;
        }

        clientToUpdate.FirstName = splitFIO[0];
        clientToUpdate.LastName = splitFIO.Length > 1 ?
splitFIO[1] : string.Empty;
        clientToUpdate.SecondName = splitFIO.Length > 2 ?
splitFIO[2] : string.Empty;
        clientToUpdate.Phone = Phone;
        clientToUpdate.Email = Email;

        db.SaveChanges();

        MessageBox.Show("Данные клиента успешно обновлены.",
"Успех", MessageBoxButton.OK, MessageBoxImage.Information);
    }
    else
    {
        MessageBox.Show("Клиент не найден в базе данных.
Попробуйте заново.", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
    }
}

FullNameTextBox.Text = string.Empty;
PhoneNumberTextBox.Text = string.Empty;
EmailTextBox.Text = string.Empty;
FrameManager.MainFrame.Navigate(new Uri("/Pages/Clients.xaml",
UriKind.Relative));
}
catch (Exception ex)
{
    MessageBox.Show($"Произошла ошибка: {ex.Message}", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Error);
}
}

private bool FIOcheck(string FIO)
{
    var splitFIO = FIO.Split(new[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);

```

```
        if (splitFIO.Length < 2 || splitFIO.Length > 3)
        {
            return false;
        }

        foreach (var part in splitFIO)
        {
            if (string.IsNullOrEmpty(part) || !FIOregex.IsMatch(part)
            || part.Length > 50)
            {
                return false;
            }
        }
        return true;
    }
}
```

Другие страницы реализованы аналогичным образом, с соблюдением общей структуры и подходов к разработке интерфейсов. Однако, каждая страница имеет свои уникальные особенности, связанные с отображением и обработкой данных конкретной сущности из базы данных. Эти различия проявляются в добавлении специфических элементов управления на страницы, которые необходимы для работы с соответствующими полями и атрибутами базы данных.

Привязка данных осуществляется непосредственно в XAML-коде страниц. Этот процесс включает использование механизмов привязки данных WPF, таких как Binding, для связывания интерфейсных элементов (например, текстовых полей, списков, комбинированных элементов) с соответствующими свойствами или столбцами базы данных. Таким образом, XAML-код каждой страницы настраивается таким образом, чтобы отображать данные, относящиеся к конкретной сущности, а также предоставлять пользователю возможность взаимодействия с этими данными.

Кроме того, различия касаются и логики обработки данных, которая реализована в code-behind файлах. Эти файлы содержат обработчики событий и методы, обеспечивающие связь между пользовательским интерфейсом и бизнес-

логикой. Например, в них реализуются методы для загрузки данных из базы, сохранения изменений, фильтрации или выполнения других операций, специфичных для конкретной сущности. Для каждой страницы такие таблицы и методы в коде могут существенно отличаться в зависимости от особенностей данных и функциональных требований.

Стоит рассмотреть страницу для отчетов, так как она максимально отличается от других страниц. На странице отчетов, есть отчеты по заявкам, по среднему времени выполнения заявок, и по типам неисправности.

На рисунках 19-21 представлены отчеты. Код `haml` страницы отчетов представлен в листине 11, а его `code-behind` в листинге 12.

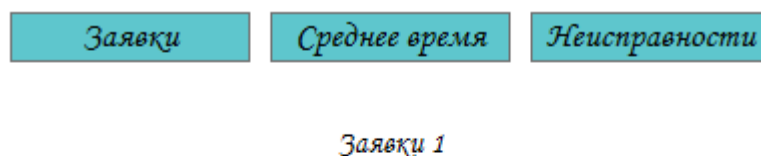


Рисунок 19 – Отчет по заявкам

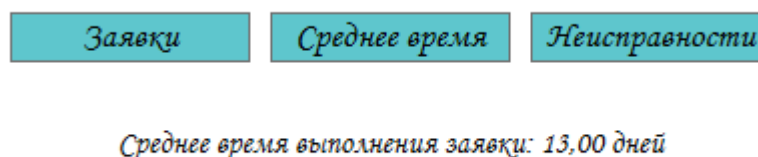


Рисунок 20 – Отчет по среднему времени выполнения заявки

Аппаратная ошибка: 0

Программная ошибка: 0

Сетевая ошибка: 1

Ошибка питания: 0

Механическая ошибка: 2

Рисунок 21 – Отчет по типам неисправностей техники

Листинг 11 – Код xaml страницы отчета

```
<Page x:Class="TechnoService.Pages.Reports"
      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:TechnoService.Pages"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="Отчеты">

    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="10*"/>
            <RowDefinition Height="60*"/>
            <RowDefinition Height="10*"/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="*"/>
        </Grid.ColumnDefinitions>
```

```

        <StackPanel VerticalAlignment="Center" HorizontalAlignment="Center"
Orientation="Horizontal" Grid.ColumnSpan="2">
            <Button x:Name="Quantity" Content="Заявки" Style="{StaticResource
Button}" Click="Quantity_Click" Margin="0 0 10 0"/>
            <Button x:Name="AVGTime" Content="Среднее время"
Style="{StaticResource Button}" Click="AVGTime_Click" Margin="0 0 10 0"/>
            <Button x:Name="Faults" Content="Неисправности"
Style="{StaticResource Button}" Click="Faults_Click"/>
        </StackPanel>
        <Label x:Name="QuantityLabel" Style="{StaticResource Label}"
Grid.Row="1" Grid.ColumnSpan="2" HorizontalAlignment="Center"/>
        <Label x:Name="AVGTimeLabel" Style="{StaticResource Label}"
Grid.Row="1" Grid.ColumnSpan="2" HorizontalAlignment="Center"/>

        <StackPanel x:Name="Statistic" VerticalAlignment="Center"
HorizontalAlignment="Center" Grid.Row="1" Grid.ColumnSpan="2"/>
    </Grid>
</Page>

```

Литсинг 12 – Code-behind

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;

namespace TechnoService.Pages
{
    public partial class Reports : Page
    {
        private readonly Entities db;
        private string QCP;
        private int FaultsCount;
        private int RequestCount;

        public Reports()
        {
            InitializeComponent();
            db = new Entities();

```

```

        QCP = Quantity.Content.ToString();
    }

    private void Quantity_Click(object sender, RoutedEventArgs e)
    {
        PreparePage();
        Quantity.Visibility = Visibility.Visible;
        QuantityLabel.Content = QCP + " " + db.Requests.Count(x =>
x.Status == 3).ToString();
    }

    private void AVGTime_Click(object sender, RoutedEventArgs e)
    {
        PreparePage();
        AVGTime.Visibility = Visibility.Visible;

        var requests = db.Requests.ToList();

        int totalDays = 0;
        int completedRequests = 0;

        foreach (var request in requests)
        {
            DateTime startDate = request.RequestDate;
            DateTime endDate;

            if (request.Status == 3) // Статус "выполнено"
            {
                endDate = db.RequestLogs
                    .Where(log => log.RequestID == request.ID &&
log.Description.Contains("выполнено"))
                    .OrderByDescending(log => log.LogDate)
                    .Select(log => log.LogDate)
                    .FirstOrDefault();
            }
            else
            {
                endDate = DateTime.Now;
            }

            if (endDate != DateTime.MinValue)
            {
                totalDays += (endDate - startDate).Days;
                completedRequests++;
            }
        }
    }

```

```

    }

    float averageTime = completedRequests > 0 ? (float)totalDays /
completedRequests : 0;
    AVGTimeLabel.Content = $"Среднее время выполнения заявки:
{averageTime:F2} дней";
    }

    private void Faults_Click(object sender, RoutedEventArgs e)
    {
        PreparePage();

        FaultsCount = db.FaultTypes.Count();

        for (int i = 1; i <= FaultsCount; i++)
        {
            FillingContent(i.ToString());
        }
    }

    private void FillingContent(string labelName)
    {
        int id = int.Parse(labelName);
        string FaultTypeName = db.FaultTypes.Where(x => x.ID ==
id).Select(u => u.Type).FirstOrDefault();
        string FaultTypeCount = db.Requests.Count(x => x.FaultType ==
id).ToString();

        Label label = new Label()
        {
            Name = "FaultTypeQuantity" + labelName,
            Content = $"{FaultTypeName}: {FaultTypeCount}",
            FontSize = 14,
            HorizontalAlignment = HorizontalAlignment.Left,
            VerticalAlignment = VerticalAlignment.Top,
            Margin = new Thickness(10)
        };

        Statistic.Children.Add(label);
    }

    private void PreparePage()
    {
        // Очищаем содержимое страницы
        Statistic.Children.Clear();
    }

```

```

        // Делаем кнопки видимыми
        Quantity.Visibility = Visibility.Visible;
        AVGTime.Visibility = Visibility.Visible;
        Faults.Visibility = Visibility.Visible;

        // Очищаем все текстовые поля, если это необходимо
        QuantityLabel.Content = string.Empty;
        AVGTimeLabel.Content = string.Empty;
    }
}
}

```

Дальше рассмотрим роль менеджера, так как из-за роли отличается меню программы. Для того, чтобы реализовать данное изменение под роль, был реализован класс, который хранит в себе текущий идентификатор пользователя. Если обратиться к базе данных, то там мы увидим, что есть связь между пользователями и ролями. И таким образом мы храним информацию о текущем пользователе и его роли. Код представлен в листинге 13.

Листинг 13 – Класс, который хранит в себе текущие данные пользователя

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;

namespace TechnoService
{
    public partial class App : Application
    {
        public int CurrentUserID { get; set; }
        public int CurrentUserPositionID { get; set; }

        public void SetCurrentUserID(int UserID)
        {
            CurrentUserID = UserID;
        }
    }
}

```

После того, как класс был реализован, нам надо сделать инициализацию идентификатора в главном окне приложения. Ниже представлен xaml код главного окна приложения в листинге 14, а также его код логики работы в листинге 15.

Листинг 14 – Код xaml главного окна

```
<Window x:Class="TechnoService.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:TechnoService"
        mc:Ignorable="d"
        Title="TechoService" Height="720" Width="1130"
        WindowStartupLocation="CenterScreen" ResizeMode="NoResize"
        Closing="Window_Closing">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="50" />
        </Grid.RowDefinitions>

        <Border BorderThickness="0 1 0 0" BorderBrush="#d9d9d9" Grid.Row="0">
            <Frame x:Name="MainFrame" NavigationUIVisibility="Hidden"
                Navigated="MainFrame_Navigated" />
        </Border>

        <Border Grid.Row="1" BorderThickness="0 1 0 0" BorderBrush="#d9d9d9">
            <StackPanel VerticalAlignment="Center" Orientation="Horizontal"
                HorizontalAlignment="Center">
                <Button x:Name="Employees" Content="Сотрудники" Width="100"
                    Height="25" Style="{StaticResource Button}" Click="Employees_Click" Margin="0
                    0 10 0" />
                <Button x:Name="Clients" Content="Клиенты" Width="100"
                    Height="25" Style="{StaticResource Button}" Click="Clients_Click" Margin="0 0
                    10 0" />
                <Button x:Name="Equipment" Content="Техника" Width="100"
                    Height="25" Style="{StaticResource Button}" Click="Equipment_Click" Margin="0
                    0 10 0" />
                <Button x:Name="Requests" Content="Заявки" Width="100"
                    Height="25" Style="{StaticResource Button}" Click="Requests_Click" Margin="0 0
                    10 0" />
            </StackPanel>
        </Border>
    </Grid>
</Window>
```

```

        <Button x:Name="Logs" Content="Логги" Width="100" Height="25"
Style="{StaticResource Button}" Click="Logs_Click" Margin="0 0 10 0"/>
        <Button x:Name="Reports" Content="Отчеты" Width="100"
Height="25" Style="{StaticResource Button}" Click="Reports_Click" Margin="0 0
0 0"/>
        <Button x:Name="Feedback" Content="Отзыв" Width="100"
Height="25" Style="{StaticResource Button}" Click="Feedback_Click" Margin="0
0 0 0"/>

    </StackPanel>
</Border>
</Grid>
</Window>

```

Листинг 15 – Код логики работы главной страницы

```

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

namespace TechnoService
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            int UserPositionID =
((App)Application.Current).CurrentUserPositionID;

            if (UserPositionID == 1)
            {
                Employees.Visibility = Visibility.Collapsed;
                Reports.Visibility = Visibility.Collapsed;
                Equipment.Visibility = Visibility.Collapsed;
            }
            else
            {

```

```
        Feedback.Visibility = Visibility.Collapsed;
    }

    FrameManager.MainFrame = MainFrame;
}

private void MainFrame_Navigated(object sender, NavigationEventArgs e)
{
    if (!(e.Content is Page page)) return;
    this.Title = $"TechnoService - {page.Title}";
}

private void Employees_Click(object sender, RoutedEventArgs e)
{
    FrameManager.MainFrame.Navigate(new Uri("Pages/Employees.xaml",
UriKind.Relative));
}

private void Clients_Click(object sender, RoutedEventArgs e)
{
    FrameManager.MainFrame.Navigate(new Uri("Pages/Clients.xaml",
UriKind.Relative));
}

private void Equipment_Click(object sender, RoutedEventArgs e)
{
    FrameManager.MainFrame.Navigate(new Uri("Pages/Equipments.xaml",
UriKind.Relative));
}

private void Requests_Click(object sender, RoutedEventArgs e)
{
    FrameManager.MainFrame.Navigate(new Uri("Pages/Requests.xaml",
UriKind.Relative));
}

private void Logs_Click(object sender, RoutedEventArgs e)
{
    FrameManager.MainFrame.Navigate(new Uri("Pages/RequestLogs.xaml",
UriKind.Relative));
}

private void Reports_Click(object sender, RoutedEventArgs e)
{

```



```
        FrameManager.MainFrame.Navigate(new Uri("Pages/Reports.xaml",
UriKind.Relative));
    }

    private void Feedback_Click(object sender, RoutedEventArgs e)
    {
        FrameManager.MainFrame.Navigate(new Uri("Pages/QRPage.xaml",
UriKind.Relative));
    }

    private void Window_Closing(object sender,
System.ComponentModel.CancelEventArgs e)
    {
        var result = MessageBox.Show("Вы действительно хотите выйти?",
"Выход", MessageBoxButton.YesNo, MessageBoxImage.Question);

        if (result == MessageBoxResult.Yes)
        {
            e.Cancel = false;
        }
        else
        {
            e.Cancel = true;
        }
    }
}
```

После того, как мы зашли под ролью «Менеджер», мы можем рассмотреть последнюю страницу в приложении, где выводится QR-код, который перенаправляет нас на платформу «Google Forms». Стоит отметить, что была использована внешняя библиотека для генерации QR-кода. Библиотека носит название «QRcoder». На рисунке 22 представлена страница с кодом.



Оставьте свой отзыв о нас

Рисунок 22 – страница с QR-кодом.

Ниже представлен xaml код страницы в листинге 16. А его код-бехинд, представлен в листине 17

Листинг 16 – xaml код страницы с QR-кодом

```
<Page x:Class="TechnoService.Pages.QRPage"
      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:TechnoService.Pages"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="Отзыв">

    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="5*"/>
        </Grid.RowDefinitions>
    </Grid>
```

```

        <RowDefinition Height="90*" />
        <RowDefinition Height="5*" />
    </Grid.RowDefinitions>

    <StackPanel
        Grid.Row="1"
        HorizontalAlignment="Center"
        VerticalAlignment="Center">
        <Image x:Name="QRCodeImage" Grid.Row="1" Stretch="None" />
        <Separator Opacity="0" Height="15" />
        <Label Content="Оставьте свой отзыв о нас" Style="{StaticResource
Label}" Margin="0,0,0,0" HorizontalAlignment="Center" />
    </StackPanel>
</Grid>
</Page>

```

Листинг 17 – code-behind страницы с QR-кодом

```

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;
using QRCode;
using System.IO;
using System.Windows.Media.Animation;

namespace TechnoService.Pages
{
    /// <summary>
    /// Логика взаимодействия для QRPage.xaml
    /// </summary>
    public partial class QRPage : Page
    {
        public QRPage()
        {

```

```

        InitializeComponent();
        GenerateQR();
    }

    private void GenerateQR()
    {
        string url = "https://forms.gle/EvKo5TpFNVowjHGa9";

        if (string.IsNullOrEmpty(url))
        {
            MessageBox.Show("Введите текст для генерации QR-кода",
"Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
            return;
        }

        try
        {
            // Генерация QR-кода
            QRCodeGenerator qrGenerator = new QRCodeGenerator();
            QRCodeData qrCodeData = qrGenerator.CreateQrCode(url,
QRCodeGenerator.ECCLLevel.Q);
            QRCode qrCode = new QRCode(qrCodeData);

            using (var qrCodeBitmap = qrCode.GetGraphic(10))
            using (var memory = new MemoryStream())
            {
                qrCodeBitmap.Save(memory,
System.Drawing.Imaging.ImageFormat.Png);
                memory.Position = 0;

                BitmapImage bitmapImage = new BitmapImage();
                bitmapImage.BeginInit();
                bitmapImage.StreamSource = memory;
                bitmapImage.CacheOption = BitmapCacheOption.OnLoad;
                bitmapImage.EndInit();

                // Отображение QR-кода
                QRCodeImage.Source = bitmapImage;
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Ошибка при генерации QR-кода: {ex.Message}",
"Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }

```

```
}  
}  
}
```