СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc100860974)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 7](#_Toc100860975)

[1.1. Обоснование актуальности задачи 7](#_Toc100860976)

[1.1.1. Актуальность темы 7](#_Toc100860977)

[1.1.2. Анализ существующих разработок по данной теме 7](#_Toc100860978)

[1.2. Обоснование выбора и описание методологии разработки программы решения задачи 8](#_Toc100860979)

[1.3. Обоснование выбора и описание инструментальных средств разработки программы решения задачи 9](#_Toc100860980)

[1.4. Содержательная постановка задачи. 11](#_Toc100860981)

[1.4.1. Описательная модель задачи 11](#_Toc100860982)

[1.4.2. Математическая модель 11](#_Toc100860983)

[1.4.3. Описание входной информации 11](#_Toc100860984)

[1.4.4. Выходная информация 11](#_Toc100860985)

[1.4.5. Описание бизнес-процессов предприятия 11](#_Toc100860986)

[1.4.6. Описание потоков данных предприятия 11](#_Toc100860987)

[1.4.7. Диаграмма вариантов использования 12](#_Toc100860988)

[1.4.8. Диаграмма последовательности 12](#_Toc100860989)

[1.4.9. Диаграмма деятельности 12](#_Toc100860990)

[1.4.10. Диаграмма развертывания 12](#_Toc100860991)

[1.4.11. Архитектура приложения 12](#_Toc100860992)

[2. ПРОЕКТНАЯ ЧАСТЬ 12](#_Toc100860993)

[2.1. Разработка структуры данных программы. 12](#_Toc100860994)

[2.2. Логическая модель программы. 18](#_Toc100860995)

[2.3. Разработка пользовательского интерфейса программы 18](#_Toc100860996)

[2.3.1. Разработка требований к интерфейсу 18](#_Toc100860997)

[2.3.2. Проектирование и описание элементов управления 19](#_Toc100860998)

[2.4. Листинг программы 39](#_Toc100860999)

[2.5.Разработка тестовых данных 58](#_Toc100861000)

[2.6. Результат работы программы 65](#_Toc100861001)

[2.7. Руководство по использования программы 67](#_Toc100861002)

[Руководство программиста 67](#_Toc100861003)

[Руководство оператора 68](#_Toc100861004)

[2.8. Разработка политики безопасности 73](#_Toc100861005)

[2.9. Эксплуатация и развитие программы. 73](#_Toc100861006)

[ЗАКЛЮЧЕНИЕ 75](#_Toc100861007)

[СПИСОК ЛИТЕРАТУРЫ 77](#_Toc100861008)

[ПРИЛОЖЕНИЕ 1 – IDEF0 78](#_Toc100861009)

[ПРИЛОЖЕНИЕ 2 – DFD 80](#_Toc100861010)

[ПРИЛОЖЕНИЕ 3 – Class Diagram 82](#_Toc100861011)

[ПРИЛОЖЕНИЕ 4 – ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ 83](#_Toc100861012)

[ПРИЛОЖЕНИЕ 5 – ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ 84](#_Toc100861013)

[ПРИЛОЖЕНИЕ 6 – ДИАГРАММА ДЕЯТЕЛЬНОСТИ 85](#_Toc100861014)

[ПРИЛОЖЕНИЕ 7 – ДИАГРАММА РАЗВЕРТЫВАНИЯ 87](#_Toc100861015)

[ПРИЛОЖЕНИЕ 7 – ЛОГИЧЕСКАЯ МОДЕЛЬ 88](#_Toc100861016)

[ПРИЛОЖЕНИЕ 8 – ФИЗИЧЕСКАЯ МОДЕЛЬ 89](#_Toc100861017)

[ПРИЛОЖЕНИЕ 9 – ВХОДНЫЕ ДАННЫЕ 90](#_Toc100861018)

[ПРИЛОЖЕНИЕ 10 – ВЫХОДНЫЕ ДАННЫЕ 91](#_Toc100861019)

# ВВЕДЕНИЕ

В последнее время, сотрудники для оформления договоров учета сделок формируют их вручную, то есть пишут руками. Но я предлагаю свою программу «Учет сделок», которая сможет автоматизировать данный процесс, повысив эффективность работы предприятия.

Для поиска и оформления контрактов необходима информационная система. От заказчика поступают запросы на работу, на которые проектировщик составляет схемы и чертежи, а директор участвует в подписании контракта. При обращении к работнику его стандартные данные (фамилия, имя, отчество, должность, зарплата, логин и пароль) также фиксируются в базе данных. По каждому факту удовлетворения интересов обеих сторон составляется документ. В документе указываются цена, сотрудник, заказчик, объект, срок действия контракта.

Для эффективности работы программы по учету сделок необходимо ведение баз данных, содержащих полную информацию об оборудовании, заказчиках и объектах.

Программа «Учет сделок» является глобальным или абстрактным представлением хранилища данных.

Целью работы является разработка приложения для создания договоров газификации.

Для реализации данной цели необходимо решить следующие задачи:

* добавление информации о компаниях;
* добавление информации об объектах;
* добавление информации о договорах;
* формирование договора;
* добавление информации об оборудовании.

# 1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

## 1.1. Обоснование актуальности задачи

### 1.1.1. Актуальность темы

В настоящим времени приложения для предприятия являются весьма актуальными. Для любых предприятий нужна разработанная база данных для автоматизации работы по производству и учета в данной области. В задаче автоматизируется создание договора, о водоснабжении, где учитывается место куда будет поставляться вода, ее качество, объем и цена, утвержденная двумя сторонами.

Автоматизировав процесс, предприятие с легкостью сможет заключать договоры. Быстрая коммуникация между отделами упростит работу создания договора.

### 1.1.2. Анализ существующих разработок по данной теме

«AccountingTrans» - программа для сбора информации об объектах и создания excel таблицы, для записи полученных результатов и предоставляет возможность распечатывания данных показателей.

«Request» - универсальная программа для записи клиентов на подачу заявки договора на газификацию в ручном виде.

«DogToday» - справочная программа для предприятия. Здесь они могут просмотреть информацию с разных сайтов, для оптимизации работы и для того, чтоб вводить более точные данные.

Плюсы и минусы этих систем представлены в таблице 1.

Таблица 1 – плюсы и минусы систем автоматизации процесса.

|  |  |  |
| --- | --- | --- |
| Наименование | Плюсы | Минусы |
| «AccountingTrans» | Широкий функционал использования | Возможны подвисания |
| Имеет простой интерфейс | Нагружает работу компьютера |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| «Request» | Достаточная проста в эксплуатации | Полное программное обеспечения платное |
| Универсальная программа | Замедление работы СУБД при большом количестве информации |
| Модифицируемое приложение | Возможна нагрузка на работу компьютера |

|  |  |  |
| --- | --- | --- |
| «DogToday» | Широкий функционал | Платная программа |
| Техподдержка приложения | Сложный в понимании интерфейс |
| Удобная интеграция | Плохая оптимизация на старых устройствах |

Вывод: проанализировав данные представление в таблице, необходимо разработать программный продукт, который будет повторять плюсы представленных программ. Приложение должно иметь широкий функционал, должна быть иметь понятный обычному пользователю интерфейс. Необходимо решить такие минусы, как нагрузку на работу компьютера и отсутствие зависания БД при большом хранении информации.

## 1.2. Обоснование выбора и описание методологии разработки программы решения задачи

Каскадная модель – модель процесса разработки программного обеспечения, в которой процесс разработки выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки.

Следуя каскадной модели, разработчик переходит от одной стадии к другой строго последовательно.

Тем самым, каскадная модель подразумевает, что переход от одной фазы разработки к другой происходит только после полного и успешного завершения предыдущей фазы, и что переходов назад либо вперёд или перекрытия фаз — не происходит.

V-образная модель применима к системам, которым особенно важно бесперебойное функционирование. Особенностью модели можно считать то, что она направлена на тщательную проверку и тестирование продукта, находящегося уже на первоначальных стадиях проектирования. Стадия тестирования проводится одновременно с соответствующей стадией разработки.

В инкрементной модели полные требования к системе делятся на различные сборки. Терминология часто используется для описания поэтапной сборки ПО. Цикл разделен на более мелкие легко создаваемые модули. Каждый модуль проходит через фазы определения требований, проектирования, кодирования, внедрения и тестирования.

В RAD-модели компоненты или функции разрабатываются несколькими высококвалифицированными командами параллельно, будто несколько мини-проектов. Временные рамки одного цикла жестко ограничены. Созданные модули затем интегрируются в один рабочий прототип.

Проанализировав данные методологии по их описанию, была выбрана каскадная модель, так как все этапы происходят в строго последовательном порядке и решение задачи переходит на следующий этап в том случае, если предыдущий был завершен.

## 1.3. Обоснование выбора и описание инструментальных средств разработки программы решения задачи

Microsoft Visual Studio 2019 — это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию.

Visual Studio Code (VS Code) — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.

В качестве средства разработки клиентской части ПО использовалась среда программирования Microsoft Visual Studio 2019. Среда разработана в соответствии с концепцией визуального программирования.

В качестве языка программирования клиентской части данного проекта был выбран язык C#.

C# является языком программирования, который разработан для создания множества приложений, работающих в среде .NET Framework. Язык C# прост, типобезопасен и объектно-ориентирован. C# упрощает разработку компонентов программного обеспечения инкапсулированные сигнатуры методов, называемые делегатами, которые поддерживают типобезопасные уведомления о событиях;

Выбор СУБД

MySQL — это реляционная система управления базами данных с открытым исходным кодом. В настоящее время эта СУБД одна из наиболее популярных в веб-приложениях — подавляющее большинство CMS использует именно MySQL, а почти все веб-фреймворки поддерживают MySQL уже на уровне базовой конфигурации (без дополнительных модулей).

Решение Microsoft SQL Server 2019 представляет собой высокопроизводительную платформу обработки и анализа данных для бизнеса любого размера, которая отвечает всем современным требованиям по работе с данными любых типов и поддержке удобной и быстрой разработки приложений. Продукт реализует концепцию комплексной обработки данных и позволяет преобразовывать ценную информацию в знания.

Поэтому в качестве СУБД используется Microsoft SQL Server 2019.

## 1.4. Содержательная постановка задачи.

### 1.4.1. Описательная модель задачи

Приложение для учета нормативно-справочной информации об изделиях, изготавливаемых на предприятии должно хранить в себе определенную информацию по изделиям и их цену. Данная информация необходима инженеру-технологу для формирования справочной информации по готовому продукту.

### 1.4.2. Математическая модель

В данной задачи математическая формула отсутствует.

### 1.4.3. Описание входной информации

Описание входной информации представлены в таблицах 2.

Таблица 2 – Входная информация.

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Источник данных |
| ИНН | INN | Числовой | «Данные компании» |
| КПП | PPC | Числовой | «Данные компании» |
| БИК | BIC | Числовой | «Данные компании» |
| Расчетный счет | Payment\_Account | Числовой | «Данные компании» |
| Корреспонденсткий счет | Correspondent\_Account | Числовой | «Данные компании» |
| ОГРН | OGRN | Числовой | «Данные компании» |

### 1.4.4. Выходная информация

Описание выходной информации представлено в таблице 3.

Таблица 3 – Выходные данные

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Источник данных |
| Номер договора | ID\_Contract | Числовой | «Договор» |
| Дата заключения | Date\_Conclusion | Дата | «Договор» |
| Номер объекта | ID\_Object | Числовой | «Договор» |
| Номер сотрудника | ID\_Employee | Числовой | «Договор» |
| Цена | Price | Денежный | «Договор» |
| Номер договора | ID\_Contract | Числовой | «Договор» |
| Дата заключения | Date\_Conclusion | Дата | «Договор» |

### 1.4.5. Описание бизнес-процессов предприятия

Бизнес-процесс IDEF0 с детализацией представлен в приложении 1.

### 1.4.6. Описание потоков данных предприятия

Потоки данных DFD с детализацией представлены в приложении 2.

### 1.4.7. Диаграмма вариантов использования

Диаграмма вариантов использования представлена в приложении 3.

### 1.4.8. Диаграмма последовательности

Диаграмма последовательности представлена в приложении 4.

### 1.4.9. Диаграмма деятельности

Диаграммы деятельности представлена в приложении 5.

### 1.4.10. Диаграмма развертывания

Диаграмма развертывания представлена в приложении 6.

### 1.4.11. Архитектура приложения

Архитектура приложения представлена в приложении 7.

# 2. ПРОЕКТНАЯ ЧАСТЬ

## 2.1. Разработка структуры данных программы.

Описание структуры таблиц представлены в таблицах 5-

Таблица 4 – Описание таблицы «Банковские реквизиты»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код реквизитов | [ID\_Bank\_Details] | Числовой | Первичный ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации реквизитов |
| ИНН | INN | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | ИНН компании |
| КПП | PPC | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | КПП компании |

Продолжение таблицы 4 – Описание таблицы «Банковские реквизиты»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| БИК | BIC | Текстовый | Размер поля данных —максимальный;  Обязательное поле — да;  Индексированное поле — нет | БИК компании |
| Расчётный счёт | [Payment\_  Account] | Текстовый | Размер поля данных —максимальный;  Обязательное поле — да;  Индексированное поле — нет | Расчётный счёт банка |
| Корреспондентский счёт | [Correspondent\_  Account] | Текстовый | Размер поля данных —максимальный;  Обязательное поле — да;  Индексированное поле — нет | Корреспондент- счёт компании |
| ОГРН | OGRN | Текстовый | Размер поля данных —максимальный;  Обязательное поле — да;  Индексированное поле — нет | ОГРН компании |

Таблица 5 – Описание таблицы «Компания»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код компании | [ID\_Company] | Числовой | Первичный ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации компании |
| Название | [Title\_Company] | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да; | Название компании |
| Директор | Director | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да; | Директор компании |
| Адрес | Address | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Адрес компании |
| Номер телефона | [Number\_Phone] | Текстовый | Размер поля данных — 20;  Обязательное поле — да;  Индексированное поле — нет | Номер телефона компании |
| Код реквизитов | [ID\_Bank\_Details] | Числовой | Внешний ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации кода реквизитов |

Таблица 6 – Описание таблицы «Объект»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код объекта | Id | Числовой | Первичный ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации объекта |
| Название | [Title\_Object] | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Название объекта |
| Назначение | Purpose | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Назначение объекта |
| Режим работы | [Work\_Mode] | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Режим работы объекта |
| Адрес | Address | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Адрес объекта |
| Описание | Description | Текстовый | Размер поля данных — максимальный;  Обязательное поле — нет;  Индексированное поле — нет | Описание объекта |
| Номер компании | [ID\_Company] | Числовой | Внешний ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации компании |

Таблица 7 – Описание таблицы «Сотрудники»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код сотрудника | [ID\_Employee] | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Используется для идентификации сотрудника |
| Фамилия | Surname | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Фамилия сотрудника |

Продолжение таблицы 7 – Описание таблицы «Сотрудники»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Имя | Name | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Имя сотрудника |
| Отчество | [Middle\_Name] | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Отчество сотрудника |
| Должность | Post | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Должность сотрудника |
| Зарплата | Salary | Денежный | Условие на значение — не может быть отрицательным;  Обязательное поле — да | Зарплата сотрудника |
| Логин | Login | Текстовый | Размер поля данных — максимальный;  Обязательное поле — нет;  Индексированное поле — нет | Логин сотрудника |
| Пароль | Password | Текстовый | Размер поля данных — максимальный;  Обязательное поле — нет;  Индексированное поле — нет | Пароль сотрудника |

Таблица 8 – Описание таблицы «Оборудование\_Топливо»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код\_  оборудования | [ID\_Equipment] | Числовой | Составной ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Используется для идентификации оборудования |
| Код топлива | [ID\_Fuel] | Числовой | Составной ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Используется для идентификации топлива |

Таблица 9 – Описание таблицы «Котёл»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код оборудования | [ID\_Equipment] | Числовой | Первичный ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации котла |
| Марка | Stamp | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Марка котла |
| Модель | Model | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Модель котла |
| Мощность | Power | Числовой | Условие на значение — не может быть отрицательным;  Обязательное поле — да | Мощность котла |
| Единица измерения | [Unit\_Of\_  Measurment] | Текстовый | Размер поля данных — 10;  Обязательное поле — да;  Индексированное поле — нет | Единица измерения мощности |
| Мощность(Сумм) | [Power(Sum)] | Числовой | Условие на значение — не может быть отрицательным;  Обязательное поле — да | Общая мощность котлов |
| Количество | Quantity | Числовой | Условие на значение — не может быть отрицательным;  Обязательное поле — да | Кол-во котлов |
| Обслуживание | Service | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Каково обслуживание котлов |

Таблица 10 – Описание таблицы «Топливо»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Номер договора | [ID\_Contract] | Числовой | Первичный ключ;  Обязательное поле — да  Условие на значение — не может быть отрицательным; | Используется для идентификации договора |
| Дата заключения | [Date\_Conclusion] | Дата | Условие на значение — не может быть отрицательным;  Обязательное поле — да;  Индексированное поле — нет | Дата заключения договора |
| Действует по | [Valid\_For] | Дата | Условие на значение — не может быть отрицательным;  Обязательное поле — да; | Срок действия договора |
| Код объекта | [ID\_Object] | Числовой | Внешний ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации объекта |
| Код сотрудника | [ID\_Employee] | Числовой | Внешний ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации сотрудника |
| Код оборудования | [ID\_Equipment] | Числовой | Внешний ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации оборудования |

Таблица 11 – Описание таблицы «Вид топлива»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения (свойства поля) | Описание |
| Код вида топлива | [ID\_Type\_Fuel] | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Используется для идентификации вида топлива |
| Название вида топлива | [Title\_Type\_Fuel] | Текстовый | Размер поля данных — 20;  Обязательное поле — да;  Индексированное поле — нет | Название вида топлива |

Таблица 12 – Описание таблицы «Топливо»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код топлива | [ID\_Fuel] | Числовой | Первичный ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации топлива |
| Наименование | [Title\_Fuel] | Текстовый | Размер поля данных — максимальный;  Обязательное поле — да;  Индексированное поле — нет | Наименование топлива |
| Код вида топлива | [ID\_Type\_Fuel] | Числовой | Внешний ключ;  Условие на значение — не может быть отрицательным;  Обязательное поле — да | Используется для идентификации вида топлива |

## 2.2. Логическая модель программы.

Логическая модель представлена в приложении 8.

Физическая модель представлена в приложении 9.

## 2.3. Разработка пользовательского интерфейса программы

### 2.3.1. Разработка требований к интерфейсу

Основные требования к пользовательскому интерфейсу:

* понятность и логичность;
* обеспечение высокой скорости работы пользователя;
* обеспечение защиты от ошибок;
* приятная цветовая гамма

### 2.3.2. Проектирование и описание элементов управления

Макет окна авторизации представлен на рисунке 1.

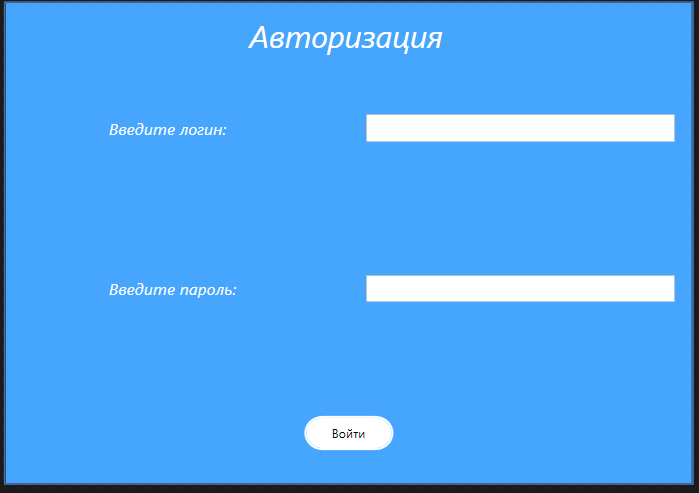


Рис. 1 – Макет окна авторизации

Интерфейс окна авторизации представлен на рисунке 2.

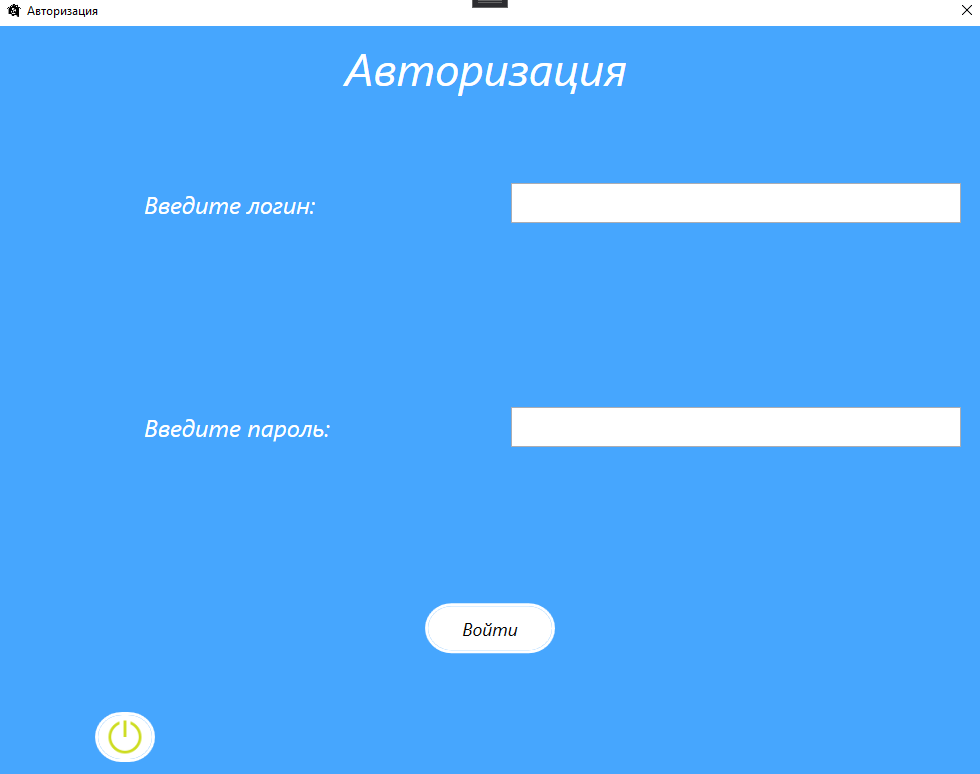


Рис. 2 – Интерфейс окна авторизации

Таблица 13 – Свойства и компоненты окна авторизации

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Border | - | Height="200" Width="200" Background="#27659c" |
| Label | - | FontFamily="Cambria Math" Content="Логин" HorizontalAlignment="Center" VerticalAlignment="Top" Foreground="#FFFBF5F5" FontSize="14" |
| TextBox | Login | HorizontalAlignment="Center" Height="23" TextWrapping="Wrap" VerticalAlignment="Center" Width="120" HorizontalContentAlignment="Center" FontFamily="Cambria Math" |
| Label | - | FontFamily="Cambria Math" Content="Пароль" HorizontalAlignment="Center" VerticalAlignment="Top" Foreground="#FFF7F1F1" FontSize="14" |
| PasswordBox | Pass | HorizontalAlignment="Center" VerticalAlignment="Top"  Width="120" HorizontalContentAlignment="Center" |
| Button | Input | Content="Вход" HorizontalAlignment="Center" VerticalAlignment="Center" Width="76" Click="Input\_Click" Foreground="#FFF7F1F1"  Margin="40"  FontFamily="Cambria Math" FontSize="14" |
| Button | Exit | Background="{x:Null}" Click="Exit\_Click" BorderBrush="{x:Null}" Foreground="{x:Null}" Margin="543,10,10,340"> |
| Image | - | Source="Image/closeBtn.png" Height="46" Width="50" |

Макет страницы «Главное меню» представлен на рисунке 3

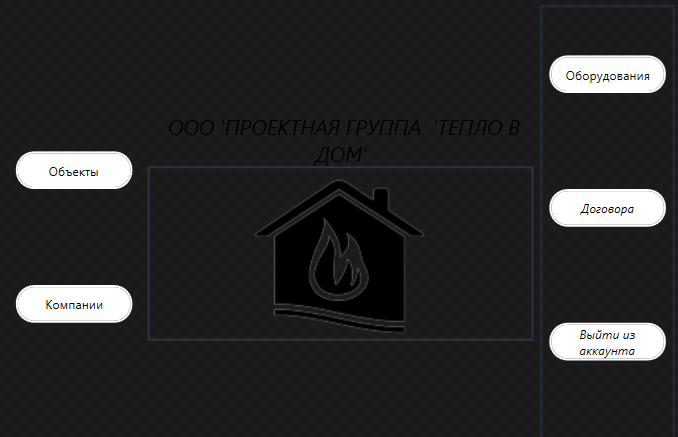


Рис. 3 – макет страницы «Главное меню»

Интерфейс страницы «Главное меню» представлен на рисунке 4.

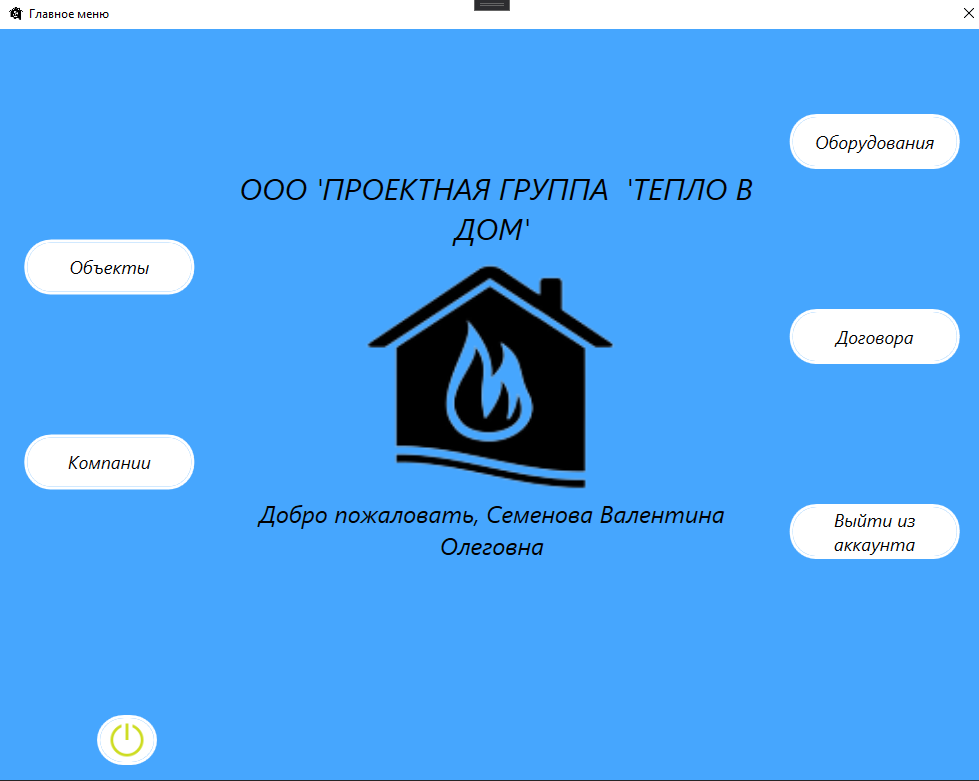


Рис. 4 – интерфейс страницы «Главное меню»

Таблица 14 – Свойства и компоненты страницы «Главное меню»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Border | border | BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True" |
| Grid | BG | PreviewMouseLeftButtonDown="BG\_PreviewMouseLeftButtonDown" Background="White" |
| Button | CloseBtn | Grid.Column="1" Grid.Row="1" Height="40" Width="40" BorderThickness="0" Style="{StaticResource Close\_btn}" |
| Grid | nav\_pnl | HorizontalAlignment="Left" Width="65" Background="#27659c" |
| TextBlock | - | Grid.Row="1" Grid.Column="0" Margin="73,-20,0,0" Text="М Е Н Ю" Foreground="White" FontSize="22" Background="Transparent" HorizontalAlignment="Left" VerticalAlignment="Center" Style="{StaticResource Up\_style}" |
| ListView | LV | Background="Transparent" BorderBrush="Transparent" ScrollViewer.HorizontalScrollBarVisibility="Disabled" Height="507" |
| Button | ImgContBtn | Style="{StaticResource font\_style}" Background="{x:Null}" BorderBrush="{x:Null}" |
| Button | ContBtn | Content="Договор" Margin="20,0,0,0" Style="{StaticResource font\_style}" Background="{x:Null}" BorderBrush="{x:Null}" |
| Button | Subscriber | Content="Абонент" Margin="25,0,0,0" Style="{StaticResource font\_style}" Background="{x:Null}" BorderBrush="{x:Null}" |
| Button | ImgBtnSvoistvo | Click="ImgBtnSvoistvo\_Click" |
| Button | Svoistvo | Content="Свойства" Margin="25,0,0,0" Style="{StaticResource font\_style}" Background="{x:Null}" BorderBrush="{x:Null}" |
| Button | NovAvtorizachia | Content="К авторизации" Margin="25,0,0,0" Style="{StaticResource font\_style}" Background="{x:Null}" BorderBrush="{x:Null} |

Макет страницы «Договор» представлен на рисунке 5.

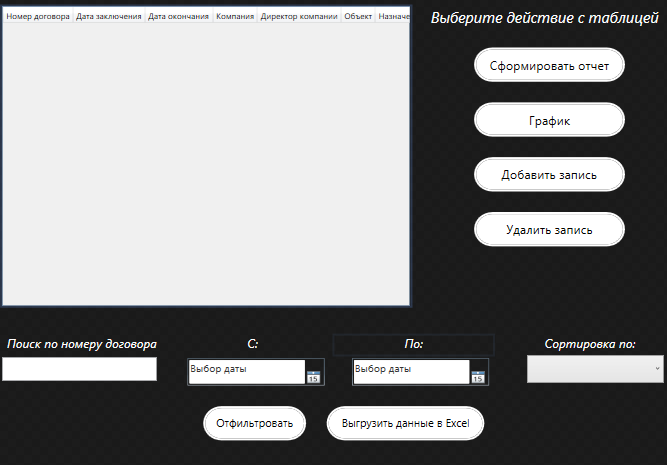


Рис. 5 – макет страницы «Договор»

Интерфейс страницы «Договор» представлен на рисунке 6.



Рис. 6 – интерфейс страницы «Договор»

Таблица 15 – Свойства и компоненты страницы «Договор»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="White |
| DataGrid | dgCont | Margin="26,10,370,121" Grid.Row="1" CanUserAddRows="False" AutoGenerateColumns="False" |
| Button | BtnWord | FontSize="15" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="26,312.4,1005.4,76.6" Grid.Row="1 |
| Button | tnExcel | FontSize="15" Click="BtnExcel\_Click" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="100,312,939,74" Grid.Row="1" |
| Button | btnDob | FontSize="15" Content="Добавить запись" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Button | btnEzm | FontSize="15" Content="Измененить запись" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" RenderTransformOrigin="0,1" CornerRadius="20" Margin="7" |
| Button | btnDelete | FontSize="15" Content="Удалить запись" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| StackPanel | - | Grid.Row="1" Grid.RowSpan="1" HorizontalAlignment="Right" Width="308" Margin="0,0,25,0" |
| TextBlock | - | FontSize="19" Text="Выберите действие с таблицей" Height="30" Width="300" Margin="10" |

Продолжение таблицы 15

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Button | SortDateS | FontSize="15" Content="Сортировка по оплате" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Button | SortDateF | FontSize="15" Content="Сортировка по дате окончания" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| TextBlock | - | FontSize="19" Text="Приложения для договора" Height="30" Width="300" Margin="10" |
| Border | - | Background="#27659c" - CornerRadius="20" Margin="7" |
| Button | VolumeBtn | FontSize="15" Content="Смотреть объём потребления" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" - CornerRadius="20" Margin="7" |
| Button | ShowingBtn | FontSize="15" Content="Смотреть качество воды" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Label | - | FontFamily="Cambria Math" Content="Поиск по оплате" FontSize="20" - |
| TextBox | SearchPay | FontFamily="Cambria Math" Margin="50,0,50,0" Height="25" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="166,16,810,46" Width="110" |
| Button | SubscriberBtn | Content="Абонент" Width="105" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |

Макет страницы «Компания» представлен на рисунке 7.

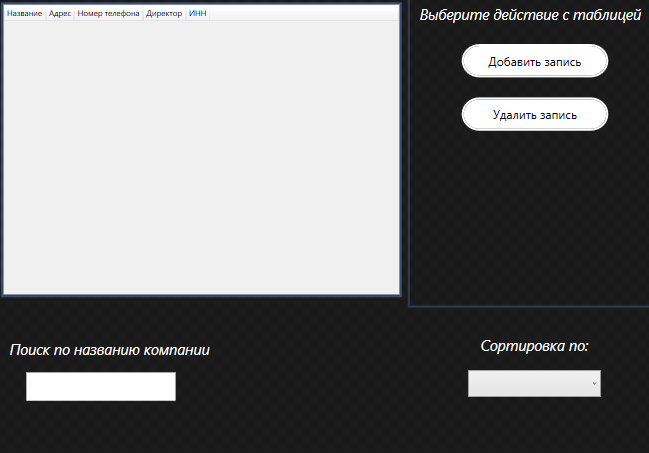


Рис. 7 – макет страницы «Компания»

Интерфейс страницы «Компания» представлен на рисунке 8

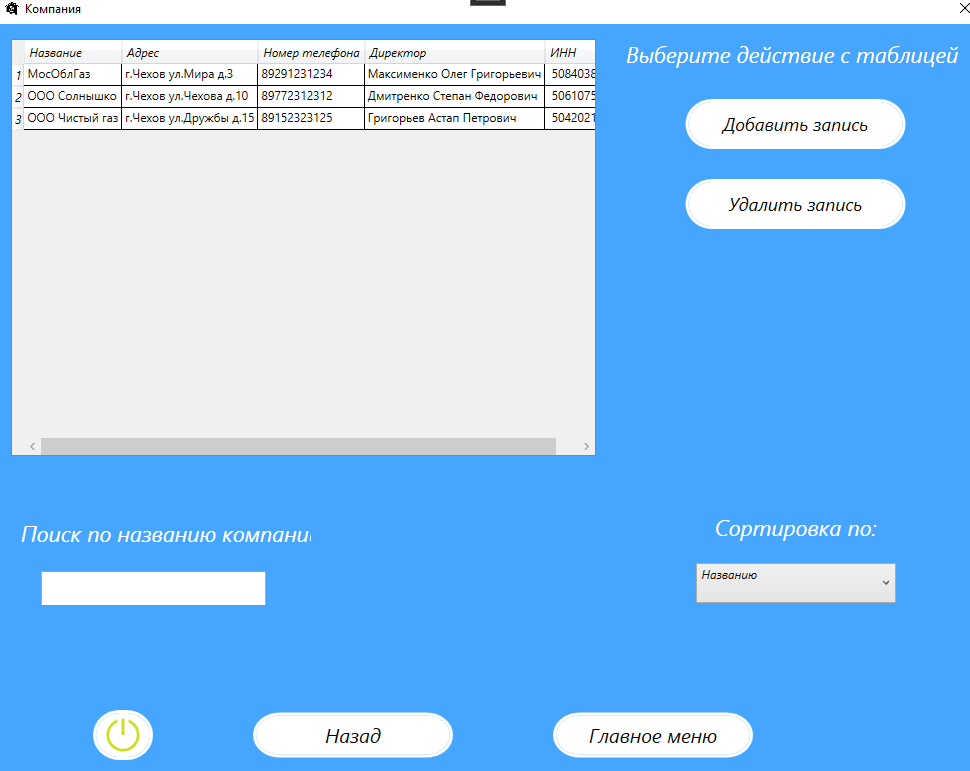


Рис. 8 – Интерфейс страницы «Компания»

Таблица 16 – Свойства и компоненты страницы «Компания»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| DataGrid | dgQuality | Margin="26,10,370,121" |
| Button | WordBtn | FontSize="15" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="913,247.4,84.4,70.4" |
| TextBlock | - | FontSize="19" Text="Выберите действие с таблицей" Height="30" Width="300" Margin="10" |
| Border | - | Background="#27659c" RenderTransformOrigin="0,1" CornerRadius="20" Margin="7" |
| Button | SortNumInd | FontSize="15" Content="Сортировка по номеру паказателя" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="7" |
| Label | - | FontFamily="Cambria Math" Content="Поиск по номеру показателя" FontSize="20" HorizontalAlignment="Center" |
| StackPanel | - | Grid.Row="1" Grid.Column="1" VerticalAlignment="Top" |
| TextBox | SearchNumInd | FontFamily="Cambria Math" Margin="50,0,50,0" Height="25" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="0,16,360,46" |
| Button | SubBtn | Content="Абонент" Width="105" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Button | ContractPG | Content="Договор" Width="105" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Button | WaterVPG | Content="Объём потребления воды" Width="160" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" |

Макет страницы «Объекты» представлен на рисунке 9.

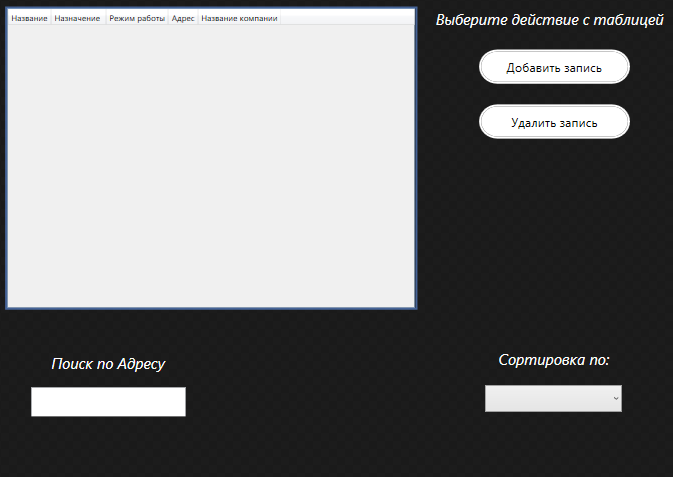


Рис. 9 – макет страницы «Объекты»

Интерфейс страницы «Объекты» представлен на рисунке 10.

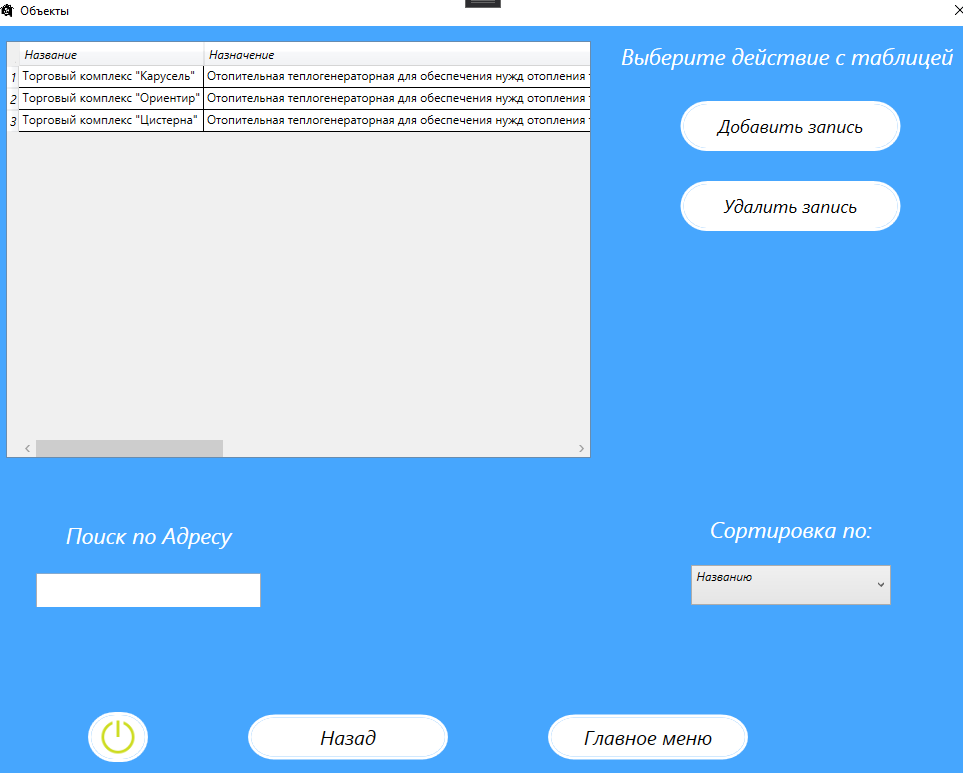


Рис. 10 – интерфейс страницы «Объекты»

Таблица 17 – Свойства и компоненты страницы «Объекты».

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="White" |
| DataGrid | dgVol | Margin="26,10,370,121" |
| Button | WordBtn | FontSize="15" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="72,332.4,880.4,9.6" |
| TextBlock | - | FontSize="19" Text="Выберите действие с таблицей" Height="30" Width="300" Margin="10" |
| Border | - | Background="#27659c" RenderTransformOrigin="0,1" CornerRadius="20" Margin="7" |
| Button | SortPrice | FontSize="15" Content="Сортировка по стоимости" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" RenderTransformOrigin="0,1" CornerRadius="20" Margin="7" |
| Button | SortTotalVol | FontSize="15" Content="Сортировка по общему объёму" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Label | - | FontFamily="Cambria Math" Content="Поиск по номеру объёма" FontSize="20" HorizontalAlignment="Center" |
| TextBox | SearchNumVol | FontFamily="Cambria Math" Margin="50,0,50,0" Height="25" |
| Border | - | Background="#27659c" CornerRadius="20" HorizontalAlignment="Center" Margin="0,16,360,46" |
| Button | SubBtn | Width="105" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" RenderTransformOrigin="0,1" CornerRadius="20" HorizontalAlignment="Center" Margin="0,16,100,46" |

Макет страницы «Оборудование» представлен на рисунке 11.

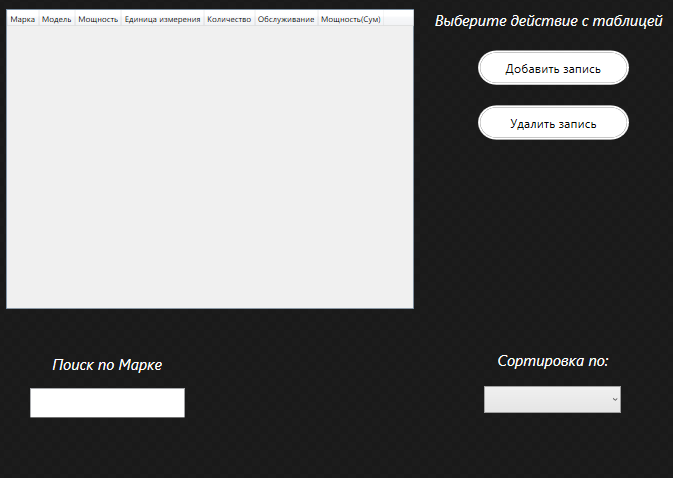


Рис. 11 – макет страницы «Оборудование»

Интерфейс страницы «Оборудование» представлен на рисунке 12.

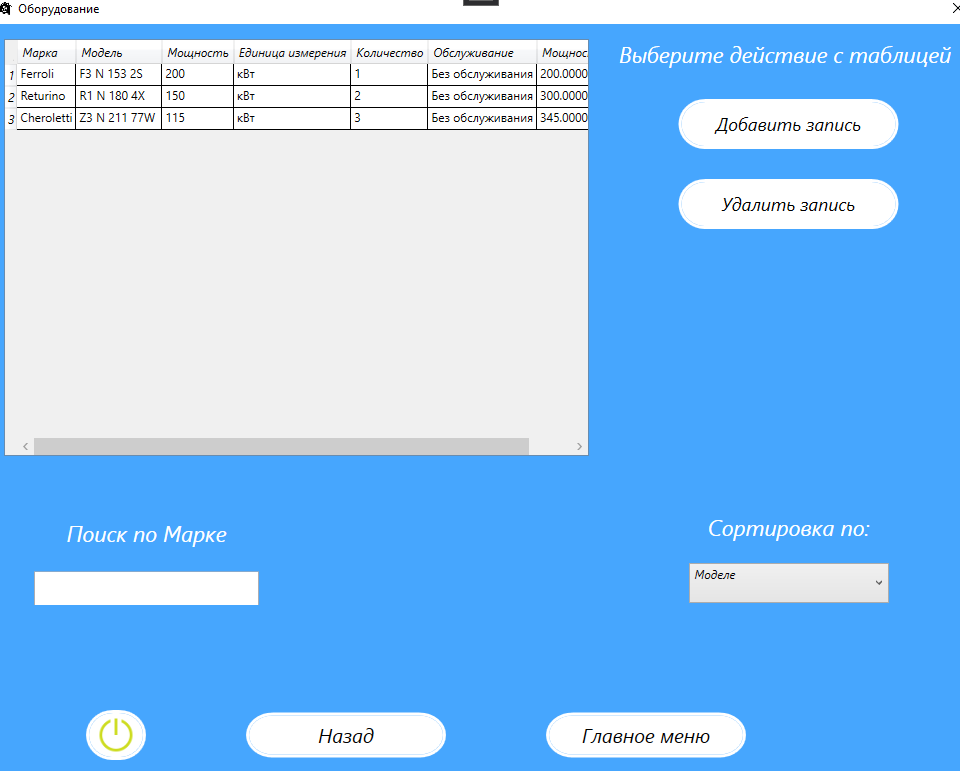


Рис. 12 – интерфейс страницы «Оборудование»

Таблица 18 – Свойства и компоненты страницы «Оборудование».

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="White" |
| DataGrid | dgSub | Margin="26,10,370,121" |
| TextBlock | - | FontSize="19" Text="Выберите действие с таблицей" Height="30" Width="300" Margin="10" |
| Button | btnDob | FontSize="15" Content="Добавить запись" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0 |
| Border | - | Background="#27659c" CornerRadius="20" Margin="7" |
| Button | btnEzm | FontSize="15" Content="Измененить запись" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="7" |
| Button | btnDelete | FontSize="15" Content="Удалить запись" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="7" |
| Button | SortNameOrg | FontSize="15" Content="Сортировка по названию организации" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Button | SortDir | Width="105" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math"  Margin="10,0" |
| TextBlock | - | FontSize="19"  Text="Приложения для договора" Height="30" Width="300" Margin="10" |
| Border | - | Background="#27659c"  CornerRadius="20" Margin="7" |
| Button | PlaceBtn | FontSize="15" Content="Место исполнения" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" |

Макет страницы «Добавление договора» представлен на рисунке 13.

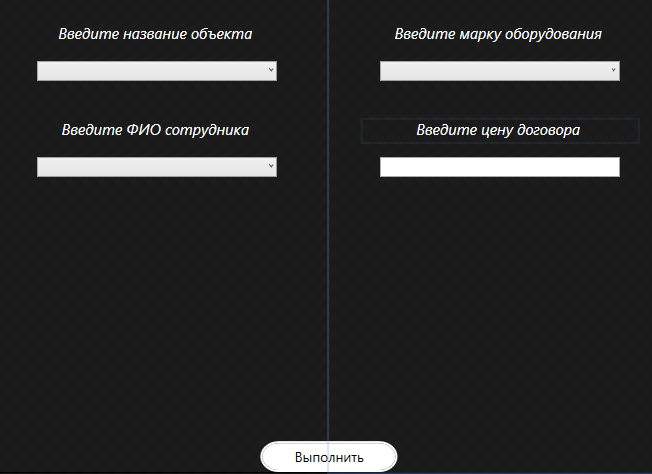


Рис. 13 – макет страницы «Добавление договора»

Интерфейс страницы «Добавление договора» представлен на рисунке 14.

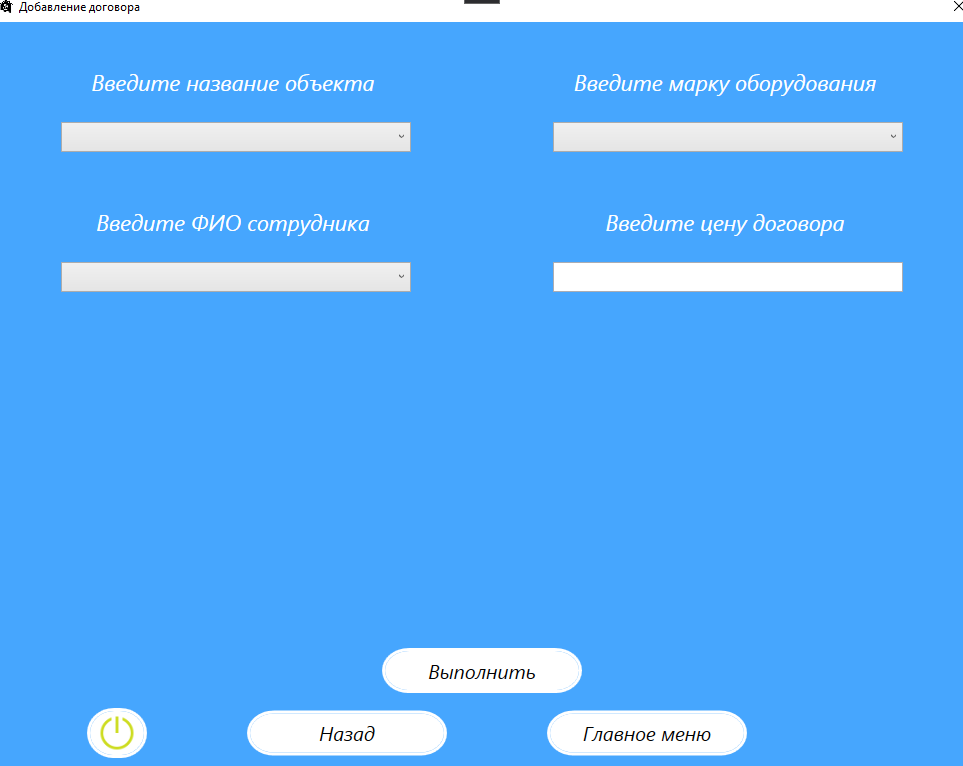


Рис. 14 – интерфейс страницы «Добавление договора»

Таблица 19 – Свойства и компоненты страницы «Добавление договора».

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="White" |
| DataGrid | dgPlace | Margin="26,10,370,121" |
| TextBlock | - | FontSize="19" Text="Выберите действие с таблицей" Height="30" Width="300" Margin="10" |
| Button | btnDob | FontSize="15" Content="Добавить запись" Height="30"  Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0 |
| Border | - | Background="#27659c" CornerRadius="20" Margin="7" |
| Button | btnEzm | FontSize="15" Content="Измененить запись" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="7" |
| Button | btnDelete | FontSize="15" Content="Удалить запись" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="7" |
| Button | SortName | FontSize="15" Content="Сортировка по названию организации" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" CornerRadius="20" HorizontalAlignment="Center" Margin="0,16,360,46" |
| Button | SortNameOrg | Width="105" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| TextBlock | - | FontSize="19"  Text="Приложения для договора" Height="30" Width="300"  Margin="10" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="7" |

Макет страницы «Добавление компании» представлен на рисунке 15.

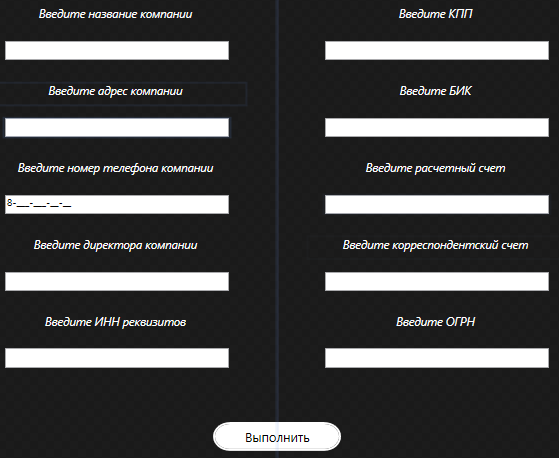


Рис. 15 – макет страницы «Добавление компании»

Интерфейс страницы «Добавление компании» представлен на рисунке 16.

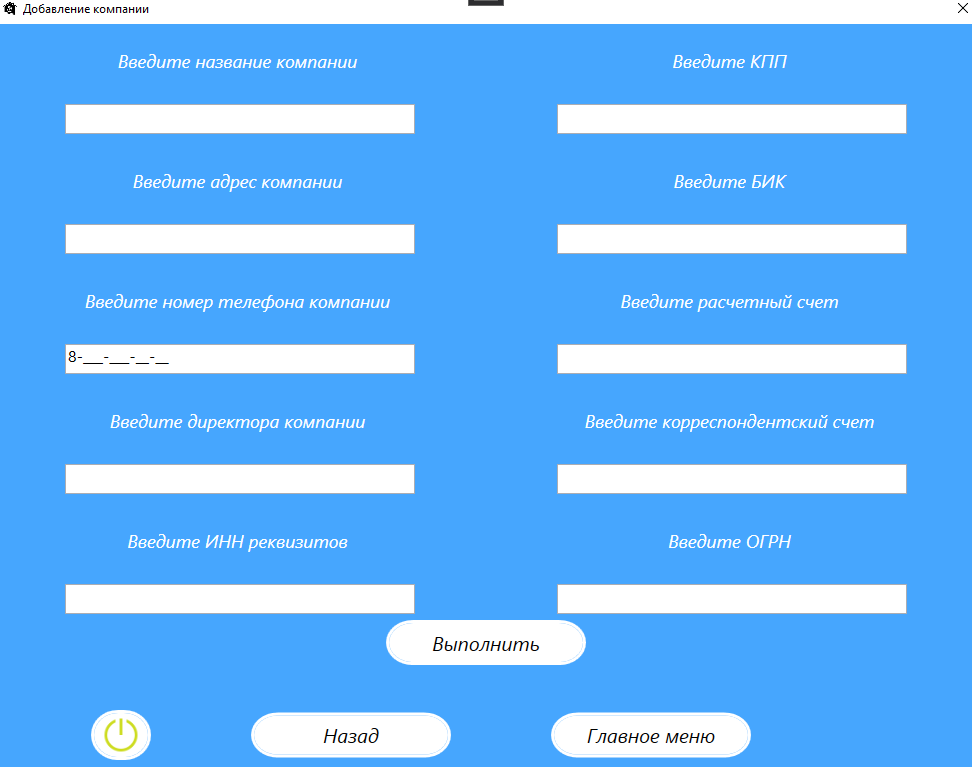


Рис. 16 – интерфейс страницы «Добавление компании»

Таблица 20 – Свойства и компоненты страницы «Добавление компании».

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="White" |
| TextBlock | dgPlace | FontFamily="Cambria Math" Margin="207,111,572,458" Height="30" Text="Введите наименование абонента " FontSize="20" |
| ComboBox | NameOrg | Margin="587,111,283,472" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,164,627,405" Height="30" Text="Введите номер показателя " FontSize="20" |
| ComboBox | NumInd | Margin="587,164,283,416" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,214,664,355" Height="30" Text="Введите номер объема " FontSize="20 |
| ComboBox | NumVol | Background="#27659c" CornerRadius="20" Margin="7" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,265,646,304" Height="30" Text="Введите ФИО сотрудника " FontSize="20" |
| ComboBox | CbName | Margin="665,265,265.4,316.2" FontFamily="Cambria Math" |
| ComboBox | CbLast | Margin="475,265,455.4,316.2" FontFamily="Cambria Math" |
| ComboBox | CbOtch | Margin="840,265,84.4,316.2" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,309,724,260" Height="30" Text="Введите оплату " FontSize="20" |
| TextBox | - | x:Name="Pay" Margin="587,309,283,273" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,356,683,213" Height="30" Text="Введите дату начала " FontSize="20" |
| DatePicker | DateStart | Margin="587,356,283,213" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,402,646,167" Height="30" Text="Введите дату окончания " FontSize="20" |
| DatePicker | DateEnd | Margin="587,402,283,167" FontFamily="Cambria Math" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="349,497,627,54" |

Макет страницы «Добавление объекта» представлен на рисунке 17.

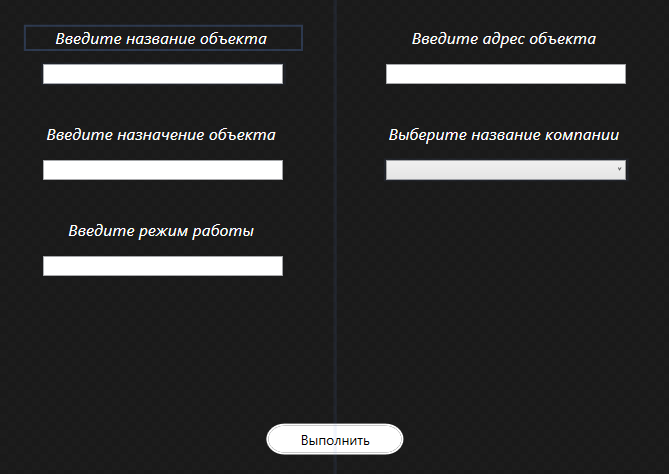


Рис. 17 – макет страницы «Добавление объекта»

Интерфейс страницы «Добавление объекта» представлен на рисунке 18.



Рис. 18 – интерфейс страницы «Добавление объекта»

Таблица 21 – Свойства и компоненты страницы «Добавление объекта».

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="White" |
| TextBlock | TN | FontFamily="Cambria Math" Margin="207,111,572,458" Height="30" |

Продолжение таблицы 21

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| TextBox | NameOrg | Height="20" Margin="587,161,283,419" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,214,664,355" Height="30" Text=" Введите расчетный счет" FontSize="20 |
| TextBox | CheckAcc | Height="20" Margin="587,214,283,366" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,265,646,304" Height="30" Text="Введите корреспондентский счёт" FontSize="20" |
| TextBox | CorrAcc | Height="20" Margin="587,265,283,315" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="190,319,708,250" Height="30" Text="Введите директора "  FontSize="20" |
| TextBox | Dir | Height="20" Margin="587,319,283,260" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="190,371,758,198" Height="30" Text="Введите адрес "  FontSize="20" |
| TextBox | - | x:Name="Pay" Margin="587,309,283,273" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,356,683,213" Height="30" Text="Введите дату начала " FontSize="20" |
| Border | - | Background="#27659c"  CornerRadius="20"  Margin="358,490,616,60 |
| Button | Ok | Content="Выполнить" FontSize="15" Height="25"  Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math"  Margin="10,0" |
| Border | - | Background="#27659c"  CornerRadius="20" Margin="559,490,416,61" |
| Button | NOk | Content="Отмена" FontSize="15" Height="25" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math"  Margin="10,0" |

Макет страницы «Добавление оборудования» представлен на рисунке 19.

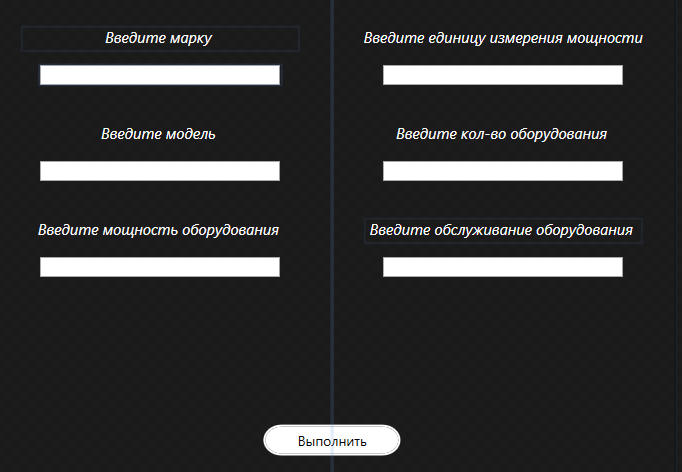


Рис. 19 – макет страницы «Добавление оборудования»

Интерфейс страницы «Добавление оборудования» представлен на рисунке 20.

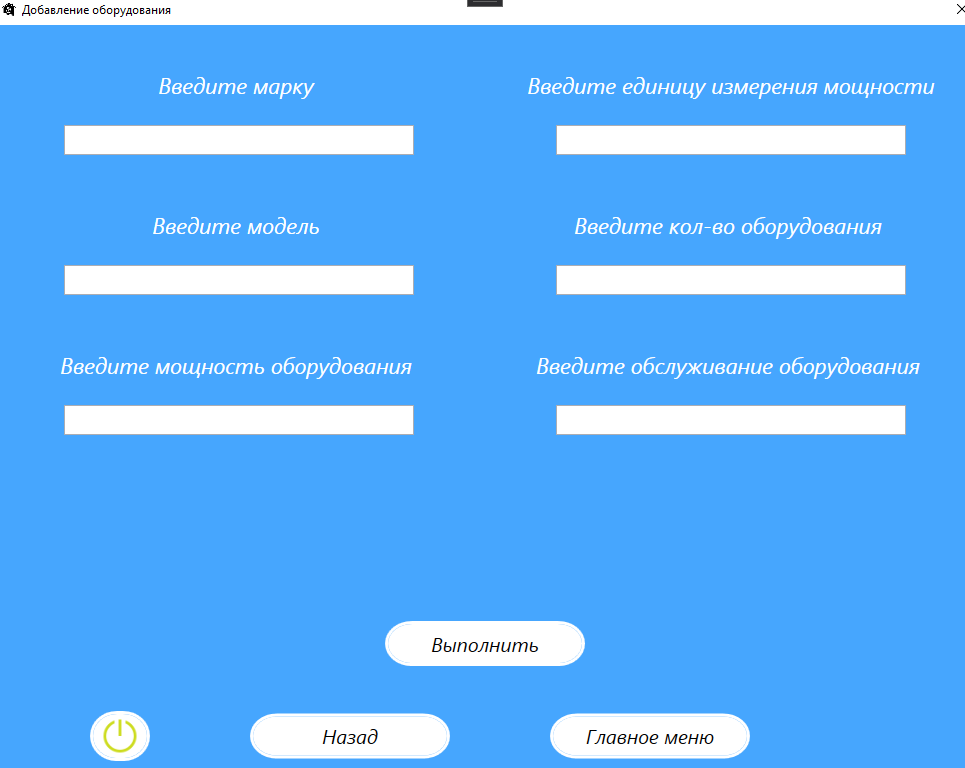


Рис. 20 – интерфейс страницы «Добавление оборудования»

Таблица 22 – Свойства и компоненты страницы «Добавление оборудования».

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="White" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,111,572,458" Height="30" Text="Введите наименование" FontSize="20" |
| TextBox | TitlePlace | Height="20" Margin="587,108,283,472" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,164,627,405" Height="30" Text=" Введите адрес" FontSize="20" |
| TextBox | AddressPlace | Height="20" Margin="587,161,283,419" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,214,664,355" Height="30" Text="Введите название организации" FontSize="20 |
| ComboBox | NameOrg | Height="20" Margin="576,265,294,314" FontFamily="Cambria Math" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="357,400,618,151" |
| Button | Ok | Content="Выполнить" FontSize="15" Height="25" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="555,400,420,151" |
| Button | NOk | Content="Отмена" FontSize="15" Height="25" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |

## 2.4. Листинг программы

Листинг логики «MainWindow» представлен в листингах 1-2.

public MainWindow()

{

Листинг 1 - «MainWindow»

InitializeComponent();

MainFrame.Content = new Authorization();

}

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

{

MessageBoxResult Result = MessageBox.Show("Закрыть приложение ?", "Закрытие", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (Result == MessageBoxResult.Yes)

{

Environment.Exit(0);

}

}

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

{

MainFrame.Content = new MainMenu();

}

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

{

{

if (MainFrame.CanGoBack)

{

MainFrame.GoBack();

}

}

}

private void MainFrame\_Navigated(object sender, NavigationEventArgs e)

{

if (WinMW.Title == "Главное меню" || WinMW.Title == "Авторизация" || WinMW.Title == "")

{

BtnMainMenu.Visibility = Visibility.Hidden;

BtnBack.Visibility = Visibility.Hidden;

}

else

{

BtnMainMenu.Visibility = Visibility.Visible;

BtnBack.Visibility = Visibility.Visible;

}

}

Листинг 2 - «MainWindow»

Листинг логики «MainMenu» представлен в листингах 3-4.

public MainMenu()

{

InitializeComponent();

TbUser.Text = "Добро пожаловать, " + App.workEmployee.Surname + " " + App.workEmployee.Name + " " + App.workEmployee.Middle\_Name;

}

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

{

NavigationService.Content = new ObjectPage();

}

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

{ NavigationService.Content = new CompaniesPage();

Листинг 3 - «MainMenu»

}

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

{

NavigationService.Content = new EquipmentPage();

}

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

{

NavigationService.Content = new ContractPage();

}

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

{

NavigationService.Content = new Authorization();

}

Листинг 4 - «MainMenu»

Листинг логики «Authorization» представлен в листингах 5-6.

public Authorization()

{

InitializeComponent();

}

private string sha256(string password)

{

SHA256Managed sHA256 = new SHA256Managed();

byte[] hash = sHA256.ComputeHash(Encoding.UTF8.GetBytes(password));

StringBuilder result = new StringBuilder();

for (int i = 0; i < hash.Length; i++)

{

result.Append(hash[i].ToString("x2"));

}

return result.ToString();

}

private void decodePassword()

{

var employees = App.db.Employee.ToList();

for (int i = 0; i < employees.Count; i++)

{

employees[i].Password = sha256(employees[i].Password);

App.db.SaveChanges();

}

}

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

{

try

{

string securePassword = sha256(password.Password);

Employee enterUser = App.db.Employee.FirstOrDefault(u => u.Login == login.Text && u.Password == securePassword);

if (enterUser != null)

{

if (enterUser.Post == "Менеджер")

{

App.workEmployee = enterUser;

NavigationService.Content = new MainMenu();

}

else

if (enterUser.Post == "Технический инспектор проектного отдела")

Листинг 5 - «Authorization»

{

App.workEmployee = enterUser;

NavigationService.Content = new MainMenu();

}

}

else throw new Exception("Сотрудник не найден! Введен неверный логин или пароль");

return;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

Листинг 6 - «Authorization»

Листинг логики «CompaniesPage» представлен в листингах 7-9.

public CompaniesPage()

{

InitializeComponent();

GetData();

CbSort.SelectedIndex = 0;

if (App.workEmployee.Post == "Технический инспектор проектного отдела")

{

BtnAdd.Visibility = Visibility.Hidden;

BtnDel.Visibility = Visibility.Hidden;

tbAction.Visibility = Visibility.Hidden;

}

}

private void OnlyLetters(object sender, TextCompositionEventArgs e)

{

Regex inp = new Regex("^[А-Я]+", RegexOptions.IgnoreCase);

Regex ing = new Regex("^[A-Z]+", RegexOptions.IgnoreCase);

if (!inp.IsMatch(e.Text) && !ing.IsMatch(e.Text))

e.Handled = true;

}

private void GetData(string search="",string Filter = "")

{

try

{

var companies = App.db.Company.ToList();

if (!String.IsNullOrEmpty(search)&&!String.IsNullOrWhiteSpace(search))

{

companies = companies.Where(c => c.Title\_Company.ToLower().Contains(search.ToLower())).ToList();

}

if (!String.IsNullOrEmpty(Filter) && !String.IsNullOrWhiteSpace(Filter))

{

if (Filter=="Директору")

{

companies = companies.ToList().OrderBy(c=>c.Director).ToList();

}

else if (Filter == "Названию")

{

Листинг 7 - «CompaniesPage»

companies = companies.ToList().OrderBy(c => c.Title\_Company).ToList();

}

else if (Filter == "Адресу")

{

companies = companies.ToList().OrderBy(c => c.Address\_Company).ToList();

}

}

if (companies.Count != 0)

{

dgCompany.ItemsSource = companies.ToList();

}

else throw new Exception("Данные не найдены!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

GetData(TbSearch.Text, CbSort.SelectedItem.ToString());

}

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

{

NavigationService.Content = new CompanyEditPage();

}

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

{

if (dgCompany.SelectedItem != null)

{

try

{

var cdwd = dgCompany.SelectedItem as Company;

var choice = MessageBox.Show("Вы точно хотите удалить выбранные данные?", "Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (choice == MessageBoxResult.Yes)

{

App.db.Company.Remove(cdwd);

App.db.SaveChanges();

dgCompany.ItemsSource = App.db.Company.ToList();

MessageBox.Show("Удаление успешно!", "Готово", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (DbUpdateException ex)

{

MessageBox.Show(ex.Message, "Ошибка!");

}

catch (ArgumentNullException ex)

{

MessageBox.Show(ex.Message, "Ошибка!");

}

}

else MessageBox.Show("Выберите запись которую хотите удалить", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

Листинг 8 - «CompaniesPage»

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

{

GetData(TbSearch.Text, ((ComboBoxItem)CbSort.SelectedValue).Content.ToString());

}

private void dgCompany\_LoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

Листинг 9 - «CompaniesPage»

Листинг логики «CompanyEditPage» представлен в листингах 10-11.

public CompanyEditPage()

{

InitializeComponent();

}

private void OnlyLetters(object sender, TextCompositionEventArgs e)

{

Regex inp = new Regex("^[А-Я]+", RegexOptions.IgnoreCase);

Regex ing = new Regex("^[A-Z]+", RegexOptions.IgnoreCase);

if (!inp.IsMatch(e.Text) && !ing.IsMatch(e.Text))

e.Handled = true;

}

private void OnlyNumbers(object sender, TextCompositionEventArgs e)

{

if (!Char.IsDigit(e.Text, 0))

{

e.Handled = true;

}

}

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

{

try

{

var choice = MessageBox.Show("Вы точно хотите добавить данный договор ? ", "Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (choice == MessageBoxResult.Yes)

{

if (TbTitleCompany.Text != null && TbPPC.Text != null && TbPaymentAccount.Text != null && TbOGRN.Text != null && TbNumberPhone.Text != null && TbINN.Text != null && TbDirector.Text != null && TbCorrespondent\_Account.Text != null && TbBIC.Text != null && TbAddress.Text != null)

{

Company newCompany = new Company();

Bank\_Details newBank = new Bank\_Details();

newBank.ID\_Bank\_Details = App.db.Bank\_Details.Max(c => c.ID\_Bank\_Details) + 1;

newBank.INN = TbINN.Text;

newBank.BIC = TbBIC.Text;

newBank.Payment\_Account = TbPaymentAccount.Text;

newBank.Correspondent\_Account = TbCorrespondent\_Account.Text;

newBank.OGRN = TbOGRN.Text;

newBank.PPC = TbPPC.Text;

newCompany.ID\_Company = App.db.Company.Max(c => c.ID\_Company) + 1;

newCompany.Title\_Company = TbTitleCompany.Text;

newCompany.Number\_Phone = TbNumberPhone.Text;

newCompany.Address\_Company = TbAddress.Text;

Листинг 10 - «CompanyEditPage»

newCompany.Director = TbDirector.Text;

newCompany.ID\_Bank\_Details = App.db.Bank\_Details.Max(c=>c.ID\_Bank\_Details) + 1;

App.db.Bank\_Details.Add(newBank);

App.db.Company.Add(newCompany);

App.db.SaveChanges();

MessageBox.Show("Компания добавлена!", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.Content = new CompaniesPage();

}

else MessageBox.Show("Введены некорректные данные или остались незаполненные поля!", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

Листинг 11 - «CompanyEditPage»

Листинг логики «ContractEditPage» представлен в листингах 12-13.

public ContractEditPage()

{

InitializeComponent();

CbTitleObject.ItemsSource = App.db.Object.ToList();

CbFIOEmployee.ItemsSource = App.db.Employee.ToList();

CbStampEquipment.ItemsSource = App.db.Main\_Equipment\_Boiler\_.ToList();

}

private void OnlyNumbers(object sender, TextCompositionEventArgs e)

{

if (!Char.IsDigit(e.Text, 0))

{

e.Handled = true;

}

}

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

{

try

{

var choice = MessageBox.Show("Вы точно хотите добавить данный договор ? ", "Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (choice == MessageBoxResult.Yes)

{

if (CbFIOEmployee.Text != null && CbStampEquipment.Text != null && CbTitleObject.Text != null && TbPrice.Text != "")

{

Contract newContract = new Contract();

newContract.ID\_Contract = App.db.Contract.Max(c => c.ID\_Contract) + 1;

newContract.ID\_Equipment = (CbStampEquipment.SelectedItem as Main\_Equipment\_Boiler\_).ID\_Equipment;

newContract.ID\_Employee = (CbFIOEmployee.SelectedItem as Employee).ID\_Employee;

newContract.ID\_Object = (CbTitleObject.SelectedItem as Models\_BD.Object).ID\_Object;

newContract.Price = Convert.ToInt32(TbPrice.Text);

newContract.Date\_Conclusion = DateTime.Now;

newContract.Valid\_For = DateTime.Now.AddDays(90);

Листинг 12 - «ContractEditPage»

App.db.Contract.Add(newContract);

App.db.SaveChanges();

MessageBox.Show("Договор добавлен!", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.Content = new ContractPage();

}

else MessageBox.Show("Введены некорректные данные или остались незаполненные поля!", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

Листинг 13 - «ContractEditPage»

Листинг логики «ContractPage» представлен в листингах 14-20.

public ContractPage()

{

InitializeComponent();

dgContract.ItemsSource = App.db.Contract.ToList();

CbSort.SelectedIndex = 0;

if (App.workEmployee.Post == "Технический инспектор проектного отдела")

{

BtnAdd.Visibility = Visibility.Hidden;

BtnDel.Visibility = Visibility.Hidden;

tbAction.Visibility = Visibility.Hidden;

}

}

private void OnlyNumbers(object sender, TextCompositionEventArgs e)

{

if (!Char.IsDigit(e.Text, 0))

{

e.Handled = true;

}

}

private void GetData(string search = "", string Filter = "")

{

try

{

var contracts = App.db.Contract.ToList();

if (!String.IsNullOrEmpty(search) && !String.IsNullOrWhiteSpace(search))

{

contracts = contracts.Where(c => c.ID\_Contract.ToString().Contains(search.ToLower())).ToList();

}

if (!String.IsNullOrEmpty(Filter) && !String.IsNullOrWhiteSpace(Filter))

{

if (Filter == "Дате заключения")

{

contracts = contracts.ToList().OrderBy(c => c.Date\_Conclusion).ToList();

}

else if (Filter == "Дате окончания")

Листинг 14 - «ContractPage»

{

contracts = contracts.ToList().OrderBy(c => c.Valid\_For).ToList();

}

else if (Filter == "Директору")

{

contracts = contracts.ToList().OrderBy(c => c.Object.Company.Director).ToList();

}

else if (Filter == "Компании")

{

contracts = contracts.ToList().OrderBy(c => c.Object.Company.Title\_Company).ToList();

}

else if (Filter == "Объекту")

{

contracts = contracts.ToList().OrderBy(c => c.Object.Title\_Object).ToList();

}

else if (Filter == "Цене")

{

contracts = contracts.ToList().OrderBy(c => c.Price).ToList();

}

}

if (contracts.Count != 0)

{

dgContract.ItemsSource = contracts.ToList();

}

else throw new Exception("Данные не найдены!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

GetData(TbSearch.Text, ((ComboBoxItem)CbSort.SelectedValue).Content.ToString());

}

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

{

NavigationService.Content = new ContractEditPage();

}

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

{

if (dgContract.SelectedItem != null)

{

try

{

var cdwd = dgContract.SelectedItem as Contract;

var choice = MessageBox.Show("Вы точно хотите удалить выбранные данные?", "Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (choice == MessageBoxResult.Yes) {

App.db.Contract.Remove(cdwd);

Листинг 15 - «ContractPage»

App.db.SaveChanges();

dgContract.ItemsSource = App.db.Contract.ToList();

MessageBox.Show("Удаление успешно!", "Готово", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (DbUpdateException ex)

{

MessageBox.Show(ex.Message, "Ошибка!");

}

catch (ArgumentNullException ex)

{

MessageBox.Show(ex.Message, "Ошибка!");

}

}

else MessageBox.Show("Выберите запись которую хотите удалить", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

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

{

GetData(TbSearch.Text, CbSort.SelectedItem.ToString());

}

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

{

try

{

if (dgContract.SelectedItem != null)

{

Contract selected = dgContract.SelectedItem as Contract;

SaveFileDialog sfd = new SaveFileDialog();

string source = $@"{Directory.GetCurrentDirectory()}\Договор.doc";

word.Application app = new word.Application();

word.Document doc = app.Documents.Open(source);

word.Bookmarks wB = doc.Bookmarks;

doc.Activate();

try

{

if (sfd.ShowDialog() == false)

{

doc.Close();

doc = null;

app.Quit();

return;

}

wB["НазваниеКомпании"].Range.Text = selected.Object.Company.Title\_Company;

wB["НазваниеКомпании1"].Range.Text = selected.Object.Company.Title\_Company;

wB["НазваниеКомпании4"].Range.Text = selected.Object.Company.Title\_Company;

wB["НазваниеКомпании5"].Range.Text = selected.Object.Company.Title\_Company;

wB["Директор"].Range.Text = selected.Object.Company.Director;

wB["АдресОбъекта"].Range.Text = selected.Object.Address\_Object;

wB["АдресОбъекта1"].Range.Text = selected.Object.Address\_Object;

Листинг 16 - «ContractPage»

wB["НазначениеОбъекта"].Range.Text = selected.Object.Purpose;

wB["РежимРаботы"].Range.Text = selected.Object.Work\_Mode;

wB["Марки"].Range.Text = selected.Main\_Equipment\_Boiler\_.Stamp;

wB["Модель"].Range.Text = selected.Main\_Equipment\_Boiler\_.Model;

wB["Мощность"].Range.Text = Convert.ToString(selected.Main\_Equipment\_Boiler\_.Power);

wB["ОбщаяМощность"].Range.Text = Convert.ToString(selected.Main\_Equipment\_Boiler\_.PowerSumm);

wB["КоличествоКотлов"].Range.Text = Convert.ToString(selected.Main\_Equipment\_Boiler\_.Quantity);

wB["Цена"].Range.Text = String.Format("{0:0 руб.}", selected.Price); ;

wB["НомерДоговора"].Range.Text = Convert.ToString(selected.ID\_Contract);

wB["НомерДоговора1"].Range.Text = Convert.ToString(selected.ID\_Contract);

wB["ДатаЗаключения"].Range.Text = selected.Date\_Conclusion.ToShortDateString();

wB["ДатаЗаключения1"].Range.Text = selected.Date\_Conclusion.ToShortDateString();

wB["ДействуетПо"].Range.Text = Convert.ToString(selected.Valid\_For);

doc.SaveAs2(sfd.FileName);

doc.Close();

doc = null;

app.Quit();

MessageBox.Show("Файл успешно создан");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

doc.Close();

doc = null;

app.Quit();

}

}

else throw new Exception("Запись не выбрана,выберите запись и повторите попытку!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void dgContract\_LoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

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

{

try

{

if (DpStart.SelectedDate == null && DpFinish.SelectedDate == null)

{

Листинг 17 - «ContractPage»

MessageBox.Show("Введены некорректные данные или остались незаполненные поля!", "Оповещение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

else

{

SaveFileDialog sfd = new SaveFileDialog();

excel.Application app = new excel.Application();

excel.Workbook wB;

excel.Worksheet wS;

wB = app.Workbooks.Add();

wS = (excel.Worksheet)wB.Worksheets.get\_Item(1);

wS.Cells[4, 1] = "Номер договора";

wS.Cells[4, 2] = "Дата подписания";

wS.Cells[4, 3] = "Дата окончания действия";

wS.Cells[4, 4] = "Название объекта";

wS.Cells[4, 5] = "Название компании";

wS.Cells[4, 6] = "Директор компании";

wS.Cells[4, 7] = "Стоимость";

try

{

if (sfd.ShowDialog() == false)

{

wB.Close();

wB = null;

app.Quit();

return;

}

List<Contract> contracts = App.db.Contract.Where(c => c.Date\_Conclusion >= DpStart.DisplayDate && c.Date\_Conclusion <= DpFinish.DisplayDate).ToList();

dgContract.ItemsSource = contracts;

for (int i = 0; i < contracts.Count; i++)

{

wS.Cells[i + 5, 1].Value = contracts[i].ID\_Contract;

wS.Cells[i + 5, 2].Value = contracts[i].Date\_Conclusion;

wS.Cells[i + 5, 3].Value = contracts[i].Valid\_For;

wS.Cells[i + 5, 4].Value = contracts[i].Object.Title\_Object;

wS.Cells[i + 5, 5].Value = contracts[i].Object.Company.Title\_Company;

wS.Cells[i + 5, 6].Value = contracts[i].Object.Company.Director;

wS.Cells[i + 5, 7].Value = contracts[i].Price;

}

wS.Columns.AutoFit();

app.Application.ActiveWorkbook.SaveAs(sfd.FileName);

MessageBox.Show("Файл успешно сохранен");

app.Quit();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

app.Quit();

}

}

}

catch (Exception)

{

throw;

}

}

Листинг 18 - «ContractPage»

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

{

try

{

int count = 0;

var app = new excel.Application();

try

{

SaveFileDialog svd = new SaveFileDialog();

svd.Title = "Сохранить как...";

svd.Filter = "Excel file(\*.xlsx)|.\*xslx";

svd.OverwritePrompt = true;

svd.CheckPathExists = true;

if (svd.ShowDialog() == false)

{

app.Quit();

}

else

{

List<Contract> specializationList = App.db.Contract.ToList();

app.Workbooks.Add(Type.Missing);

var worksheet = (excel.Worksheet)app.Worksheets[1];

count = specializationList.Count;

for (int i = 0; i < count; i++)

{

worksheet.Cells[i + 1, 1] = specializationList[i].Date\_Conclusion;

worksheet.Cells[i + 1, 2] = specializationList[i].Price;

}

worksheet.Columns.AutoFit();

var xlCharts = (excel.ChartObjects)worksheet.ChartObjects(Type.Missing);

var myChart = (excel.ChartObject)xlCharts.Add(110, 0, 350, 250);

var chart = myChart.Chart;

var seriesCollection = (excel.SeriesCollection)chart.SeriesCollection(Type.Missing);

var series = seriesCollection.NewSeries();

series.XValues = worksheet.get\_Range("A1", "A" + count);

series.Values = worksheet.get\_Range("B1", "B" + count);

chart.ChartType = excel.XlChartType.xl3DColumn;

app.Application.ActiveWorkbook.SaveAs(svd.FileName);

app.Quit();

MessageBox.Show("Отчет сформирован!", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

app.Quit();

}

}

catch (Exception)

{

throw;

}

}

Листинг 19 - «ContractPage»

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

{

try

{

if (DpStart.SelectedDate == null && DpFinish.SelectedDate == null)

{

MessageBox.Show("Введены некорректные данные или остались незаполненные поля!", "Оповещение", MessageBoxButton.OK, MessageBoxImage.Warning);

}else

if (DpStart.SelectedDate >= DpFinish.SelectedDate)

{

MessageBox.Show("Дата окончания не может быть раньше даты началы или равна ей,исправьте данные и повторите попытку", "Оповещение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

else dgContract.ItemsSource = App.db.Contract.Where(c => c.Date\_Conclusion >= DpStart.DisplayDate && c.Date\_Conclusion <= DpFinish.DisplayDate).ToList();

}

catch

{

MessageBox.Show("Введены некорректные данные или остались незаполненные поля!", "Оповещение", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

Листинг 20 - «ContractPage»

Листинг логики «EquipmentEditPage» представлен в листингах 21-22.

public EquipmentEditPage()

{

InitializeComponent();

}

private void OnlyNumbers(object sender, TextCompositionEventArgs e)

{

if (!Char.IsDigit(e.Text, 0))

{

e.Handled = true;

}

}

private void OnlyLetters(object sender, TextCompositionEventArgs e)

{

Regex inp = new Regex("^[А-Я]+", RegexOptions.IgnoreCase);

Regex ing = new Regex("^[A-Z]+", RegexOptions.IgnoreCase);

if (!inp.IsMatch(e.Text) && !ing.IsMatch(e.Text))

e.Handled = true;

}

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

{

try

{

var choice = MessageBox.Show("Вы точно хотите добавить данное оборудование ? ", "Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (choice == MessageBoxResult.Yes)

{

if (TbUnitOfMeasurement.Text != null && TbStamp.Text != null && TbService.Text != null && TbQuantity.Text != "" && TbPower.Text != "" && TbModel.Text != null )

{

Листинг 21 - «EquipmentEditPage»

Main\_Equipment\_Boiler\_ newEquipment = new Main\_Equipment\_Boiler\_();

newEquipment.ID\_Equipment = App.db.Main\_Equipment\_Boiler\_.Max(c => c.ID\_Equipment) + 1;

newEquipment.Model = TbModel.Text;

newEquipment.Stamp = TbStamp.Text;

newEquipment.Power = Convert.ToInt32(TbPower.Text);

newEquipment.Quantity = Convert.ToInt32(TbQuantity.Text);

newEquipment.Unit\_Of\_Measurement = TbUnitOfMeasurement.Text;

newEquipment.Service = TbService.Text;

App.db.Main\_Equipment\_Boiler\_.Add(newEquipment);

App.db.SaveChanges();

MessageBox.Show("Оборудование добавлено!", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.Content = new EquipmentPage();

}

else MessageBox.Show("Введены некорректные данные или остались незаполненные поля!", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

Листинг 22 - «EquipmentEditPage»

Листинг логики «EquipmentPage» представлен в листингах 23-25.

public EquipmentPage()

{

InitializeComponent();

dgEquipment.ItemsSource = App.db.Main\_Equipment\_Boiler\_.ToList();

CbSort.SelectedIndex = 0;

if (App.workEmployee.Post == "Технический инспектор проектного отдела")

{

BtnAdd.Visibility = Visibility.Hidden;

BtnDel.Visibility = Visibility.Hidden;

tbAction.Visibility = Visibility.Hidden;

}

}

private void GetData(string search = "", string Filter = "")

{

try

{

var equipments = App.db.Main\_Equipment\_Boiler\_.ToList();

if (!String.IsNullOrEmpty(search) && !String.IsNullOrWhiteSpace(search))

{

equipments = equipments.Where(c => c.Stamp.ToString().Contains(search.ToLower())).ToList();

}

if (!String.IsNullOrEmpty(Filter) && !String.IsNullOrWhiteSpace(Filter)){

if (Filter == "Моделе")

{

equipments = equipments.ToList().OrderBy(c => c.Model).ToList();

Листинг 23 - «EquipmentPage»

}

else if (Filter == "Мощности")

{

equipments = equipments.ToList().OrderBy(c => c.Power).ToList();

}

else if (Filter == "Сумманой мощности")

{

equipments = equipments.ToList().OrderBy(c => c.PowerSumm).ToList();

}

else if (Filter == "Кол-ву")

{

equipments = equipments.ToList().OrderBy(c => c.Quantity).ToList();

}

}

if (equipments.Count != 0)

{

dgEquipment.ItemsSource = equipments.ToList();

}

else throw new Exception("Данные не найдены!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

private void dgEquipment\_LoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

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

{

NavigationService.Content = new EquipmentEditPage();

}

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

{

if (dgEquipment.SelectedItem != null)

{

try

{

var cdwd = dgEquipment.SelectedItem as Main\_Equipment\_Boiler\_;

var choice = MessageBox.Show("Вы точно хотите удалить выбранные данные?", "Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (choice == MessageBoxResult.Yes)

{

App.db.Main\_Equipment\_Boiler\_.Remove(cdwd);

App.db.SaveChanges();

dgEquipment.ItemsSource = App.db.Main\_Equipment\_Boiler\_.ToList();

MessageBox.Show("Удаление успешно!", "Готово", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (DbUpdateException ex){

Листинг 24 - «EquipmentPage»

MessageBox.Show(ex.Message, "Ошибка!");

}

catch (ArgumentNullException ex)

{

MessageBox.Show(ex.Message, "Ошибка!");

}

}

else MessageBox.Show("Выберите запись которую хотите удалить", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

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

{

GetData(TbSearch.Text, ((ComboBoxItem)CbSort.SelectedValue).Content.ToString());

}

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

{

GetData(TbSearch.Text, CbSort.SelectedItem.ToString());

}

}

Листинг 25 - «EquipmentPage»

Листинг логики «ObjectEditPage» представлен в листингах 26-27.

public ObjectEditPage()

{

InitializeComponent();

CbTitleCompany.ItemsSource = App.db.Company.ToList();

}

private void OnlyLetters(object sender, TextCompositionEventArgs e)

{

Regex inp = new Regex("^[А-Я]+", RegexOptions.IgnoreCase);

Regex ing = new Regex("^[A-Z]+", RegexOptions.IgnoreCase);

if (!inp.IsMatch(e.Text) && !ing.IsMatch(e.Text))

e.Handled = true;

}

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

{

try

{

var choice = MessageBox.Show("Вы точно хотите добавить данный объект ? ", "Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (choice == MessageBoxResult.Yes)

{

if (TbAddressObject.Text != null && TbPurpose.Text != null && TbTitleObject.Text != null && TbWorkMode.Text != null && CbTitleCompany.Text != null )

{

Models\_BD.Object newObject = new Models\_BD.Object();

newObject.ID\_Object = App.db.Object.Max(c => c.ID\_Object) + 1;

newObject.Address\_Object = TbAddressObject.Text;

newObject.Purpose = TbPurpose.Text;

newObject.Title\_Object = TbTitleObject.Text;

newObject.Work\_Mode = TbWorkMode.Text;

newObject.ID\_Company = (CbTitleCompany.SelectedItem as Company).ID\_Company;

App.db.Object.Add(newObject);

Листинг 26 - «ObjectEditPage»

App.db.SaveChanges();

MessageBox.Show("Объект добавлен!", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.Content = new ObjectPage();

}

else MessageBox.Show("Введены некорректные данные или остались незаполненные поля!", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

Листинг 27 - «ObjectEditPage»

Листинг логики «ObjectPage» представлен в листингах 28-30.

public ObjectPage()

{

InitializeComponent();

dgObject.ItemsSource = App.db.Object.ToList();

CbSort.SelectedIndex = 0;

if (App.workEmployee.Post == "Технический инспектор проектного отдела")

{

BtnAdd.Visibility = Visibility.Hidden;

BtnDel.Visibility = Visibility.Hidden;

tbAction.Visibility = Visibility.Hidden;

}

}

private void GetData(string search = "", string Filter = "")

{

try

{

var objects = App.db.Object.ToList();

if (!String.IsNullOrEmpty(search) && !String.IsNullOrWhiteSpace(search))

{

objects = objects.Where(c => c.Address\_Object.ToString().Contains(search.ToLower())).ToList();

}

if (!String.IsNullOrEmpty(Filter) && !String.IsNullOrWhiteSpace(Filter))

{

if (Filter == "Названию")

{

objects = objects.ToList().OrderBy(c => c.Title\_Object).ToList();

}

else if (Filter == "Назначению")

{

objects = objects.ToList().OrderBy(c => c.Purpose).ToList();

}

else if (Filter == "Адресу")

{

objects = objects.ToList().OrderBy(c => c.Address\_Object).ToList();

}

else if (Filter == "Режиму работы")

Листинг 28 - «ObjectPage»

{

objects = objects.ToList().OrderBy(c => c.Work\_Mode).ToList();

}

else if (Filter == "Компании")

{

objects = objects.ToList().OrderBy(c => c.Company.Title\_Company).ToList();

}

}

if (objects.Count != 0)

{

dgObject.ItemsSource = objects.ToList();

}

else throw new Exception("Данные не найдены!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

GetData(TbSearch.Text, CbSort.SelectedItem.ToString());

}

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

{

GetData(TbSearch.Text, ((ComboBoxItem)CbSort.SelectedValue).Content.ToString());

}

private void dgObject\_LoadingRow(object sender, DataGridRowEventArgs e)

{

e.Row.Header = (e.Row.GetIndex() + 1).ToString();

}

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

{

NavigationService.Content = new ObjectEditPage();

}

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

{

if (dgObject.SelectedItem != null)

{

try

{

var cdwd = dgObject.SelectedItem as Models\_BD.Object;

var choice = MessageBox.Show("Вы точно хотите удалить выбранные данные?", "Внимание!", MessageBoxButton.YesNo, MessageBoxImage.Question);

if (choice == MessageBoxResult.Yes)

{

App.db.Object.Remove(cdwd);

App.db.SaveChanges();

dgObject.ItemsSource = App.db.Object.ToList();

MessageBox.Show("Удаление успешно!", "Готово", MessageBoxButton.OK, MessageBoxImage.Information);

}}

Листинг 29 - «ObjectPage»

catch (DbUpdateException ex)

{

MessageBox.Show(ex.Message, "Ошибка!");

}

catch (ArgumentNullException ex)

{

MessageBox.Show(ex.Message, "Ошибка!");

}

}

else MessageBox.Show("Выберите запись которую хотите удалить", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

Листинг 30 - «ObjectPage»

## 2.5.Разработка тестовых данных

Информация по тесту №1 представлена рисунке 21.

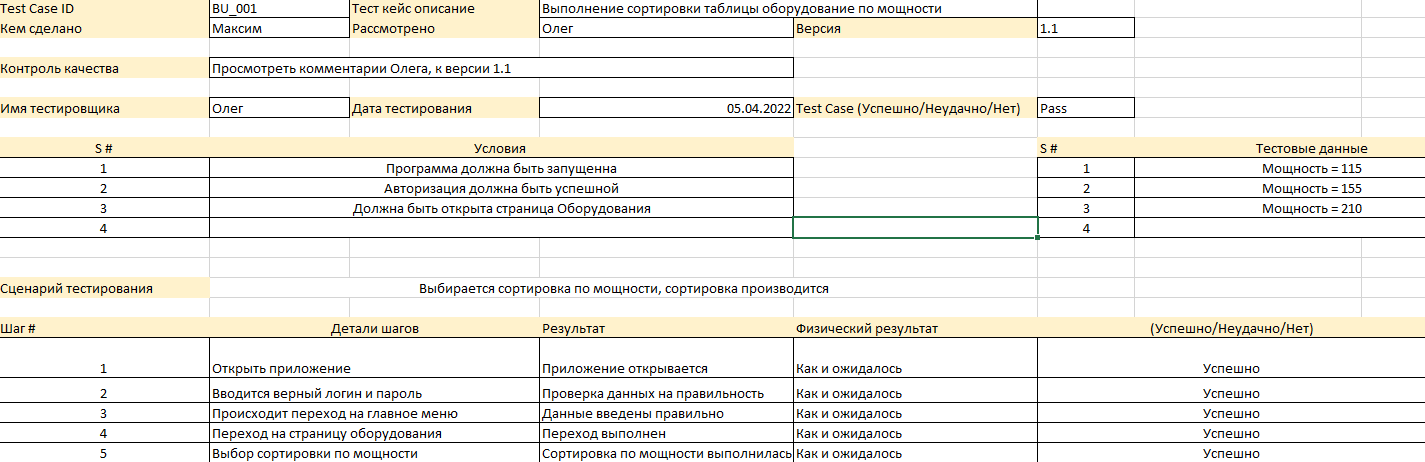


Рис. 21 – Тест № 1

Информация по тесту №2 представлена рисунке 22.

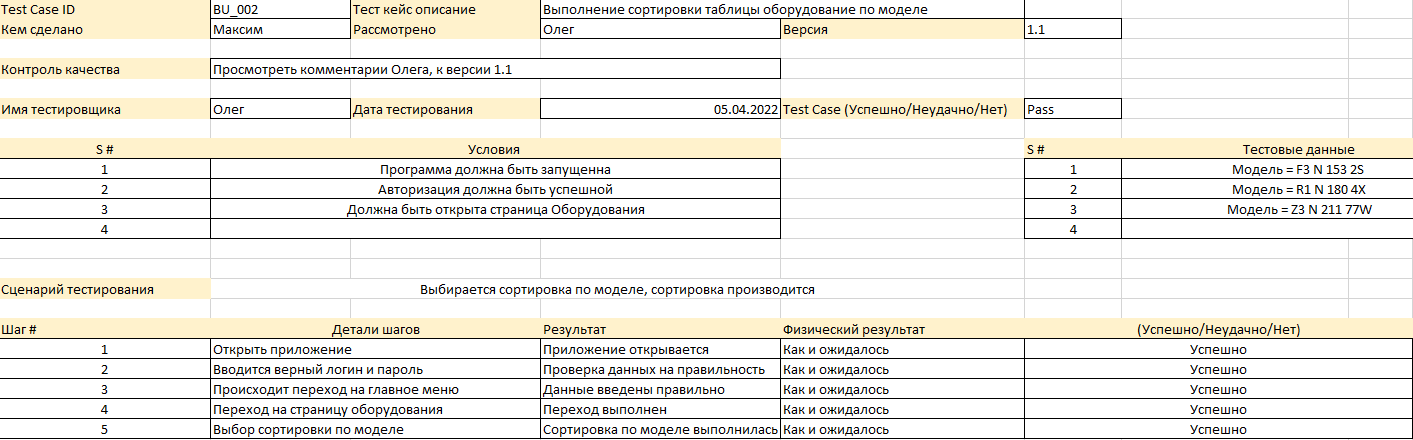


Рис. 22 – Тест № 2

Информация по тесту №3 представлена рисунке 23.

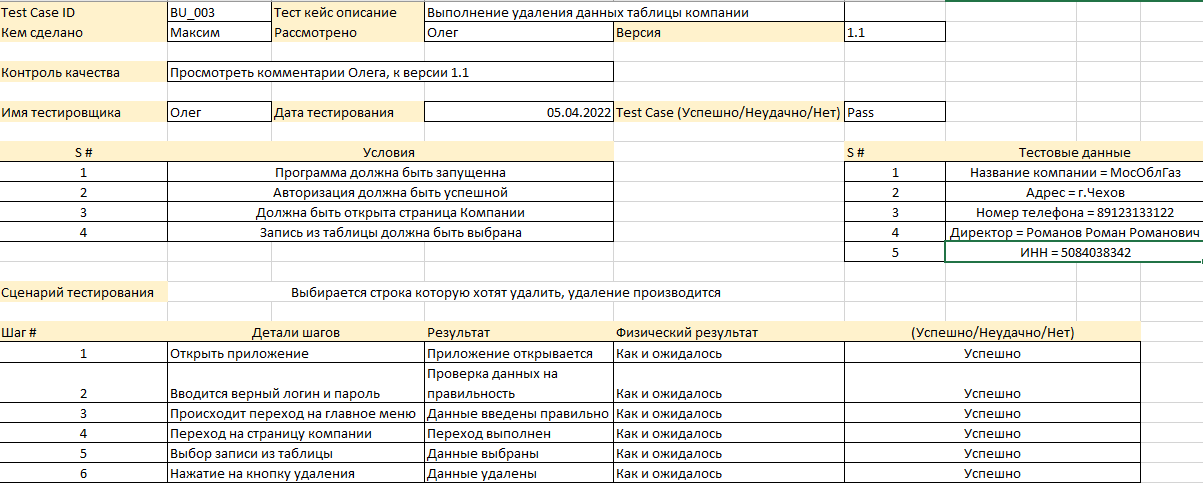


Рис. 23 – Тест № 3

Информация по тесту №4 представлена рисунке 24.

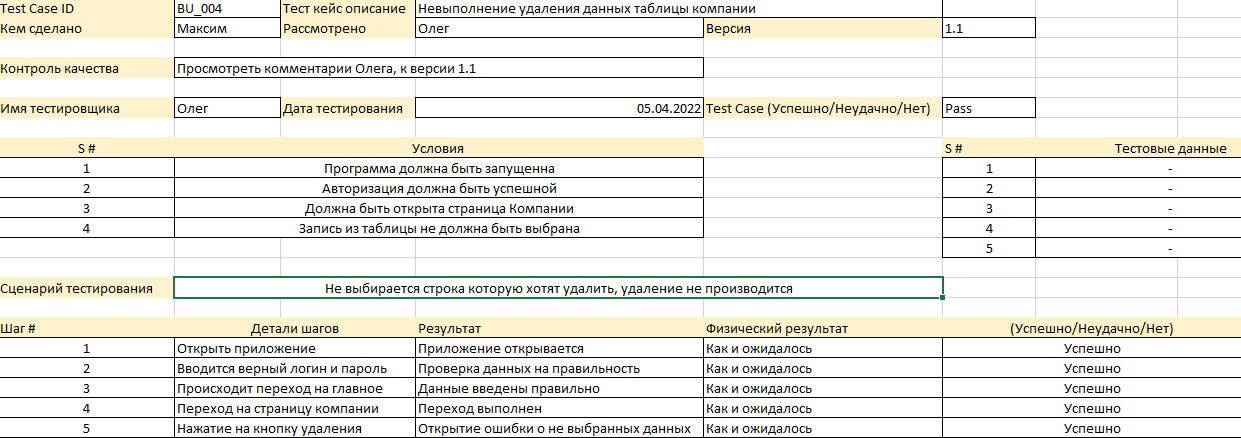


Рис. 24 – Тест № 4

Информация по тесту №5 представлена рисунке 25.

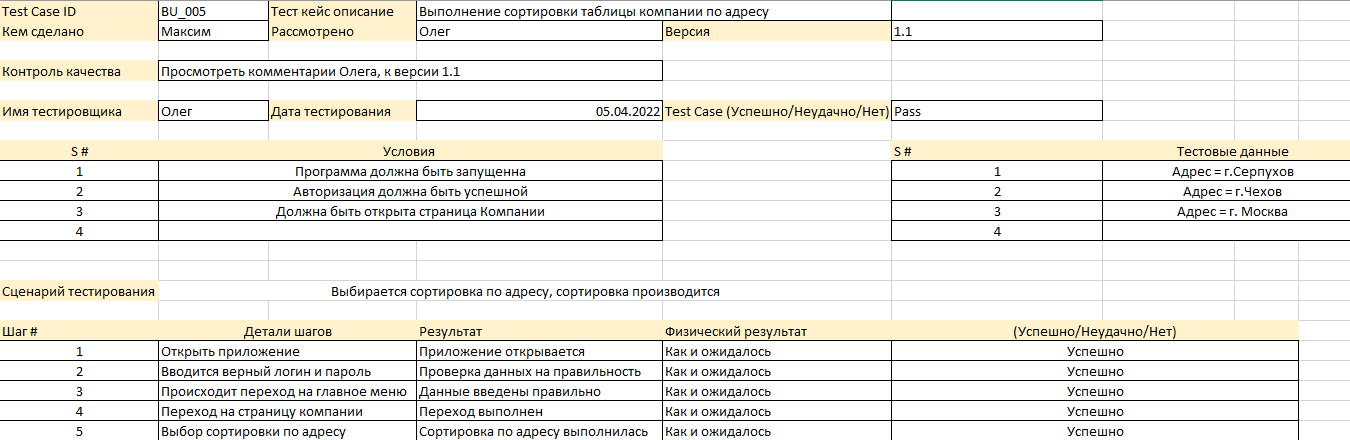


Рис. 25 – Тест № 5

Информация по тесту №6 представлена рисунке 26.

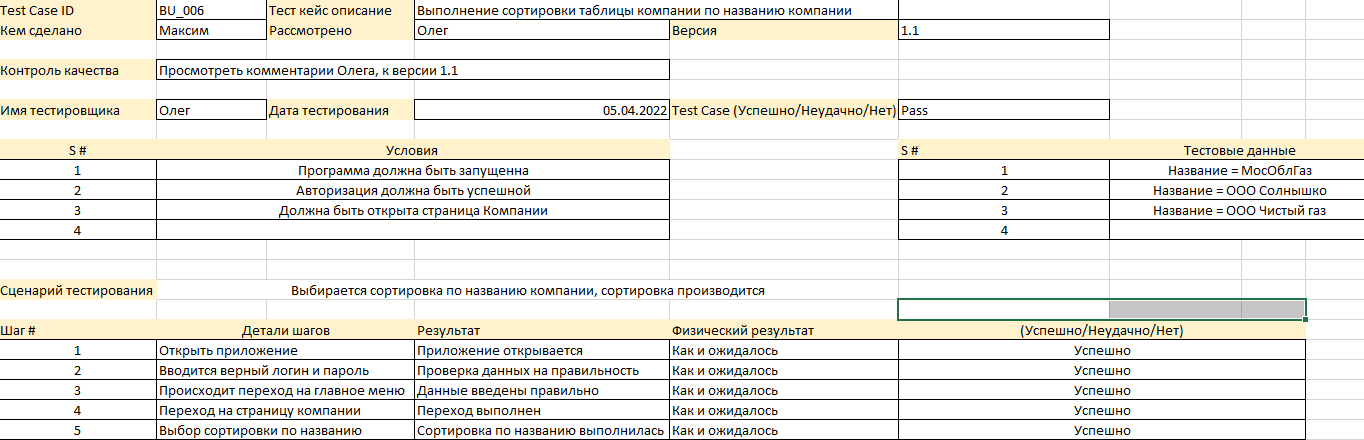


Рис. 26 – Тест № 6

Информация по тесту №7 представлена рисунке 27.

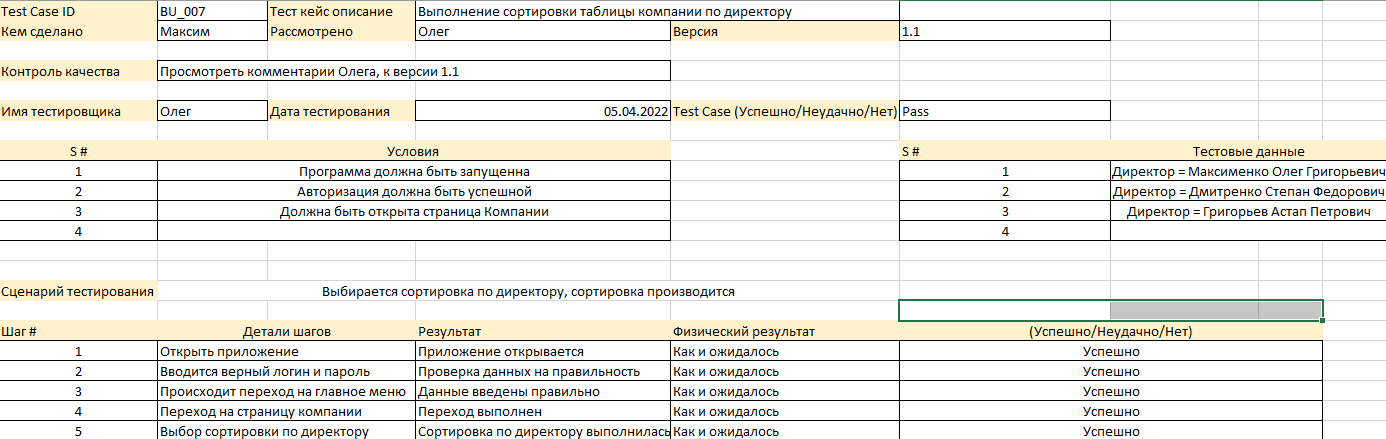


Рис. 27 – Тест № 7

Информация по тесту №8 представлена рисунке 28.

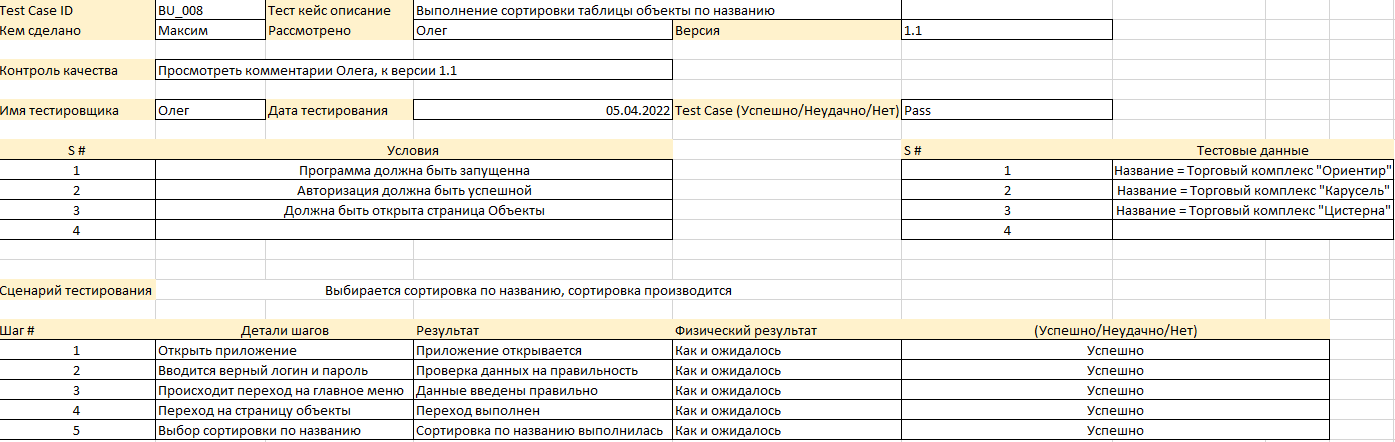


Рис. 28 – Тест № 8

Информация по тесту №9 представлена рисунке 29.

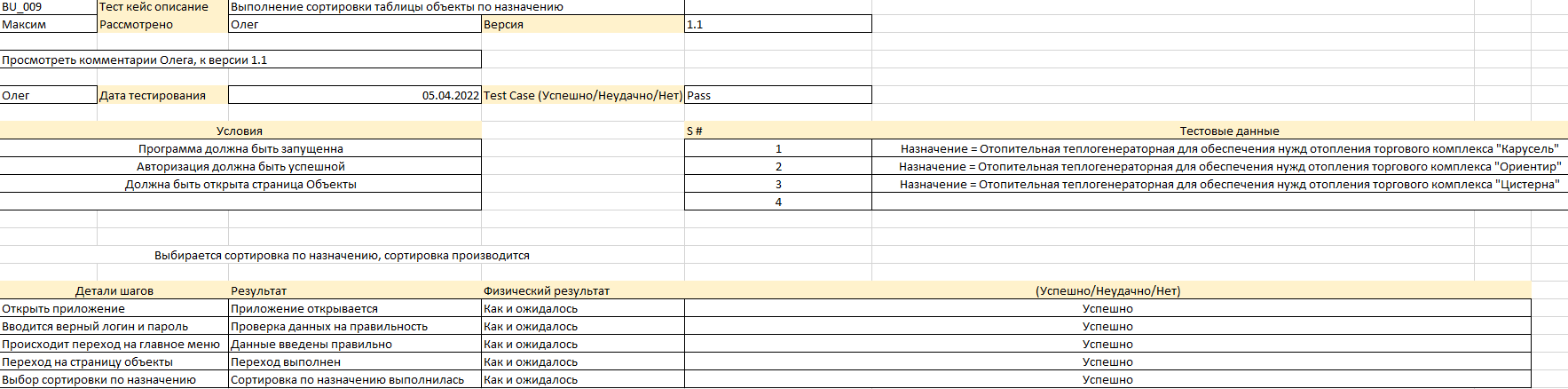


Рис. 29 – Тест № 9

Информация по тесту №10 представлена рисунке 30.

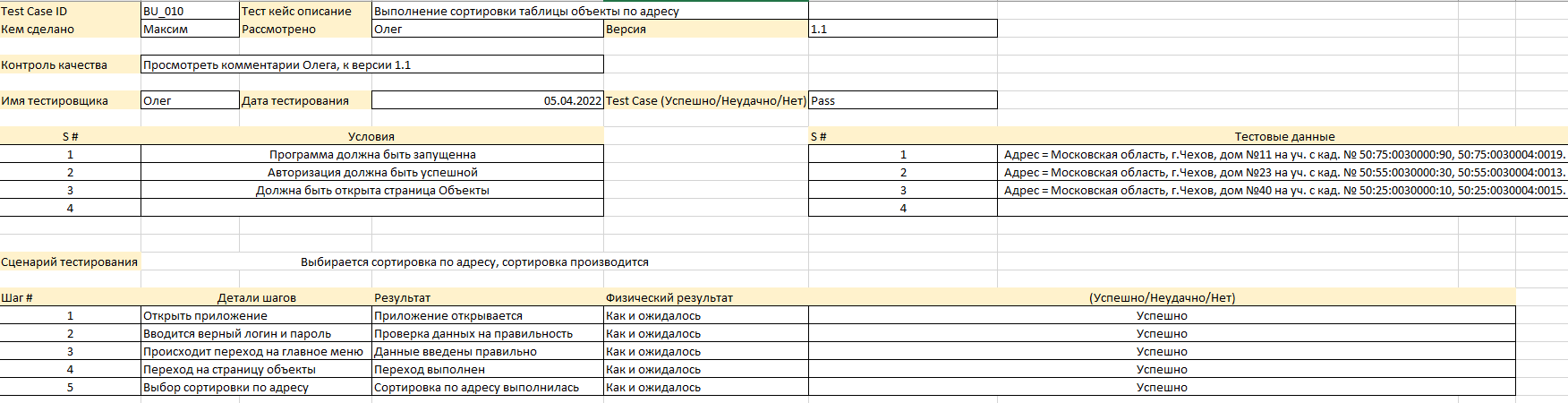


Рис. 30 – Тест № 10

Информация по тесту №11 представлена рисунке 31.

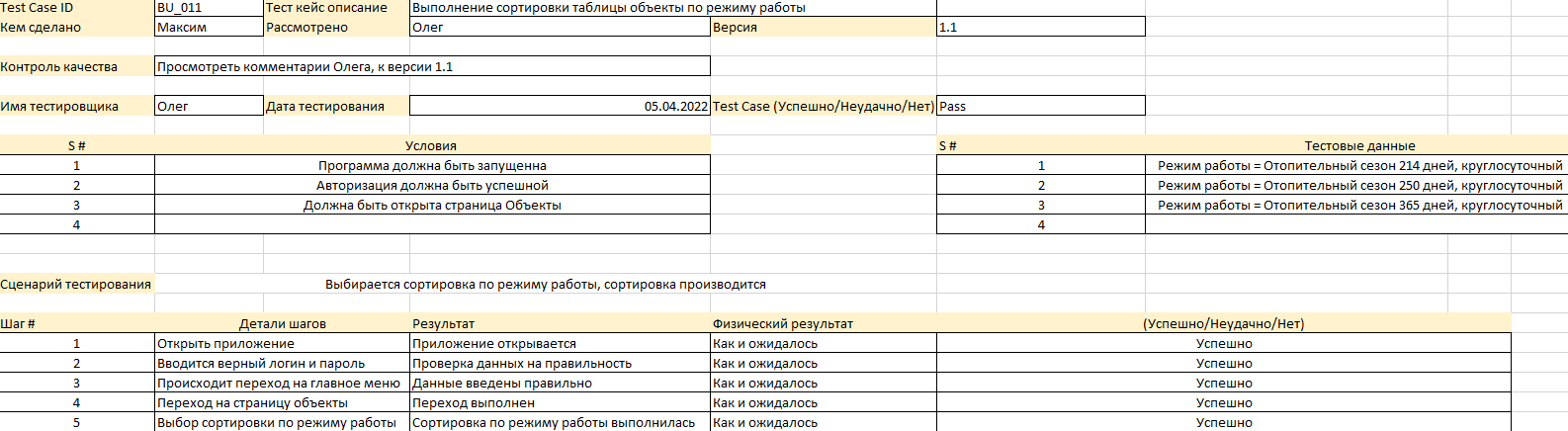


Рис. 31 – Тест № 11

Информация по тесту №12 представлена рисунке 32.

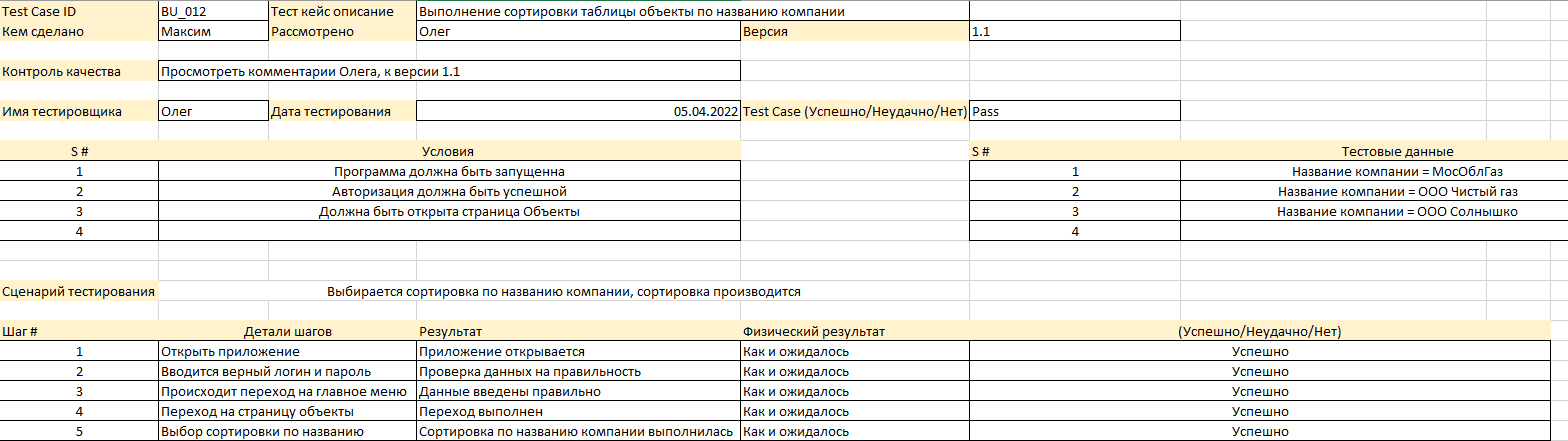


Рис. 32 – Тест № 12

Информация по тесту №13 представлена рисунке 33.

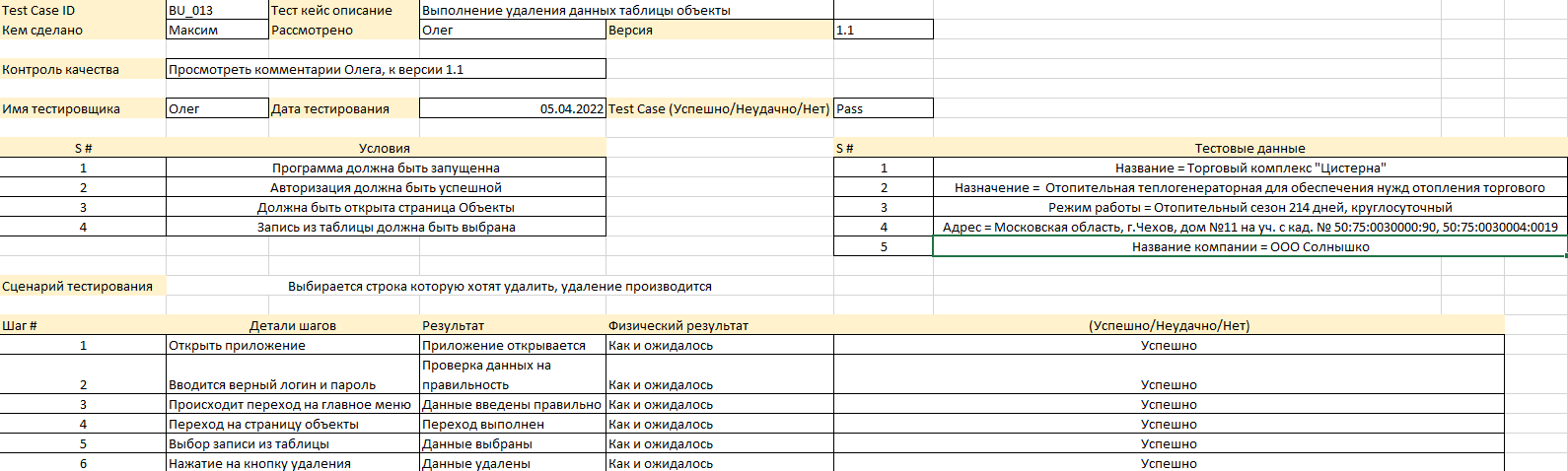


Рис. 33 – Тест № 13

Информация по тесту №14 представлена рисунке 34.

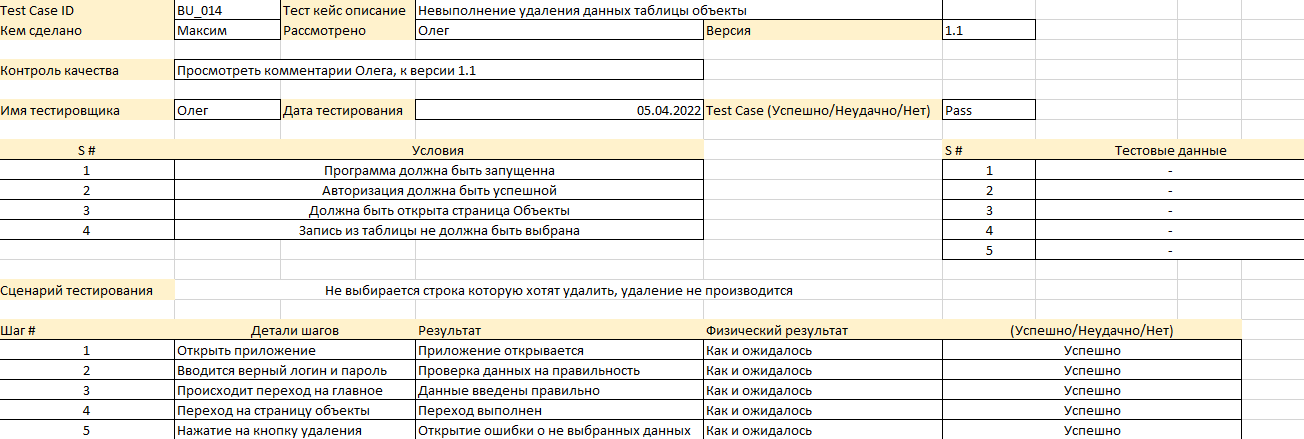


Рис. 34 – Тест № 14

Информация по тесту №15 представлена рисунке 35.

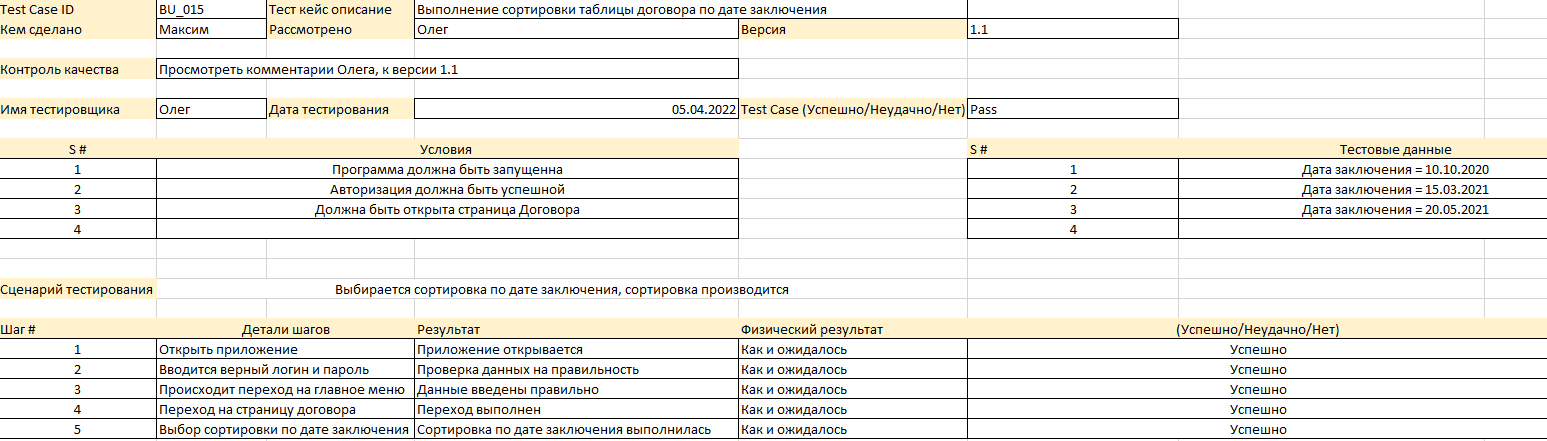


Рис. 15 – Тест № 35

Информация по тесту №16 представлена рисунке 36.

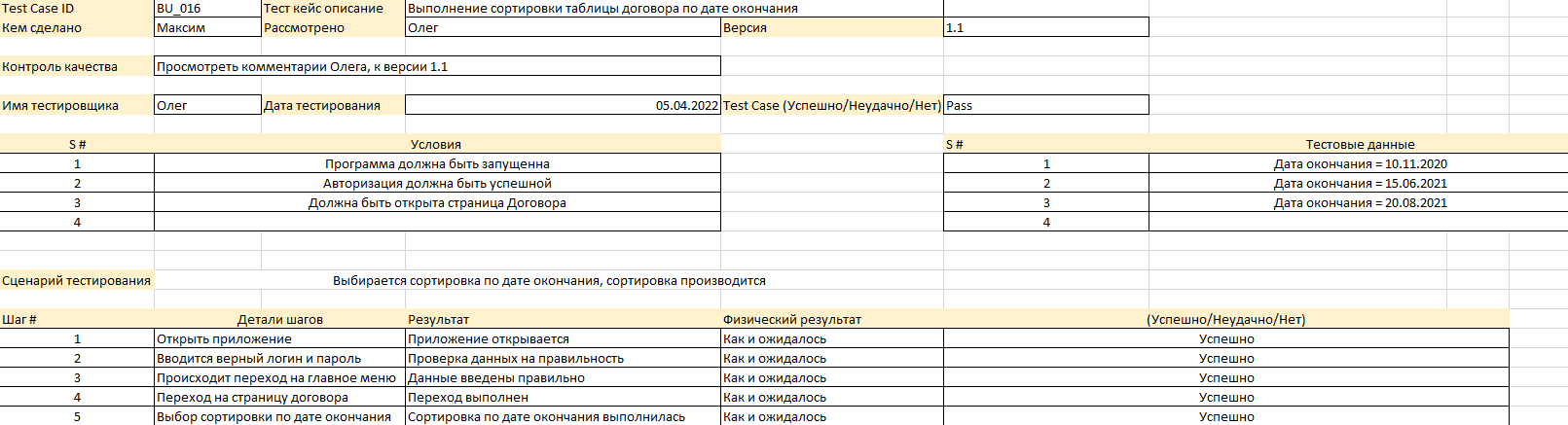


Рис. 36 – Тест № 16

Информация по тесту №17 представлена рисунке 37.

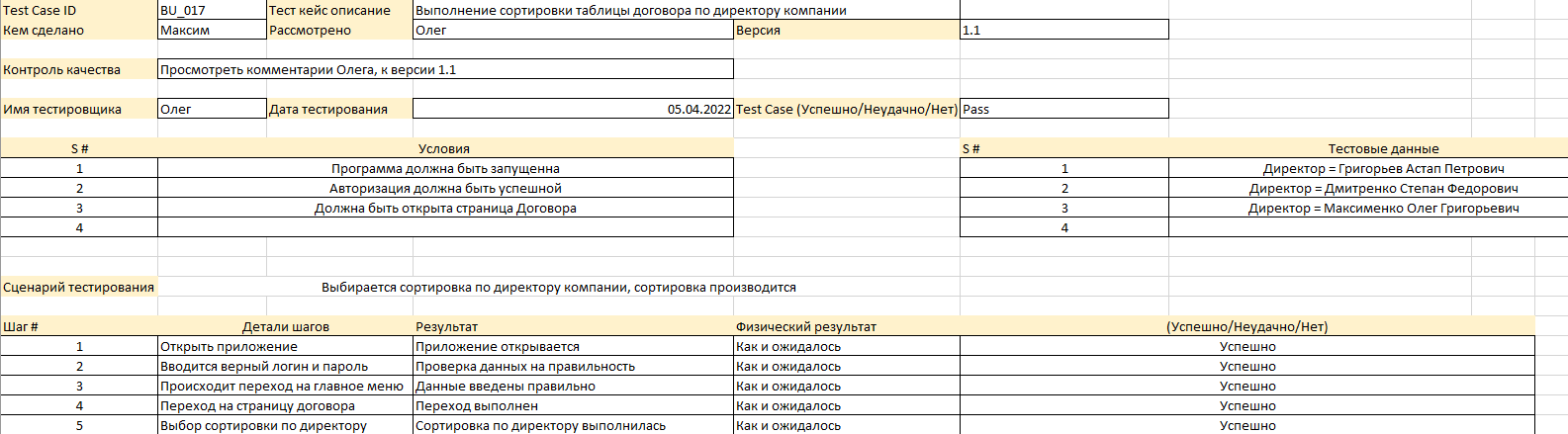


Рис. 37 – Тест № 17

Информация по тесту №18 представлена рисунке 38.

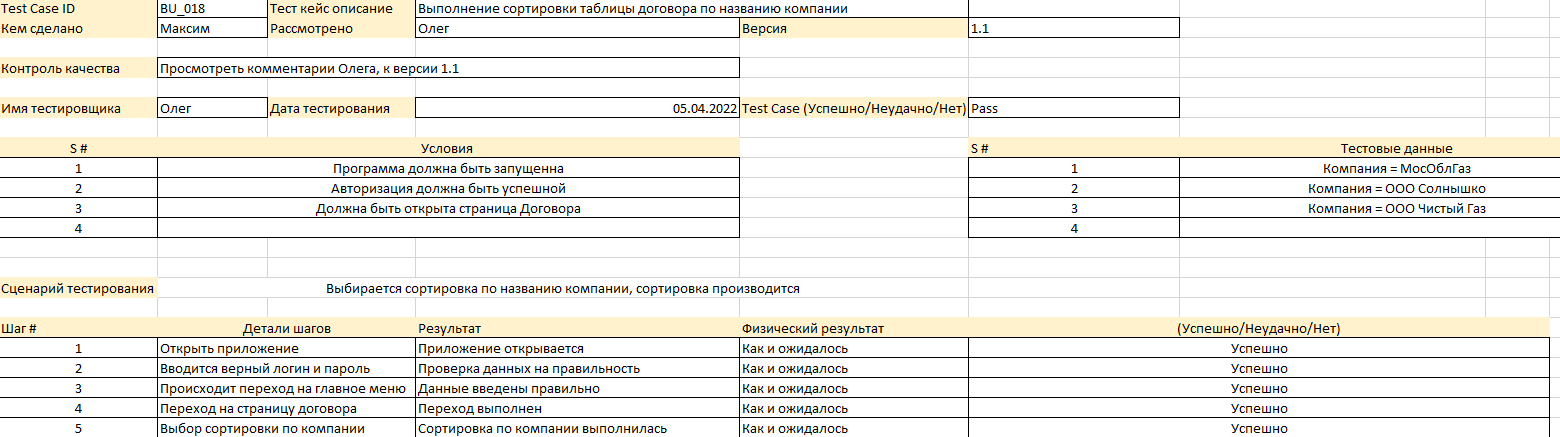


Рис. 38 – Тест № 18

Информация по тесту №19 представлена рисунке 39.

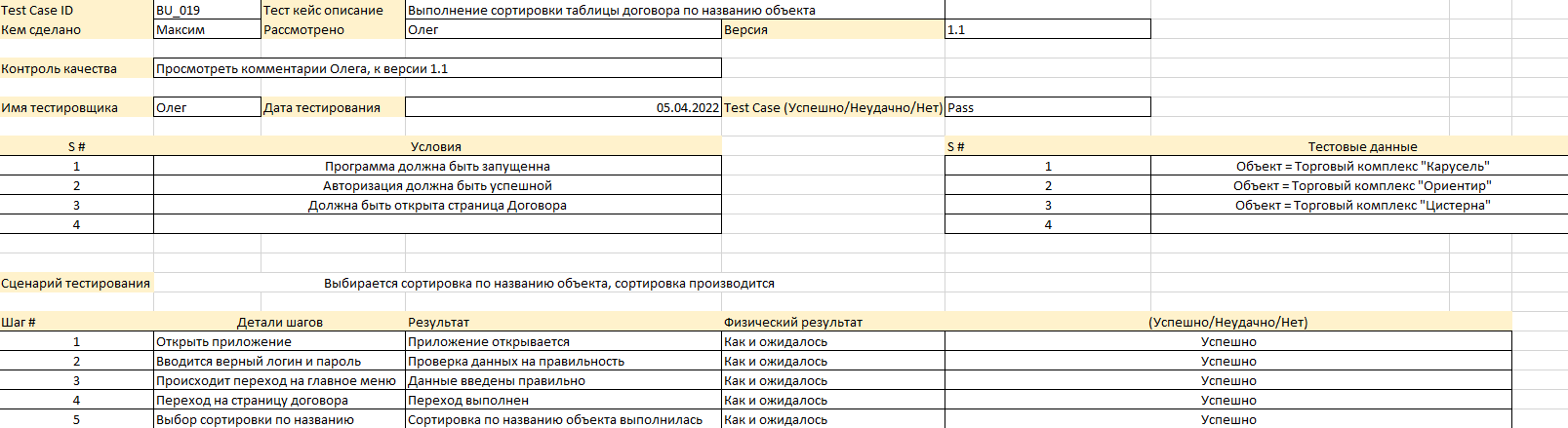


Рис. 39 – Тест № 19

Информация по тесту №20 представлена рисунке 40.

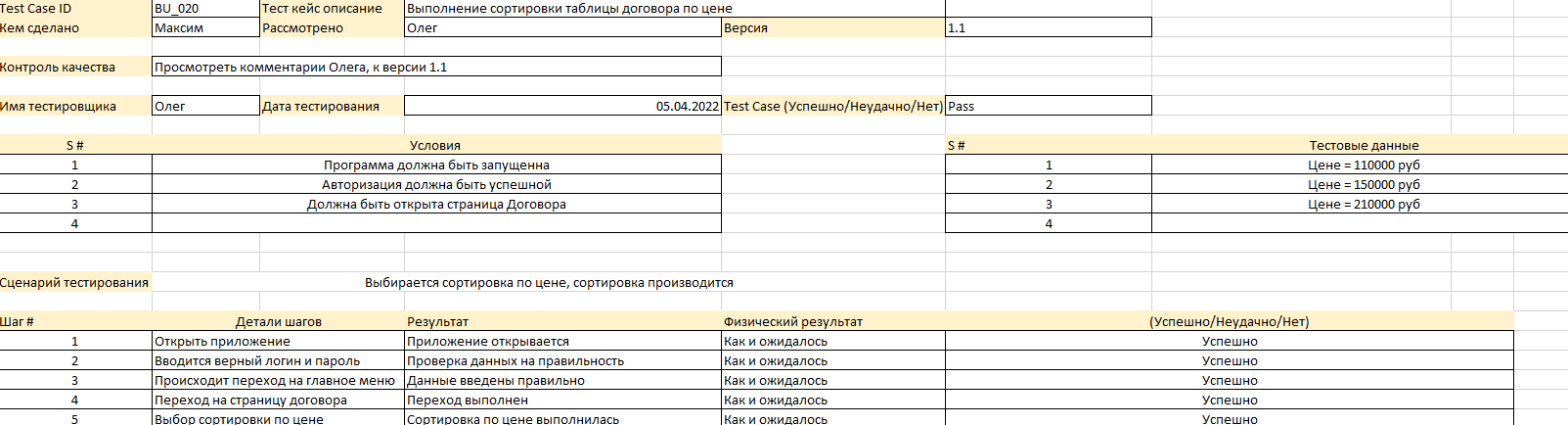


Рис. 40 – Тест № 20

Результат выполнения тестов представлен на рисунке 41

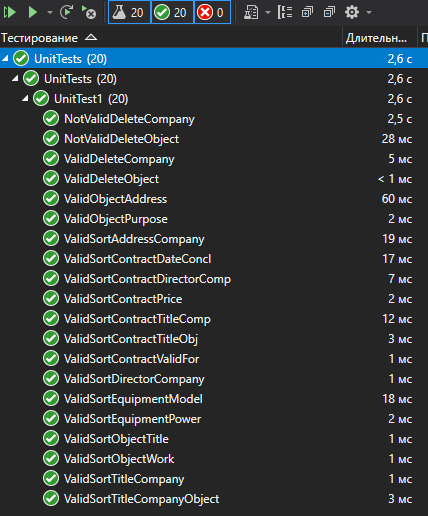


Рис. 41 – результат выполнения тестов

## 2.6. Результат работы программы

Результат работы программы представлен на рисунках 42-45

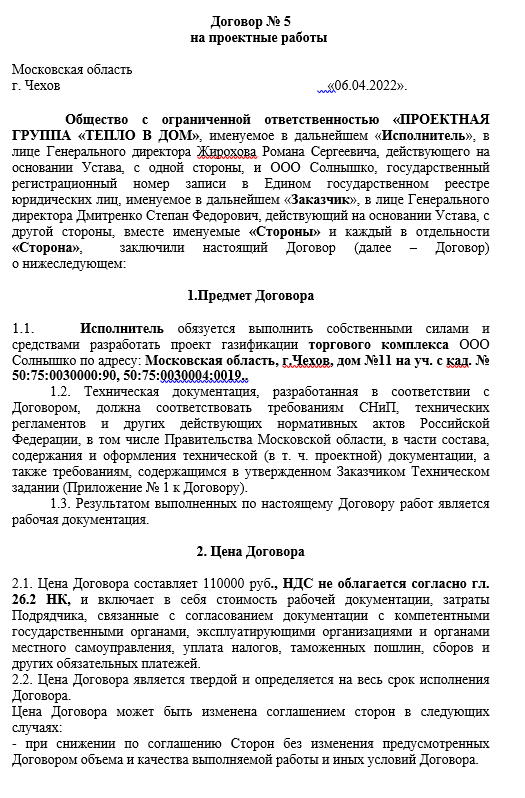


Рис. 42 – результат работы программы

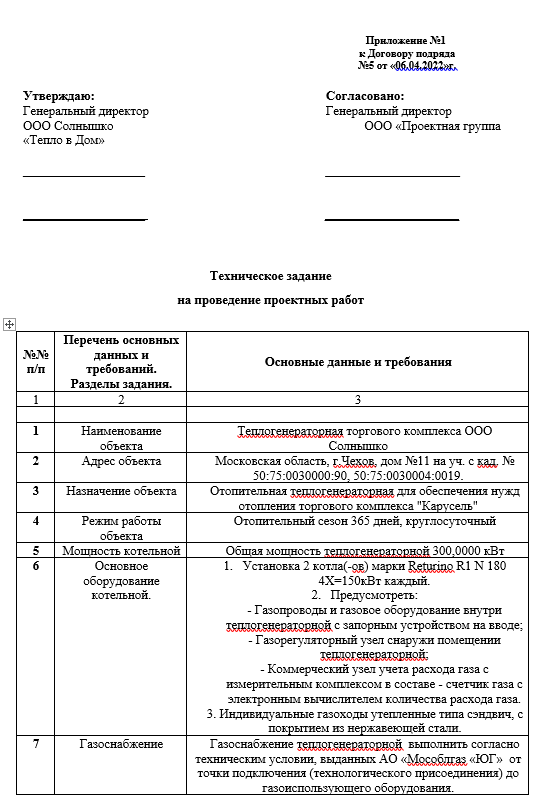


Рис. 43 – результат работы программы

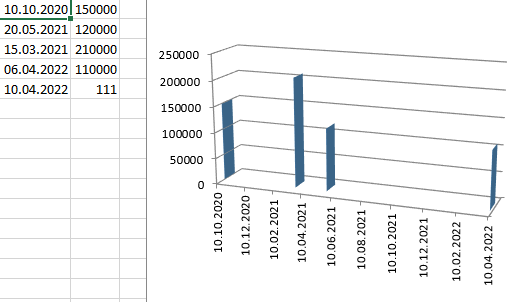


Рис. 44 – результат работы программы

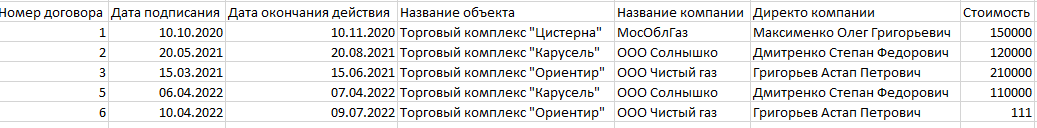


Рис. 45 – результат работы программы

## 2.7. Руководство по использования программы

Руководства

Руководство администратора

Для работы с базами данных необходимо установить приложение установить программу для управления базами данных на сервере. Поэтому надо установить SQL Server Management Studio 19. Все необходимые материалы для установки можно найти по адресу <https://docs.microsoft.com/ru-ru/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>.

Присоединение базы данных:

1. В SQL Server Management Studio обозревателе объектов Компонент SQL Server Database Engine подключитесь к экземпляру компонента и разверните его представление в SSMS.
2. Щелкните правой кнопкой мыши узел «**Базы данных»** и выберите команду «**Присоединить»**.

Чтобы указать присоединяемую базу данных, в диалоговом окне «**Присоединение баз данных»** нажмите кнопку «**Добавить»**, в диалоговом окне «**Расположение файлов базы данных»** выберите диск, на котором находится база данных, и разверните дерево каталогов, чтобы найти и выбрать MDF-файл.

Таблица 23 – Сообщение системному программисту

|  |  |
| --- | --- |
| Сообщение | Описание действий при получении сообщения |
| Попытка соединения не удалась | Проверьте работу сервера |
| Проверьте подключение или повторите попытку позже |

### Руководство программиста

Данная программа предназначена для автоматизации учета закупленного оборудования.

Программа предусматривает авторизацию для двух ролей «Менеджер» и «Технический инспектор проектного отдела»

Для корректной работы данной программы требуется следующее аппаратное и программное обеспечение:

* процессор AMD A8-5500 или совместимый с ним;
* 4 Гб оперативной памяти;
* видеоадаптер, поддерживаемый Windows;
* операционная система: Windows 7, Windows 8, Windows 10;
* Microsoft Word 2007 или выше.

Настройка программы.

Для начала установки программы вам необходимо установить Visual Studio 2019. Скачать установочные файлы можно на официальном сайте Microsoft: https://visualstudio.microsoft.com/ru/downloads/. Далее нужно запустить установочный файл Setup. Следуйте инструкции мастера установки. Выберите нужный путь расположения файлов на вашем компьютере, либо установите в уже заданную директорию по умолчанию.

Важно: не забудьте куда вы установили программу!

После успешной установки, в заданной директории при установке, вы найдете файлы приложения и саму программу. Файлы приложения можно оставить там же или перенести их в стандартную директорию файлов Visual Studio 2019.

Внимание: обычно все файлы Visual Studio 2019 находятся в директории: C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\Common7\IDE\

Таблица 24 – Сообщения программисту

|  |  |
| --- | --- |
| Сообщение | Описание действий при получении сообщения |
| Ошибка ввода логина и пароля | Неверный логин или пароль |
| Введите логин или пароль |
| Файл создан | Файл успешно создан! |
| Договор удален | Договор успешно удален! |

### Руководство оператора

Запускаем приложение. В открывшемся окне необходимо авторизоваться. Интерфейс окна «Авторизация» представлен на рисунке 46.

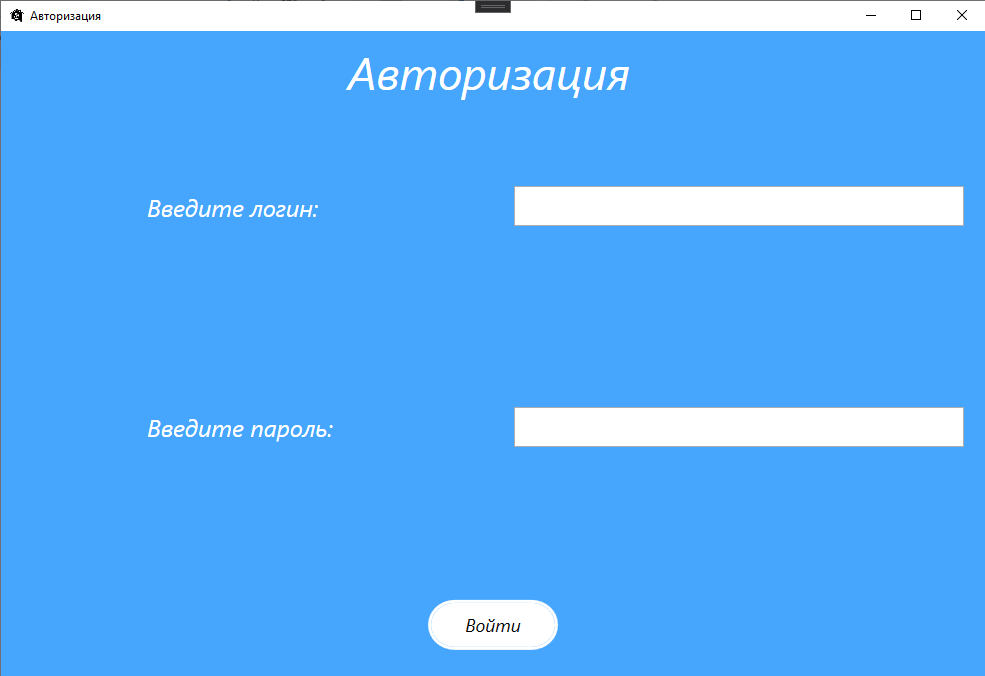


Рис. 46 – Окно «Авторизация»

После успешной авторизации Вам откроет главное меню приложения. Интерфейс окна «Главное меню» представлен на рисунке 47

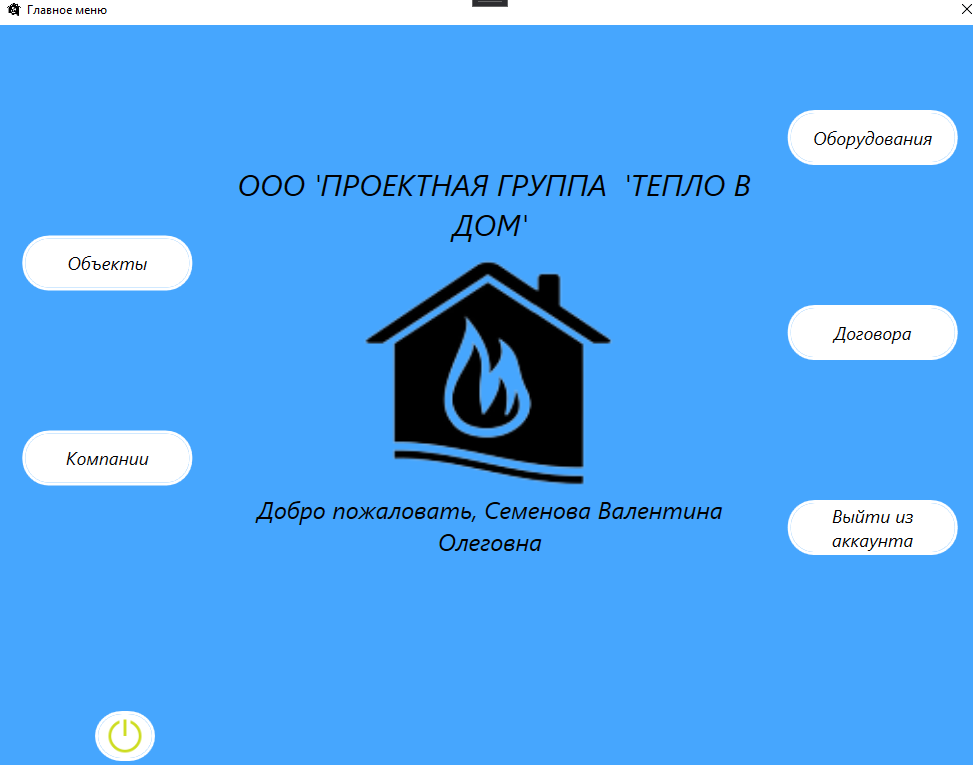


Рис. 47 – Окно «Главное меню»

Кнопка «Объекты» позволяет просмотреть информацию об объектах, а также выполнить добавление нового, изменение или удаление существующего объекта.

Кнопка «Компании» позволяет просмотреть информацию о компаниях, а также выполнить добавление новой, удаление или изменение существующей компании.

Кнопка «Оборудования» позволяет просмотреть информацию о котлах, а также выполнить добавление нового или изменение существующего котла.

Кнопка «Договора» позволяет переместиться на форму добавления нового договора, удаления существующего или формирование его в .doc. Интерфейс данной формы представлен на рисунке 48.



Рис. 48 – Форма договоров

**Страница «Добавление договора»**

Для того чтобы открыть окно добавления договора необходимо на форме «Договора» нажать на кнопку «Добавить запись». Интерфейс окна добавления объекта показан на рисунке 49.

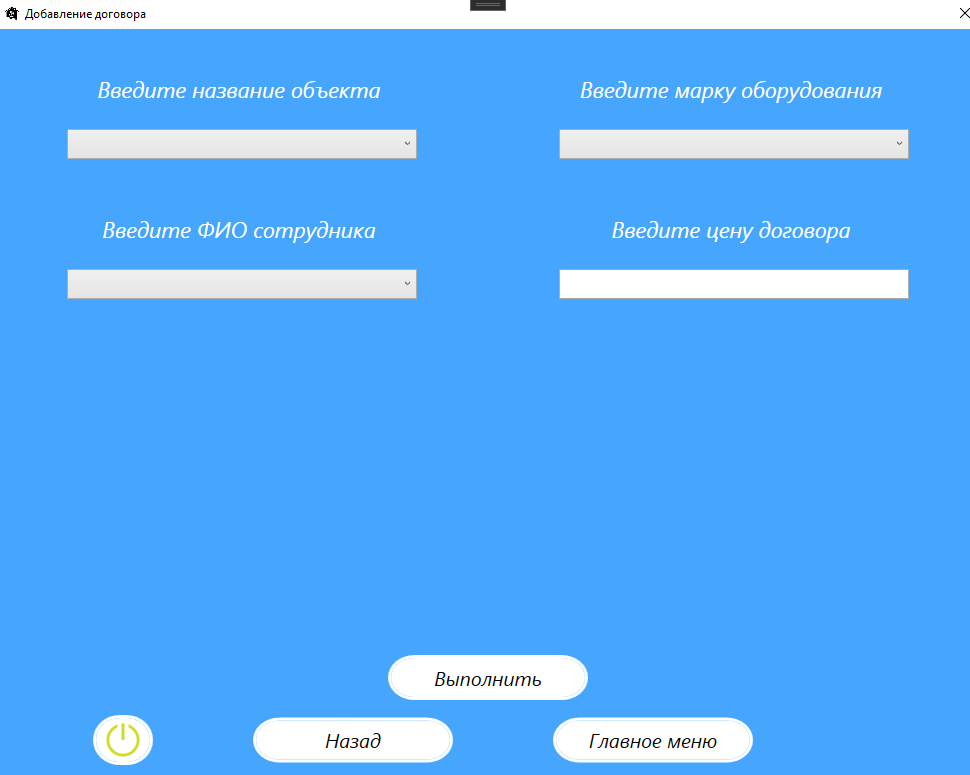


Рис. 49 – Окно добавления договора

Для добавления договора необходимо заполнить все данные в соответствии с указанными названиями сверху от каждого поля.

После заполнения и проверки всех нужных данных нажмите на кнопку «Выполнить» и договор автоматически появится в таблице со всеми договорами.

**Страница «Компании»**

Для того чтобы открыть страницу компаний необходимо на форме «Главное меню» нажать на кнопку «Компании». Интерфейс страницы компании показан на рисунке 50.

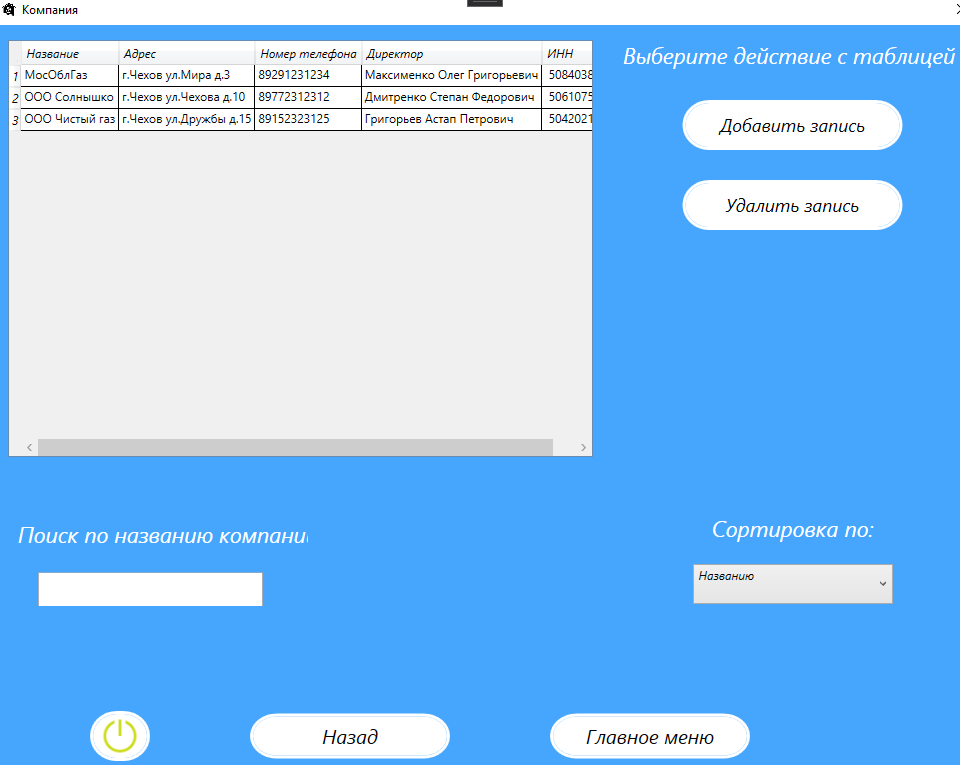


Рис. 50 – Страница компании

Здесь Вы можете добавлять/удалять компании, сортировать по названию, директору и адресу, выполнять поиск по названию.

**Страница «Добавление компании»**

Для того чтобы перейти к данной странице, необходимо на странице «Компании» нажать на кнопку «Добавить запись». Интерфейс данной страницы представлен на рисунке 51.



Рис. 51 – Страница «Добавление компании»

Для добавления компании необходимо заполнить все данные в соответствии с указанными названиями сверху от каждого поля.

После заполнения и проверки всех нужных данных нажмите на кнопку «Выполнить» и компания автоматически появится в таблице со всеми компаниями.

**Страница «Объекты»**

Для того чтобы открыть страницу компаний необходимо на форме «Главное меню» нажать на кнопку «Объекты». Интерфейс страницы объекты показан на рисунке 52.

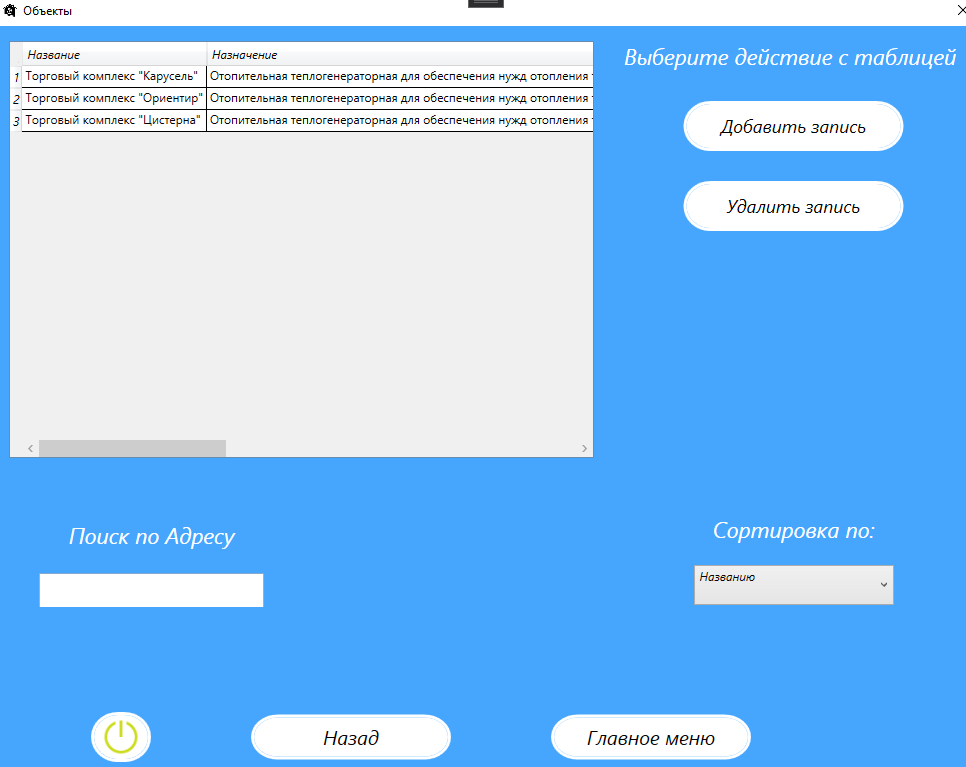


Рис. 52 – Страница объекты

Здесь Вы можете добавлять/удалять объекты, сортировать по названию, назначению, режиму работы, компании и адресу, выполнять поиск по адресу.

**Страница «Добавление объекта»**

Для того чтобы перейти к данной странице, необходимо на странице «Объекты» нажать на кнопку «Добавить запись». Интерфейс данной страницы представлен на рисунке 53.



Рис. 53 – Страница «Добавление объекта»

Для добавления объекта необходимо заполнить все данные в соответствии с указанными названиями сверху от каждого поля.

После заполнения и проверки всех нужных данных нажмите на кнопку «Выполнить» и объект автоматически появится в таблице со всеми объектами.

## 2.8. Разработка политики безопасности

Основной целью, на достижение которой направлены все положения настоящей Политики, является защита информационных ресурсов от возможного нанесения им материального, физического, морального или иного ущерба, посредством случайного или преднамеренного взаимодействия на информацию, её носители, процессы обработки и передачи, а также минимизация рисков ИБ.  
В данной программе была использована авторизация пользователя, данные для которой хранятся в базе, а также производится шифрование пароля.  
Метод шифрования представлен в листинге 31.

private string sha256(string password)

{

SHA256Managed sHA256 = new SHA256Managed();

byte[] hash = sHA256.ComputeHash(Encoding.UTF8.GetBytes(password));

StringBuilder result = new StringBuilder();

for (int i = 0; i < hash.Length; i++)

{

result.Append(hash[i].ToString("x2"));

}

return result.ToString();

}

Листинг 31 – шифрование пароля

## 2.9. Эксплуатация и развитие программы.

Данное приложение было разработано непосредственно для организации ООО 'ПРОЕКТНАЯ ГРУППА 'ТЕПЛО В ДОМ' для улучшения автоматизации процесса.  
До этого момента работа по введению и заключению договоров осуществлялась в ручной форме, но теперь, благодаря программе “Учет сделок” работа с договорами стала значительно проще. Таким образом можно сделать вывод, что данная программа будет иметь дальнейшее сопровождение и в случае успеха сможет стать программой для массового использования малых торговых организаций.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной производственной практики было разработано приложения для создания договоров газификации.

Первым этапом был произведен анализ актуальности темы и сравнения уже существующих разработок по данной теме, а именно «AccountingTrans», «Request», «DogToday». Сравнивались функциональные особенности приложений, их архитектуры и технологии, при помощи которых разработаны данные приложения.

Для организации хранения данных приложения были рассмотрены основные системы управления базами данных и основываясь на используемых технологиях для написания приложения и интеграции с уже имеющимися сервисами было принято использовать Microsoft SQL Server.

Клиентское приложение написано с использованием языка C#. Программа используется двумя пользователями. Согласно роли – менеджер, ему доступен следующий функционал:

– просмотр информации по договорам;

– добавление новой или удаления старой информации по договорам

– формирования договора;

– просмотр информации по компаниям;

– добавление новой или удаление старой информации по компаниям;

– просмотр информации по объектам;

– добавление новой или удаление старой информации по объектам;

– просмотр информации по оборудованиям;

– добавление новой или удаление старой информации по оборудованиям;

Согласно роли – Технический инспектор проектного отдела, ему доступен следующий функционал:

– просмотр информации по договорам;

– формирования договора;

– просмотр информации по компаниям;

– просмотр информации по объектам;

– просмотр информации по оборудованиям;

Благодаря использованию различных пользовательских элементов управления, было разработано приложение с удобным и интуитивно понятным пользовательским интерфейсом.

# СПИСОК ЛИТЕРАТУРЫ

1. Документация по SQL [Электронный ресурс] (Дата обращения: 27.05.2021) URL:https://docs.microsoft.com

2. SQL Задачи и решения [Электронный ресурс] (Дата обращения 26.05.2021) URL:<http://www.sql-tutorial.ru/>

3. Инструкции Transact-SQL [Электронный ресурс] (Дата обращения 01.06.2021) URL:[https://docs.microsoft.com/ru-ru/sql/t-sql/statements/statements?view=sql- server-ver15](https://docs.microsoft.com/ru-ru/sql/t-sql/statements/statements?view=sql-server-ver15)

4. Ввод в TextBox только цифр и необходимых символов C# [Электронный ресурс] (Дата обращения 03.06.2021)

URL:[https://vscode.ru/prog-lessons/vvod-v-textbox-tolko-tsifr-neobhodimyih-](https://vscode.ru/prog-lessons/vvod-v-textbox-tolko-tsifr-neobhodimyih-simvolov-c.html) [simvolov-c.html](https://vscode.ru/prog-lessons/vvod-v-textbox-tolko-tsifr-neobhodimyih-simvolov-c.html)

5. Общие сведения об объектной модели Word [Электронный ресурс] (Дата обращения 02.06.2021)

URL:[https://docs.microsoft.com/ru-ru/visualstudio/vsto/word-object-model-](https://docs.microsoft.com/ru-ru/visualstudio/vsto/word-object-model-overview?redirectedfrom=MSDN&view=vs-2019) [overview?redirectedfrom=MSDN&view=vs-2019](https://docs.microsoft.com/ru-ru/visualstudio/vsto/word-object-model-overview?redirectedfrom=MSDN&view=vs-2019)

6. SET IDENTITY\_INSERT (Transact-SQL) [Электронный ресурс] (Дата обращения 05.06.2021)

URL:<https://docs.microsoft.com/ru-ru/sql/t-sql/statements/set-identity-insert-transact-> [sql?view=sql-server-ver15](https://docs.microsoft.com/ru-ru/sql/t-sql/statements/set-identity-insert-transact-sql?view=sql-server-ver15)

# ПРИЛОЖЕНИЕ 1 – IDEF0

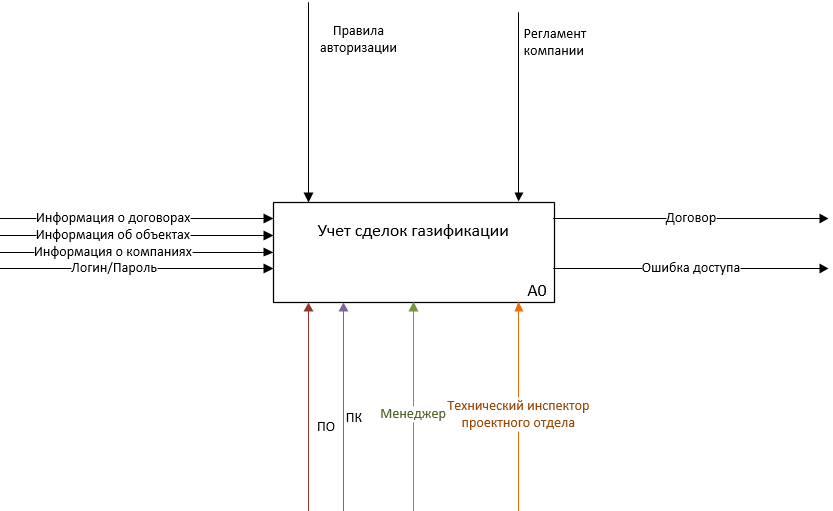


Рис.54 - IDEF0

ОКОНЧАНИЕ ПРИЛОЖЕНИЕ 1 – IDEF0

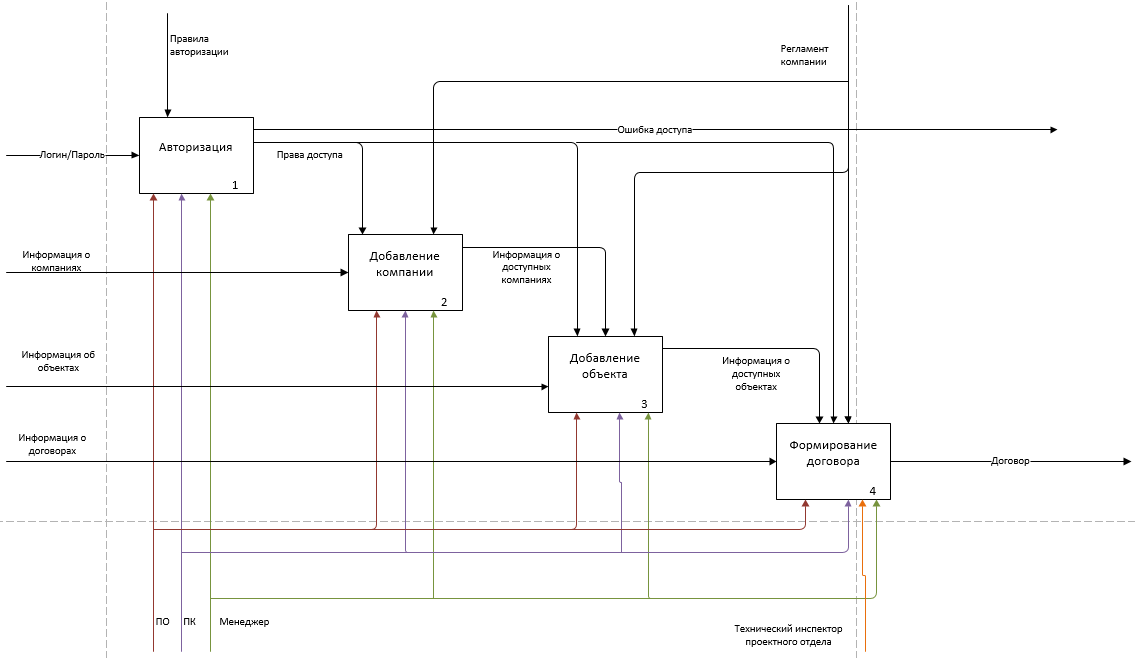


Рис.55 - IDEF0

# ПРИЛОЖЕНИЕ 2 – DFD

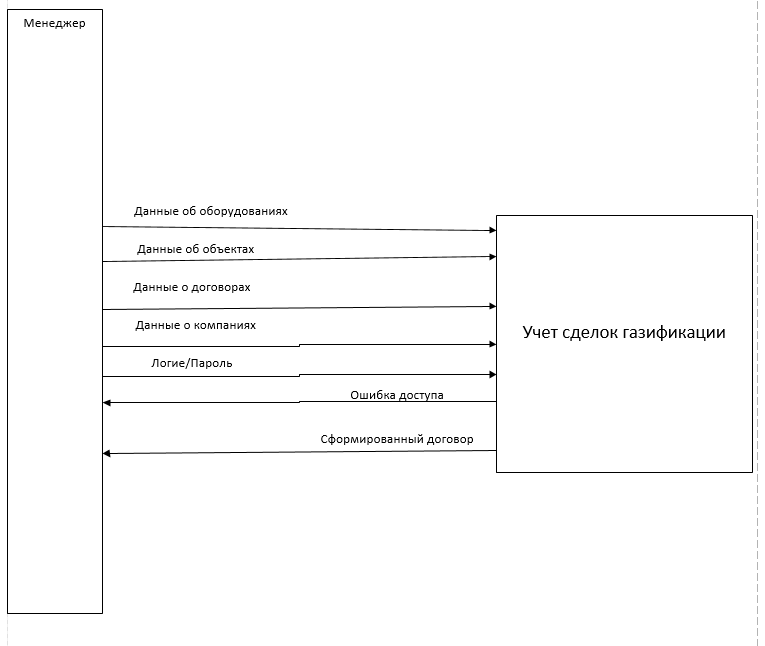


Рис.56 – DFD

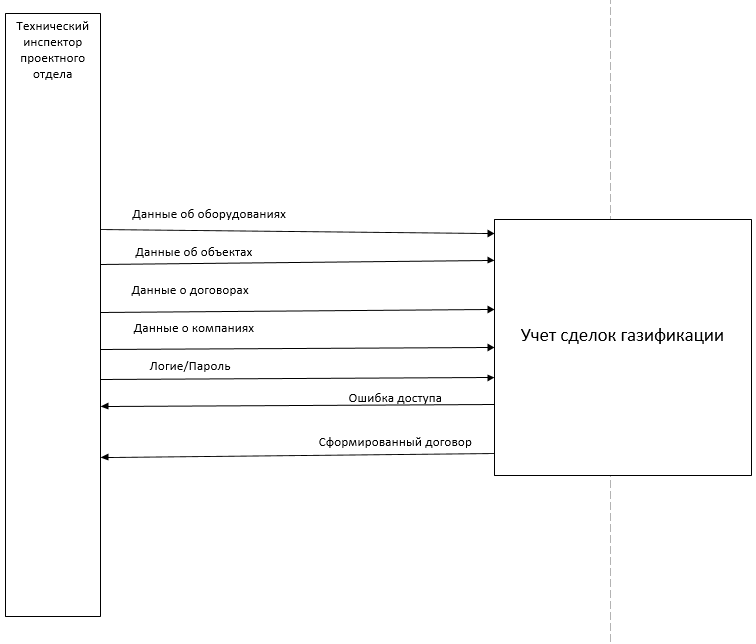


Рис.57 – DFD

ОКОНЧАНИЕ ПРИЛОЖЕНИЕ 2 – DFD

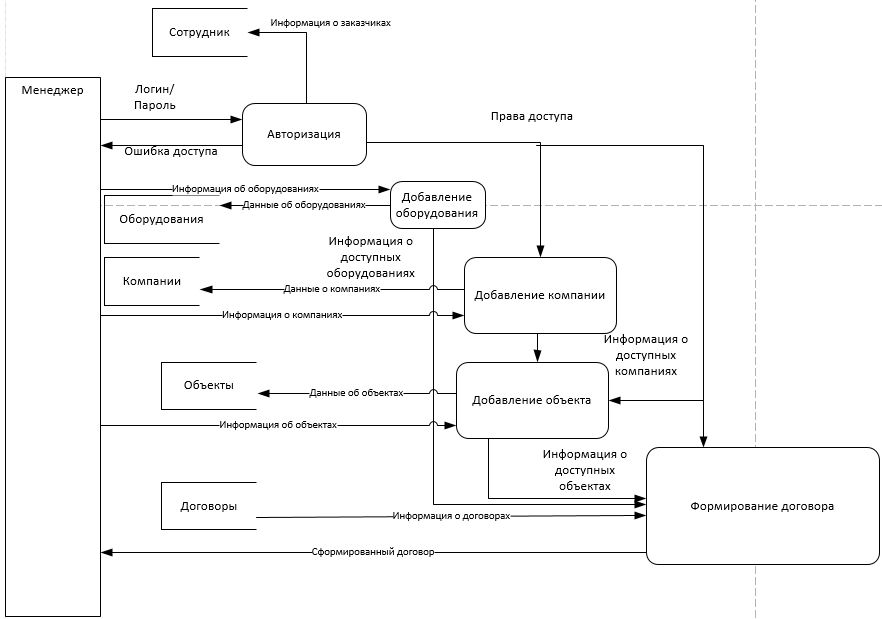


Рис. 58- DFD

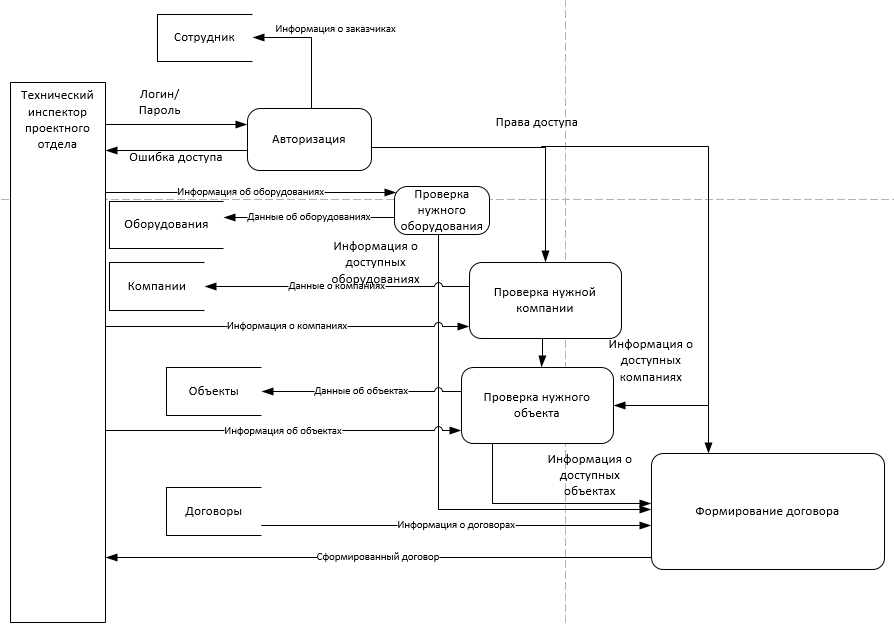


Рис. 59- DFD

# ПРИЛОЖЕНИЕ 3 – Class Diagram

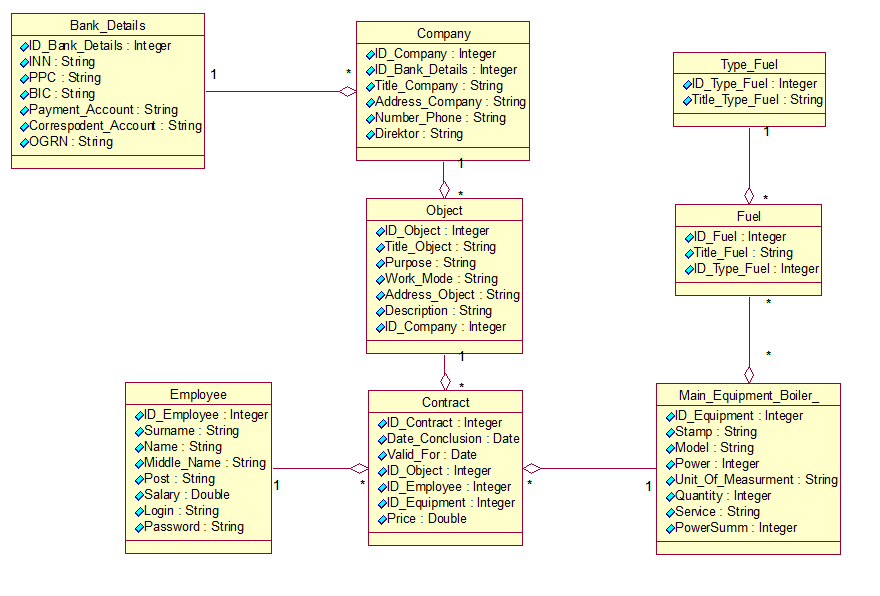


Рис. 60 - Class Diagram

# ПРИЛОЖЕНИЕ 4 – ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ

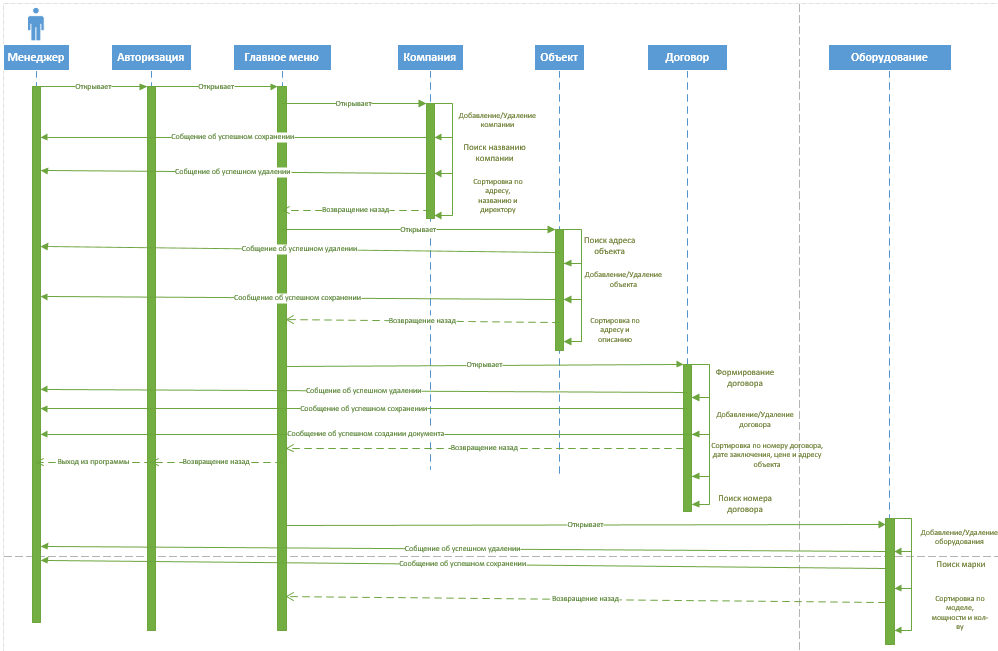


Рис. 61 – Диаграмма последовательности

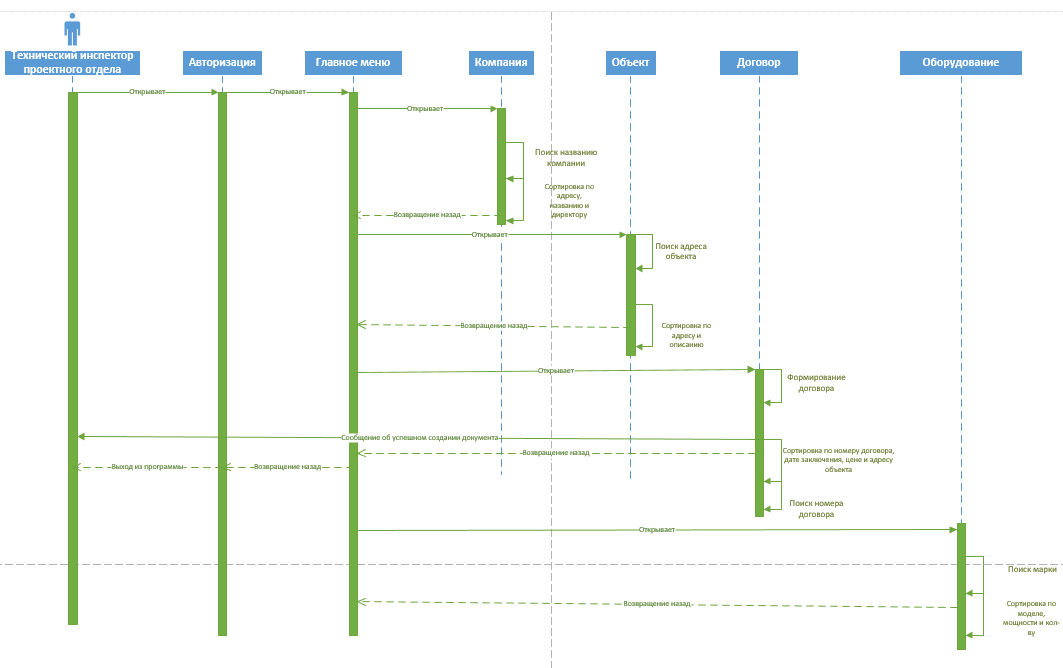


Рис. 62 – Диаграмма последовательности

# ПРИЛОЖЕНИЕ 5 – ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

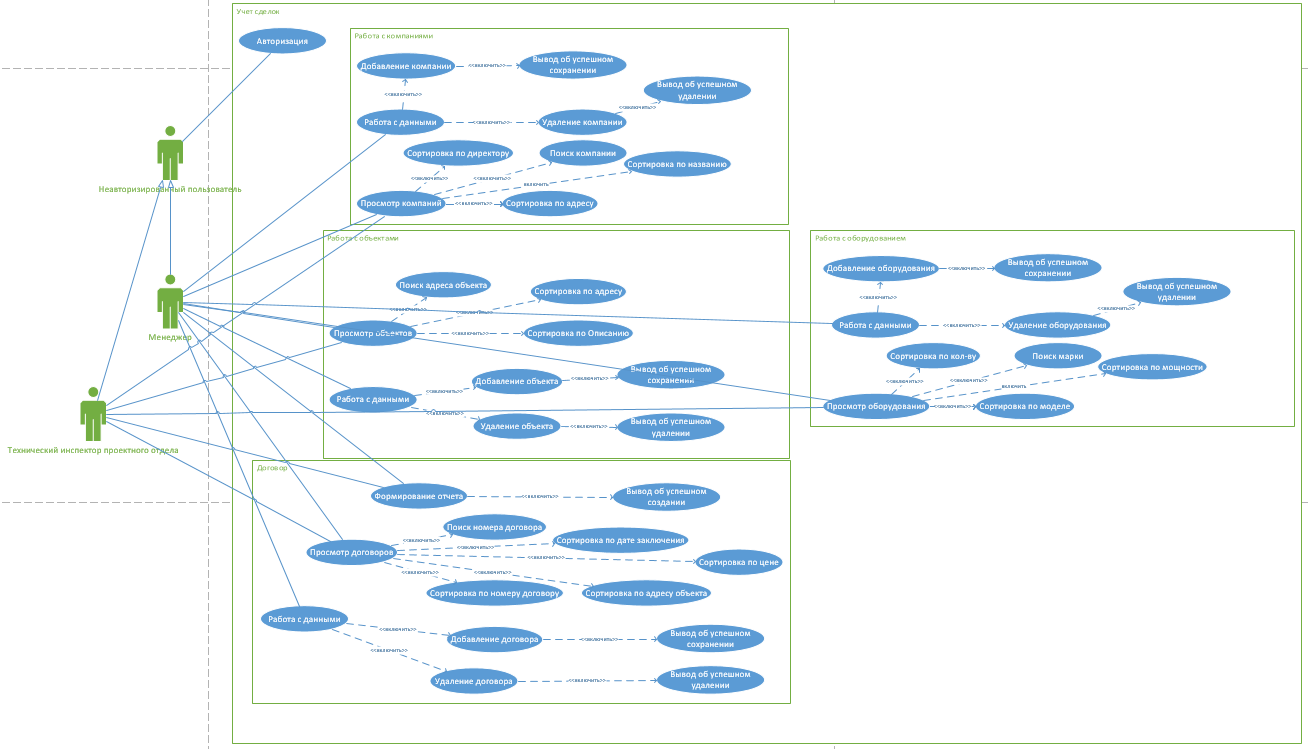


Рис.63 – Диаграмма вариантов использования.

# ПРИЛОЖЕНИЕ 6 – ДИАГРАММА ДЕЯТЕЛЬНОСТИ

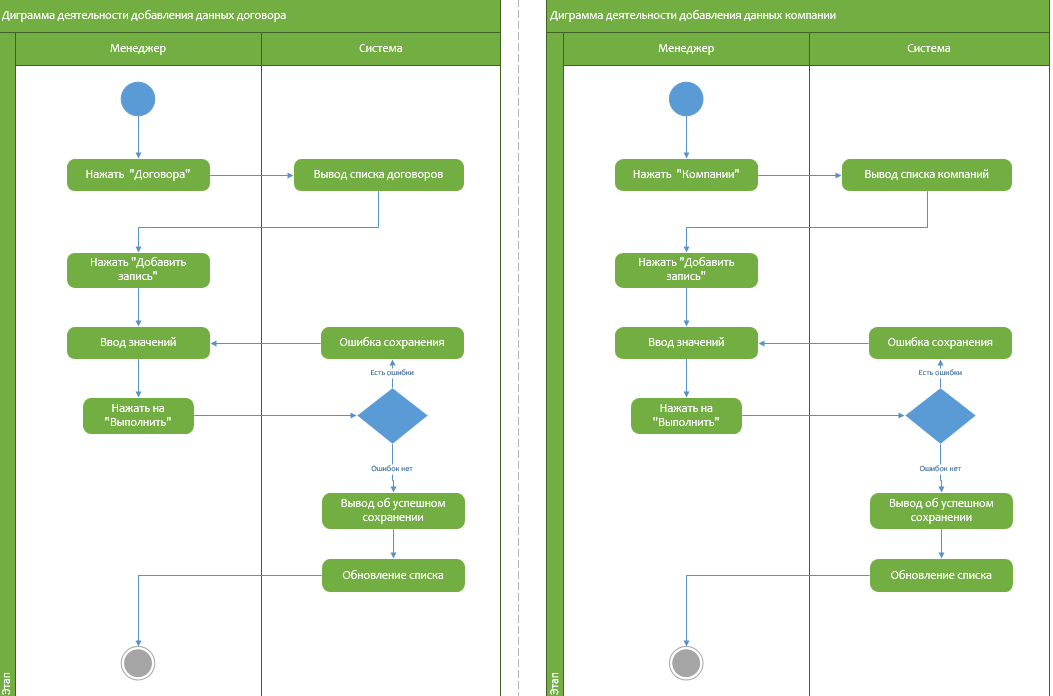


Рис.64 – Диаграмма вариантов использования.

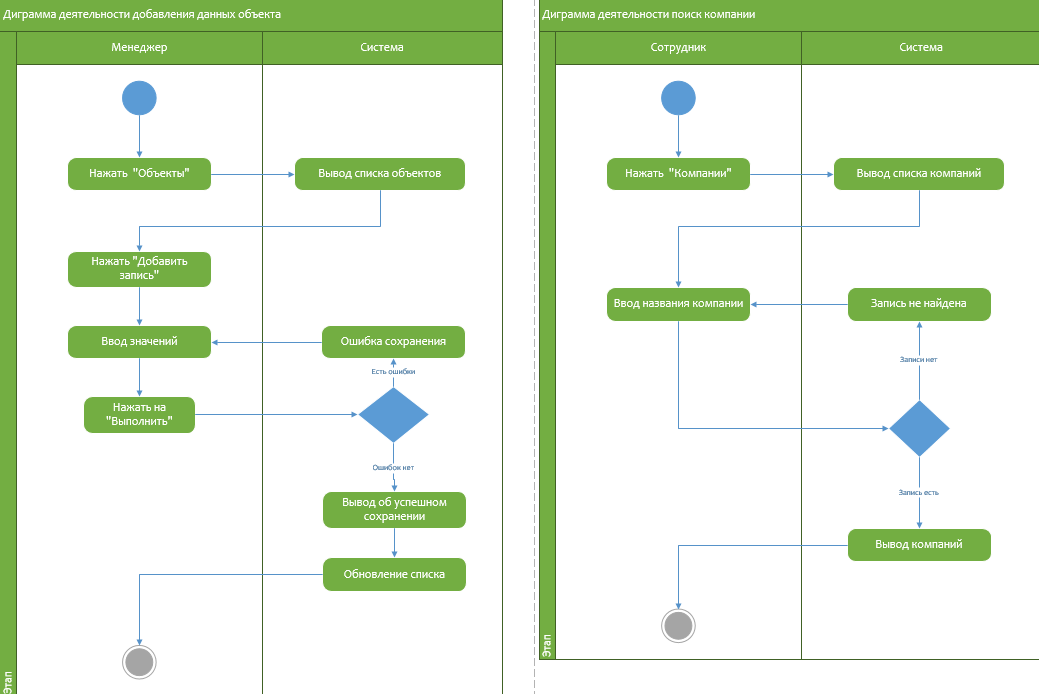


Рис.65 – Диаграмма вариантов использования.

ОКОНЧАНИЕ ПРИЛОЖЕНИЕ 6 – ДИАГРАММА ДЕЯТЕЛЬНОСТИ

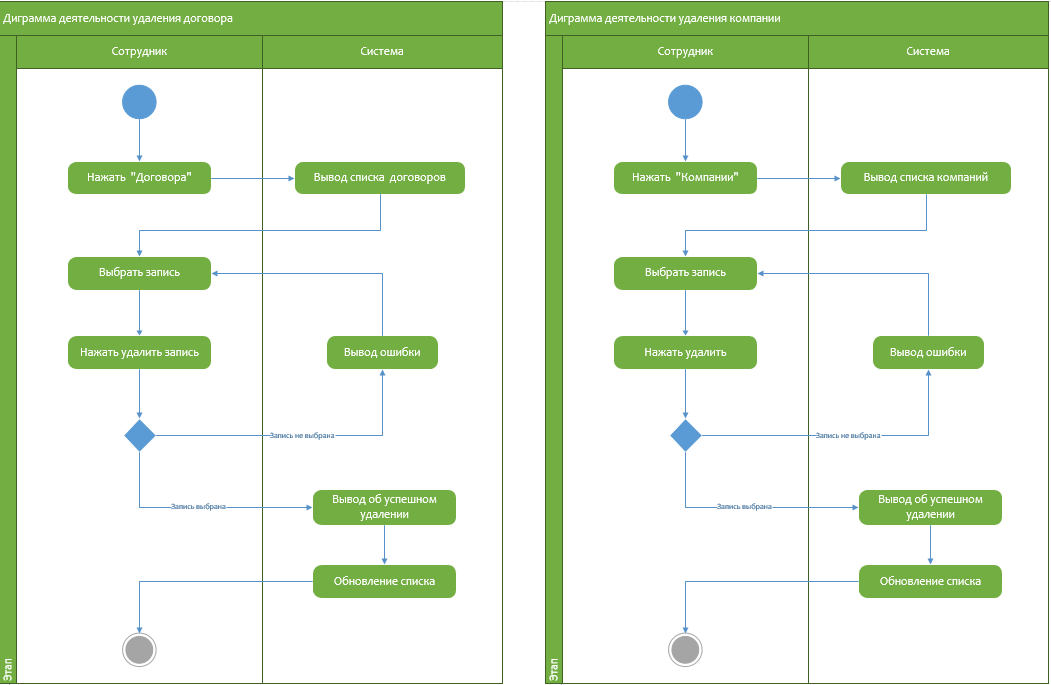


Рис.66 – Диаграмма вариантов использования.

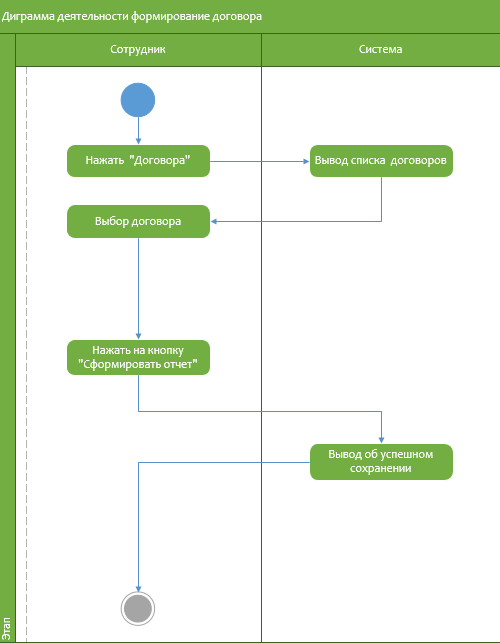


Рис.67 – Диаграмма вариантов использования.

# ПРИЛОЖЕНИЕ 7 – ДИАГРАММА РАЗВЕРТЫВАНИЯ

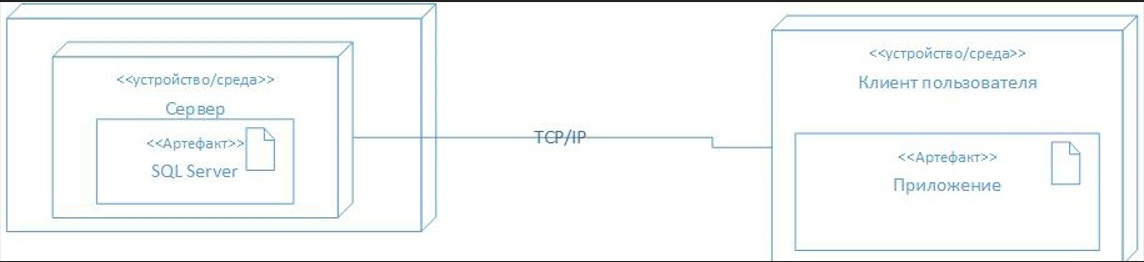


Рис.68 – Диаграмма развертывания.

# ПРИЛОЖЕНИЕ 7 – ЛОГИЧЕСКАЯ МОДЕЛЬ

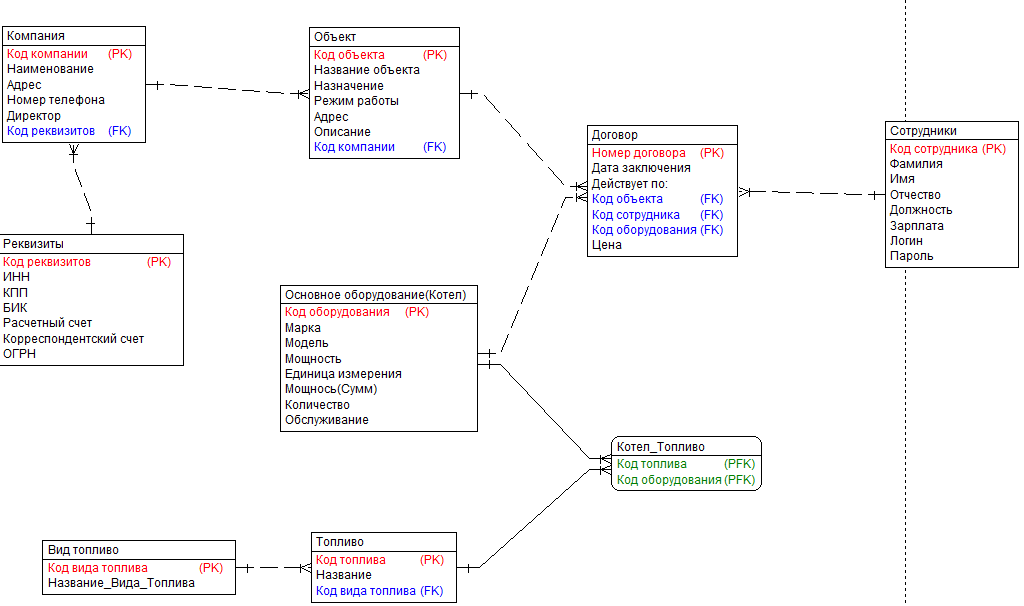


Рис. 69 – Логическая модель.

# ПРИЛОЖЕНИЕ 8 – ФИЗИЧЕСКАЯ МОДЕЛЬ

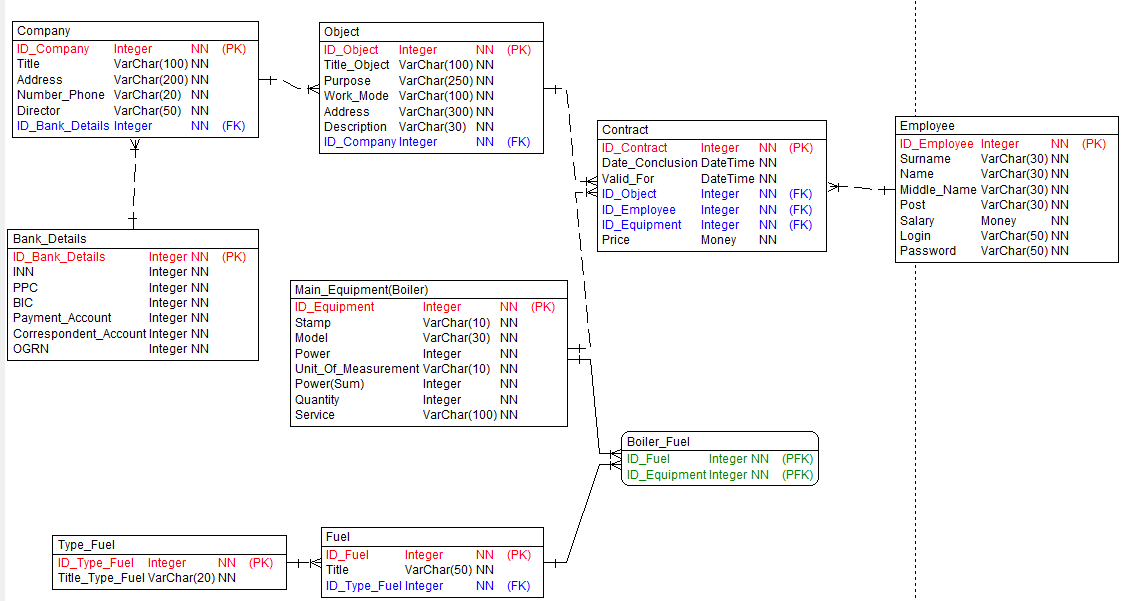


Рис. 70 – Физическая модель

# ПРИЛОЖЕНИЕ 9 – ВХОДНЫЕ ДАННЫЕ

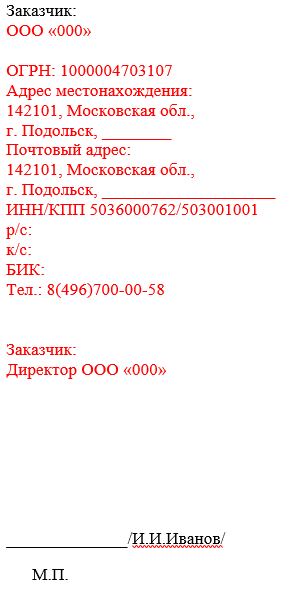


Рис. 71 – Входные данные

# ПРИЛОЖЕНИЕ 10 – ВЫХОДНЫЕ ДАННЫЕ

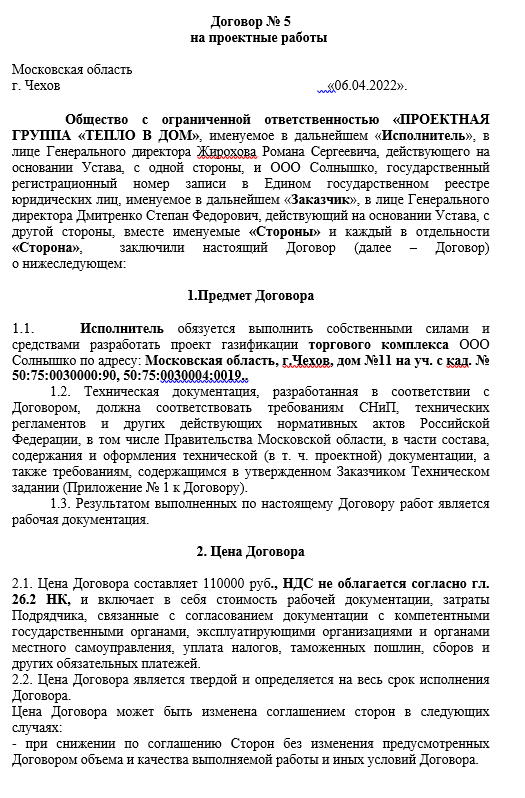
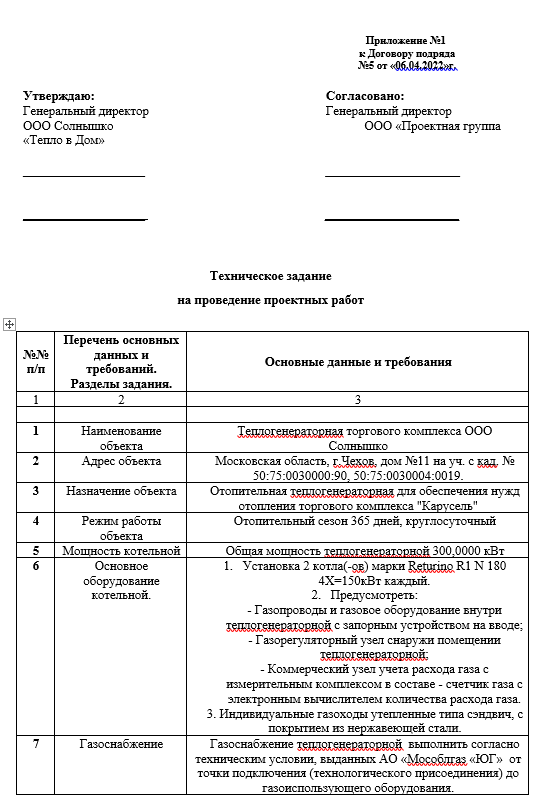
 

Рис. 72 – Выходные данные