СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 2](#_Toc100678021)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 3](#_Toc100678022)

[1.1. Обоснование актуальности задачи. 3](#_Toc100678023)

[1.2. Обоснование выбора и описание методологии разработки программы решения задачи 3](#_Toc100678024)

[1.3. Обоснование выбора и описание инструментальных средств разработки программы решения задачи 3](#_Toc100678025)

[1.4. Содержательная постановка задачи. 3](#_Toc100678026)

[2. ПРОЕКТНАЯ ЧАСТЬ 4](#_Toc100678027)

[2.1. Разработка структуры данных программы. 4](#_Toc100678028)

[2.2. Логическая модель программы. 4](#_Toc100678029)

[2.3. Разработка пользовательского интерфейса программы 4](#_Toc100678030)

[2.4. Листинг программы 4](#_Toc100678031)

[2.5.Разработка тестовых данных 4](#_Toc100678032)

[2.6. Результат работы программы 4](#_Toc100678033)

[2.7. Руководство по использования программы 4](#_Toc100678034)

[2.8. Разработка политики безопасности 4](#_Toc100678035)

[2.9. Эксплуатация и развитие программы. 4](#_Toc100678036)

[ЗАКЛЮЧЕНИЕ 5](#_Toc100678037)

# ВВЕДЕНИЕ

В последнее время, сотрудники для оформления договоров для учетов сделок формируют их вручную, то есть пишут руками.Для данной работы следует разработать приложение, чтобы облегчить и автоматезировать производство.

Для поиска и оформления договоров необходима информационная система. От абонента поступают запрос на обеспечение холодной водой, где заключается объём воды, так же отображается качество данной воды, а директор участвует в подписании контракта. При обращении к сотруднику его стандартные данные (фамилия, имя, отчество, должность, телефон) также фиксируются в базе данных, но в самой программе не отображаются данные сотрудника, ими занимается другой отдел. По каждому факту удовлетворения интересов обеих сторон составляется документ. В документе указываются оплата, сотрудник, абонент, дата подписания договора, дата окончания договора, и входящие приложения в договор.

Для эффективности работы программы по учету сделок необходимо ведение баз данных, содержащих полную информацию о абонентах, показателях качества питьевой воды, объёме потребления питьевой воды и местах исполнения.

Программа «WaterSupply» является глобальным или абстрактным представлением хранилища данных.

Целью работы является разработка приложения для создания договоров водоснабжение.

Для реализации данной цели необходимо решить следующие задачи:

* добавление информации о абонентах;
* добавление информации об договорах;
* добавление информации о договорах;
* формирование договора;
* формирование графика на месяц.

# 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

«СборРос» - программа для сбора показателей со счетчиков и создания excel таблицы, для записи полученных результатов и предоставляет возможность распечатывания данных показателей.

«Zalondor» - универсальная программа для записи абонентов на подачу заявки договора на водоснабжение в ручном виде.

«РосПост» - справочная программа для предприятия. Здесь они могут просмотреть информацию с разных сайтов, для оптимизации работы.

Плюсы и минусы этих систем представлены в таблице 1.

Таблица 1 – плюсы и минусы систем автоматизации процесса.

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

Продолжение таблицы 1

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

|  |  |  |
| --- | --- | --- |
| «РосПост» | 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 |
| Номер договора | Contract\_number | Числовой | «Договор холодного водоснабжения» |
| ФИО сотрудник | FIO employee | Текстовый | «Договор холодного водоснабжения» |
| Абонент наименование | Subscriber Name | Текстовый | «Договор холодного водоснабжения» |
| Название места исполнения | Name place completion | Текстовый | «Договор холодного водоснабжения» |
| Приложение №1 | App1 | Числовой | «Договор холодного водоснабжения» |

### 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 | Contract\_number | Числовой | «Договор» |
| ФИО сотрудник | FIO employee | Текстовый | «Договор» |
| Абонент наименование | Subscriber Name | Текстовый | «Договор» |
| Название места исполнения | Name place completion | Текстовый | «Договор» |
| Оплата | Pay | Денежный | «Договор» |
| Приложение №1 | App1 | Числовой | «Договор» |
| Приложение №2 | App2 | Числовой | «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 «Пользователь»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| ID пользователя | Id | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие  назначения- не может быть отрицательным | ID пользователя |
| Логин | Login | Текстовый | Размер – 50 символов  Обязательное поле  Не может быть пустым | Логин пользователя для входа  в программу |
| Пароль | Password | Текстовый | Размер – 50 символов  Обязательное поле  Не может быть пустым | Пароль пользователя для входа в программу |

Таблица 5 – Описание таблицы «Договор»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Номер договора | Contract\_number | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Номер договора |
| Дата начала | Signature\_date | Дата | Первичный ключ  Обязательное поле. Не может быть пустым | Дата начала договора |
| Оплата | Payment | Денежный | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Сумма оплаты договора |
| Дата окончания | Finish\_Date | Дата | Обязательное поле. Не может быть пустым | Дата окончания договора |
| Номер индикатора | Indicator\_number | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Номер индикатора |
| Номер объема | Volume\_number | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Номер объема |
| Код сотрудника | Code\_Employee | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код сотрудника |
| ИНН абонента | Subscribers\_TIN | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | ИНН абонента |

Таблица 6 – Описание таблицы «Качество питьевой воды»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Номер индикатора | Indicator\_number | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Номер индикатора |
| Индикаторы | Indicators | Текстовый | Размер – 100 символов  Обязательное поле  Не может быть пустым | Индикаторы |

Продолжение таблицы 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Единица измерения | Units\_measurement | Текстовый | Размер – 10 символов  Обязательное поле  Не может быть пустым | Единица измерения |
| Нормативы | Regulations | Текстовый | Размер – 50 символов  Обязательное поле  Не может быть пустым | Нормативы |
| Показатель вредности | Indicator\_harmfulness | Текстовый | Размер – 30 символов  Обязательное поле  Не может быть пустым | Показатель вредности |
| Класс опасности | Hazard\_class | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Класс опасности |

Таблица 7 – Описание таблицы «Работник»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Код работника | Code\_Employee | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Код работника |
| Фамилия | Surname | Текстовый | Размер – 70 символов  Обязательное поле  Не может быть пустым | Фамилия работника |
| Имя | NameEmployee | Текстовый | Размер – 250 символов  Обязательное поле  Не может быть пустым | Имя работника |
| Отчество | Midllename | Текстовый | Размер – 70 символов  Обязательное поле  Не может быть пустым | Отчество работника |
| Должность | Post | Текстовый | Размер – 450 символов  Обязательное поле  Не может быть пустым | Должность работника |
| Номер телефона | Number\_Phone | Текстовый | Размер – 20 символов  Обязательное поле  Не может быть пустым | Номер телефона  работника |

Таблица 8 – Описание таблицы «Абонент»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| ИНН абонента | Subcribers\_TIN | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | ИНН абонента |
| Наименование организации | Name\_organization | Текстовый | Размер – 70 символов  Обязательное поле  Не может быть пустым | Наименование организации |
| Расчетный счет | Cheking\_account | Текстовый | Размер – 250 символов  Обязательное поле  Не может быть пустым | Расчетный счет |
| Директор | Director | Текстовый | Размер – 70 символов  Обязательное поле  Не может быть пустым | Директор |
| Адрес | Address | Текстовый | Размер – 450 символов  Обязательное поле  Не может быть пустым | Адрес |
| Номер телефона | Number | Текстовый | Размер – 20 символов  Обязательное поле  Не может быть пустым | Номер телефона |
| Корреспондентский счет | Correspondent\_account | Числовой | Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Корреспондентский счет |

Таблица 9 – Описание таблицы «Место исполнения»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Номер места исполнения | Place\_execution\_number | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Номер места исполнения |
| Название | Title | Текстовый | Размер – 70 символов  Обязательное поле  Не может быть пустым | Название |
| Адрес | Address | Текстовый | Размер – 400 символов  Обязательное поле  Не может быть пустым | Адрес |
| ИНН абонента | Subscribers\_TIN | Числовой | Внешний ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | ИНН абонента |

Таблица 10 – Описание таблицы «Объем потребления питьевой воды»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Идентификатор | Тип данных | Условия и ограничения | Описание |
| Номер объема | Volume\_number | Числовой | Первичный ключ  Обязательное поле  Размер поля - длинное целое  Условие назначения- не может быть отрицательным | Номер объема |
| Общий объем | Total\_volume | Числовой | Обязательное поле  Условие назначения- не может быть отрицательным | Общий объем |
| Цена | Price | Денежный | Обязательное поле Условие назначения- не может быть отрицательным | Цена |
| Общее количество месяцев | Total\_number\_months | Числовой | Обязательное поле  Условие назначения- не может быть отрицательным | Общее количество месяцев |

## 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.

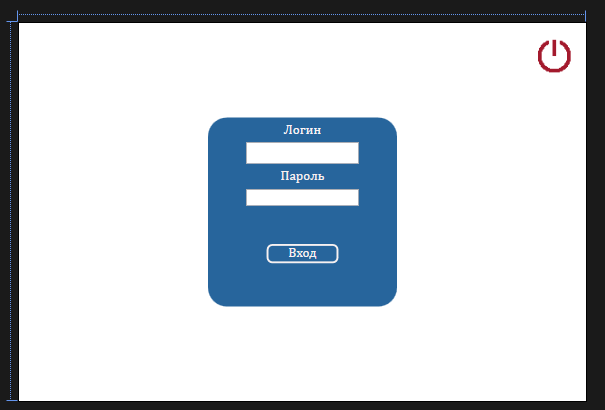


Рис. 1 – Макет окна авторизации

Интерфейс окна авторизации представлен на рисунке 2.

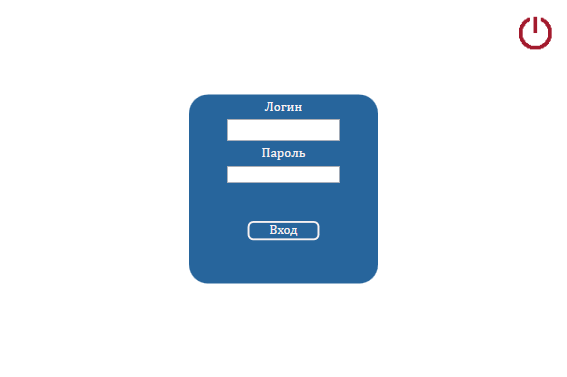


Рис. 2 – Интерфейс окна авторизации

Таблица 11 – 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" |

Продолжение таблицы 11.

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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

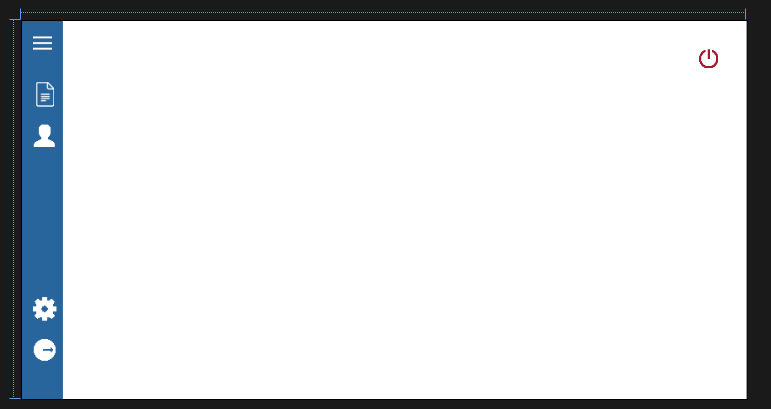


Рис. 3 – макет страницы «Главное меню»

Интерфейс страницы «Главное меню» представлен на рисунке 4.



Рис. 4 – интерфейс страницы «Главное меню»

Таблица 12 – Свойства и компоненты страницы «Главное меню»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Border | border | BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" SnapsToDevicePixels="True" |

Продолжение таблицы 12

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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" 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" Background="{x:Null}" BorderBrush="{x:Null}" |
| Button | NovAvtorizachia | Content="К авторизации" Margin="25,0,0,0" Background="{x:Null}" BorderBrush="{x:Null}" |
| Border | Bd | Height="50" BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding BorderThickness}" Background="{TemplateBinding Background}" Padding="{TemplateBinding Padding}" |

Макет страницы «Договор» представлен на рисунке 5.

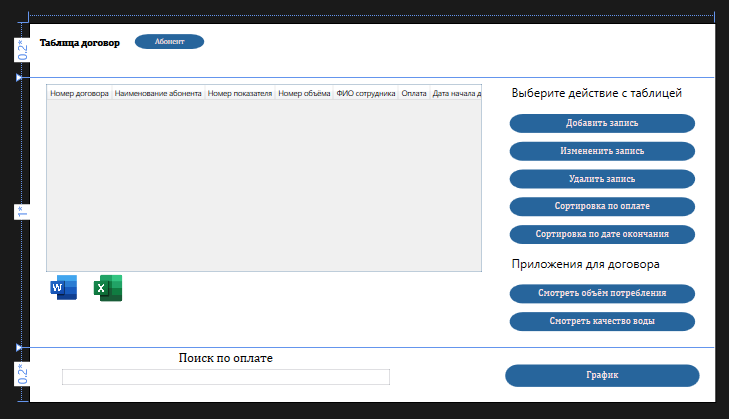


Рис. 5 – макет страницы «Договор»

Интерфейс страницы «Договор» представлен на рисунке 6.

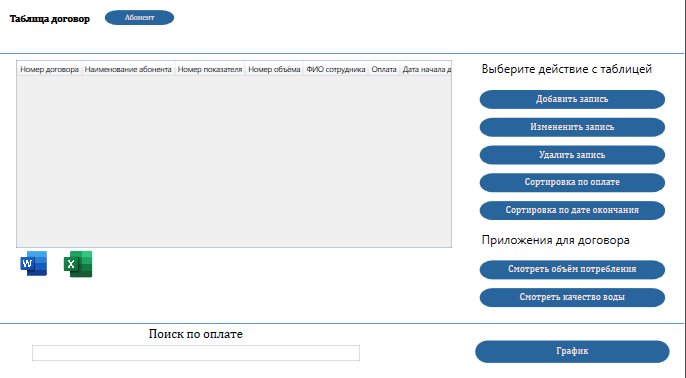


Рис. 6 – интерфейс страницы «Договор»

Таблица 13 – Свойства и компоненты страницы «Меню»

|  |  |  |
| --- | --- | --- |
| 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" |

Продолжение таблицы 13

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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" 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" |
| Border | - | Background="#27659c" RenderTransformOrigin="0,1" CornerRadius="20" Margin="7" |
| Border | - | Background="#27659c" RenderTransformOrigin="0,1" CornerRadius="20" Margin="7" |
| Button | SortDateS | FontSize="15" Content="Сортировка по оплате" 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" |

Продолжение таблицы 13

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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}" 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" |
| Label | - | FontFamily="Cambria Math" FontSize="16" Content="Таблица договор" Margin="10,16,0,0" FontWeight="Bold" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="753,26,25,24" |

Макет страницы «Показатель качества питьевой воды» представлен на рисунке 7.

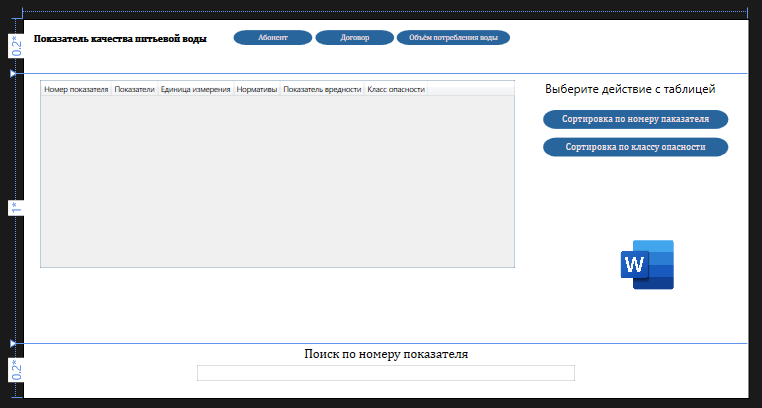


Рис. 7 – макет страницы «Показатель качества питьевой воды»

Интерфейс страницы «Показатель качества питьевой воды» представлен на рисунке 8

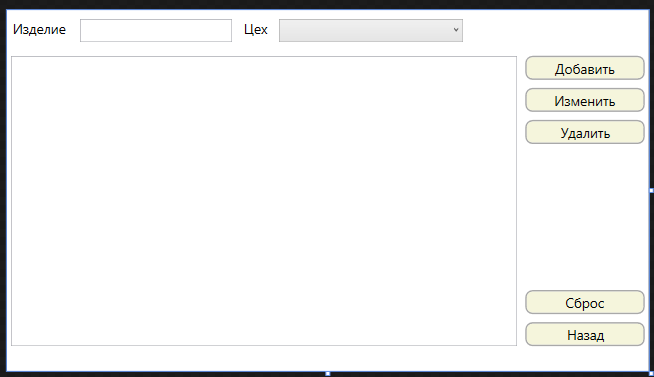


Рис. 8 – Интерфейс страницы «Показатель качества питьевой воды»

Таблица 14 – Свойства и компоненты страницы «Показатель качества питьевой воды»

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="White" |
| 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" |

Продолжение таблицы 14

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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" |
| Button | SortHazzard | FontSize="15" Content="Сортировка по классу опасности" Height="30" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| 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" FontFamily="Cambria Math" Margin="10,0" |

Макет страницы «Объём потребления питьевой воды» представлен на рисунке 9.

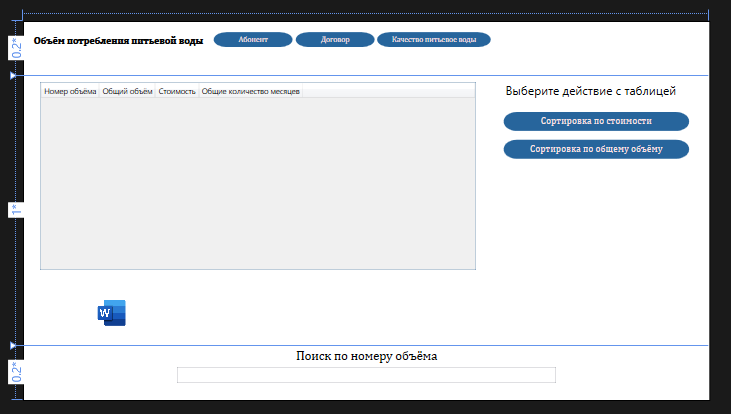


Рис. 9 – макет страницы «Объём потребления питьевой воды»

Интерфейс страницы «Объём потребления питьевой воды» представлен на рисунке 10.

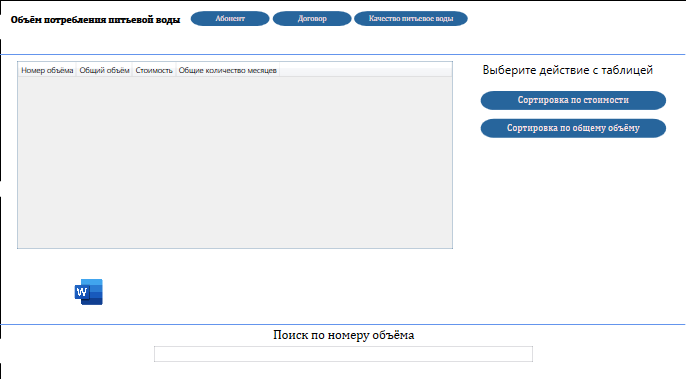


Рис. 10 – интерфейс страницы «Объём потребления питьевой воды»

Таблица 15 – Свойства и компоненты страницы «Объём потребления питьевой воды».

|  |  |  |
| --- | --- | --- |
| 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" |

Продолжение таблицы 15

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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" |
| Button | ContractPG | Width="105" Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0 |
| Button | btnCancel | Content="Назад" Width="210" FontSize="22" Margin="5" Click="btnCancel\_Click" |
| Label | - | FontFamily="Cambria Math" FontSize="16" Content="Объём потребления питьевой воды" Margin="10,16,0,0" FontWeight="Bold" |

Макет страницы «Абонент» представлен на рисунке 11.

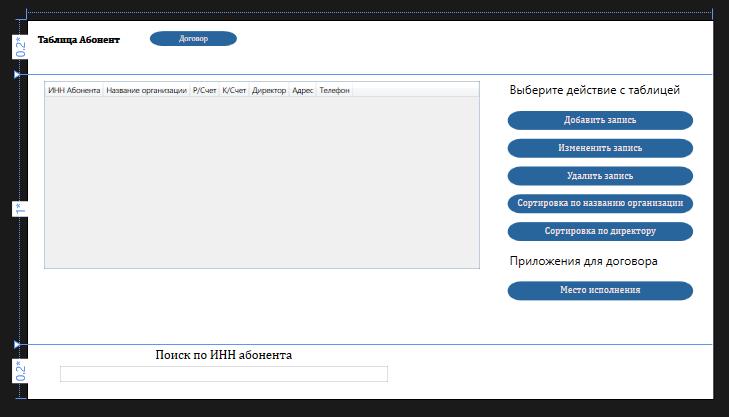


Рис. 11 – макет страницы «Абонент»

Интерфейс страницы «Абонент» представлен на рисунке 12.



Рис. 12 – интерфейс страницы «Объём потребления питьевой воды»

Таблица 16 – Свойства и компоненты страницы «Абонент».

|  |  |  |
| --- | --- | --- |
| 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 |

Продолжение таблицы 16

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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" |
| Border | - | Background="#27659c" CornerRadius="20" HorizontalAlignment="Center" Margin="0,16,360,46" |
| 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}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Label | - | FontFamily="Cambria Math" Content="Поиск по ИНН абонента" FontSize="20" |
| TextBox | - | FontFamily="Cambria Math" Margin="50,0,50,0" Height="25" |

Макет страницы «Место исполнения» представлен на рисунке 13.

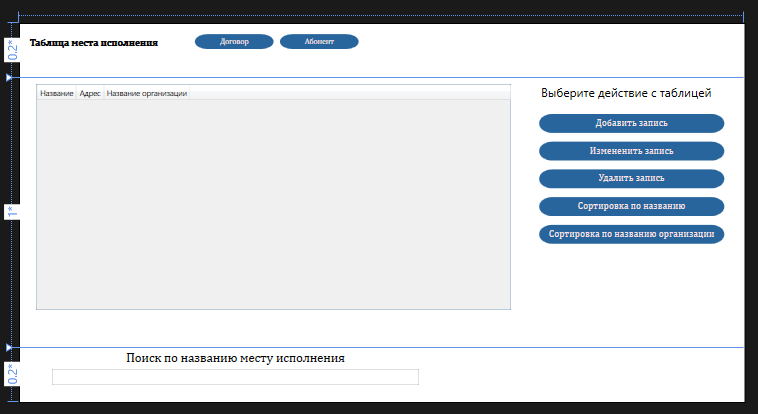


Рис. 13 – макет страницы «Место исполнения»

Интерфейс страницы «Место исполнения» представлен на рисунке 14.

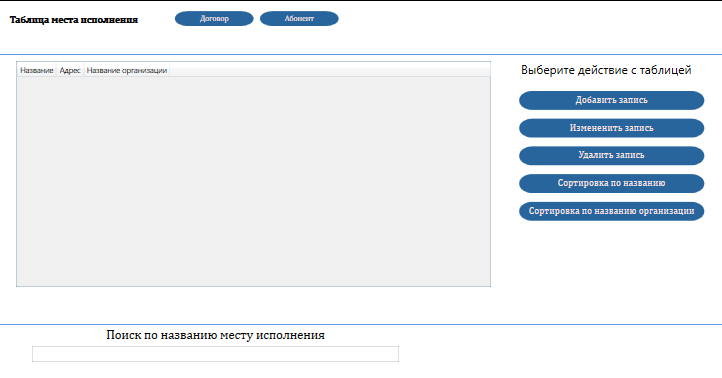


Рис. 14 – интерфейс страницы «Место исполнения»

Таблица 17 – Свойства и компоненты страницы «Место исполнения».

|  |  |  |
| --- | --- | --- |
| 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 |

Продолжение таблицы 17

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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" 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" |
| Button | SubBtn | FontSize="15" Content="Абонент" Height="30"  Background="{x:Null}" BorderBrush="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Label | - | FontFamily="Cambria Math"  Content=" Таблица места исполнения " FontSize="20" |
| TextBox | - | FontFamily="Cambria Math" Margin="50,0,50,0" Height="25" x:Name="SearchTIN" |

Макет страницы «Добавление/Изменение договора» представлен на рисунке 15.

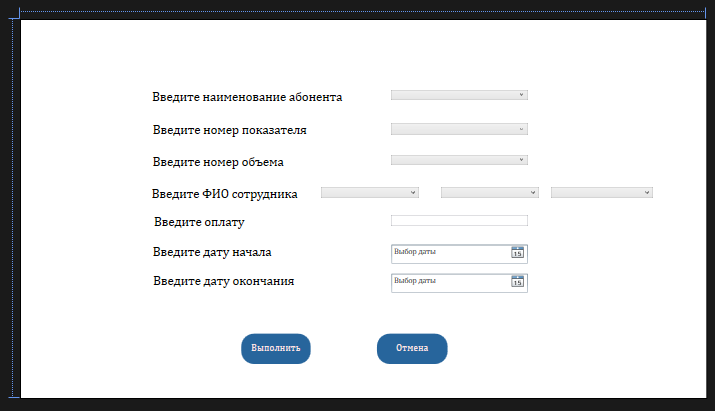


Рис. 15 – макет страницы «Добавление/Изменение договора»

Интерфейс страницы «Добавление/Изменение договора» представлен на рисунке 15.

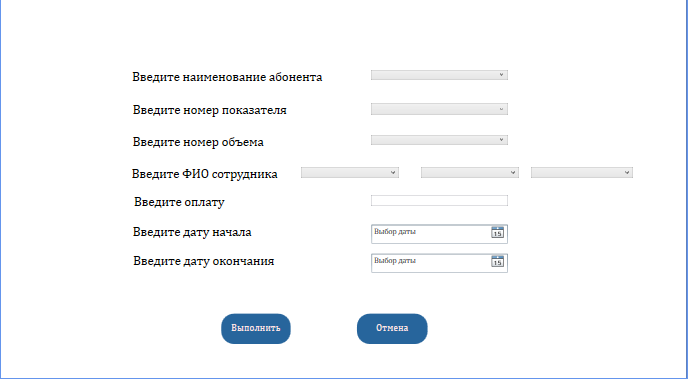


Рис. 15 – интерфейс страницы «Добавление/Изменение договора»

Таблица 18 – Свойства и компоненты страницы «Добавление/Изменение договора».

|  |  |  |
| --- | --- | --- |
| 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" |

Продолжение таблицы 18

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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" |
| Button | Ok | Content="Выполнить" FontSize="15" Height="25" Background="{x:Null}" Foreground="#FFFBE5E5" FontFamily="Cambria Math" Margin="10,0" |
| Border | - | Background="#27659c" CornerRadius="20" Margin="564,497,410,54" |
| 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/Изменение абонента» представлен на рисунке 16.

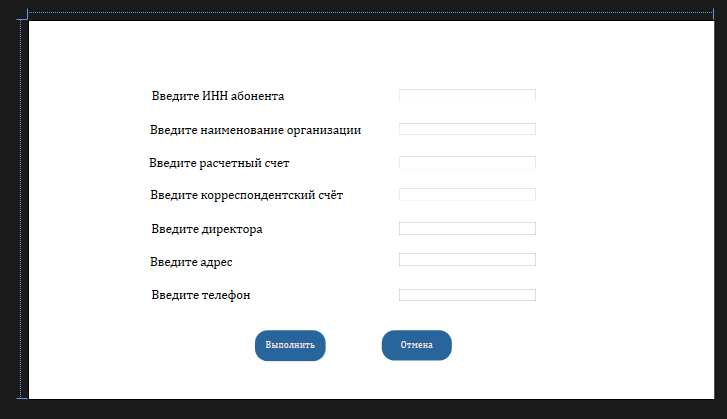


Рис. 16 – макет страницы «Добавление/Изменение абонента»

Интерфейс страницы «Добавление/Изменение абонента» представлен на рисунке 17.



Рис. 17 – интерфейс страницы «Добавление/Изменение договора»

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

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| Grid | - | Background="White" |
| TextBlock | TN | FontFamily="Cambria Math" Margin="207,111,572,458" Height="30" Text="Введите наименование абонента " FontSize="20" |
| TextBox | TINSub | Height="20" Margin="587,108,283,472" FontFamily="Cambria Math" |
| TextBlock | - | FontFamily="Cambria Math" Margin="207,164,627,405" Height="30" Text="Введите номер показателя " FontSize="20" |

Продолжение таблицы 19

|  |  |  |
| --- | --- | --- |
| 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/Изменение абонента» представлен на рисунке 16.

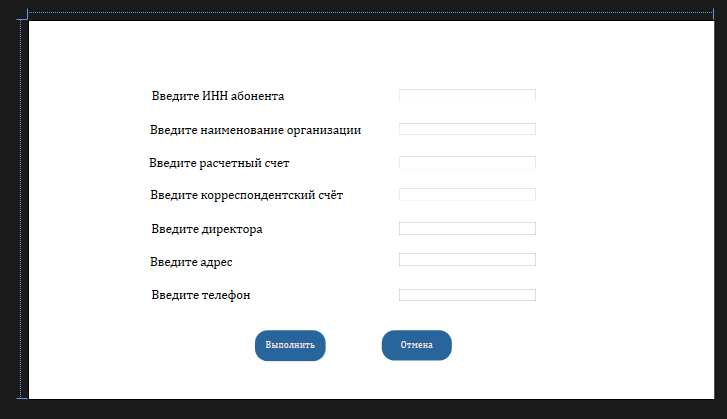


Рис. 16 – макет страницы «Добавление/Изменение абонента»

Интерфейс страницы «Добавление/Изменение абонента» представлен на рисунке 17.



Рис. 17 – интерфейс страницы «Добавление/Изменение договора»

Таблица 20 – Свойства и компоненты страницы «Добавление/Изменение договора».

|  |  |  |
| --- | --- | --- |
| 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" |

Продолжение таблицы 20

|  |  |  |
| --- | --- | --- |
| Компонент | Наименование компонента | Свойства компонента |
| 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 разметки окна «MainWindow» представлен в листинге 1-2.

using System;

using System.Collections.Generic;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WaterSupply{

Листинг 1 - «MainWindow»

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

public static string Encrypt(string value)

{

StringBuilder Sb = new StringBuilder();

using (SHA256 hash = SHA256Managed.Create())

{

Encoding enc = Encoding.UTF8;

Byte[] result = hash.ComputeHash(enc.GetBytes(value));

foreach (Byte b in result)

Sb.Append(b.ToString("x2"));

}

return Sb.ToString();

}

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

{

string EncryptPassword = Encrypt(Pass.Password);

try

{

User EnterUser = DataBase.a.User.Where(l => l.login == Login.Text && l.password == EncryptPassword).FirstOrDefault();

if (EnterUser != null)

{

MainWn win = new MainWn();

this.Close();

win.Show();

}

else throw new Exception("Пользователь не найден");

}

catch (Exception)

{

MessageBox.Show("Произошла ошибка с sql сервером","Ошибка!", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

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

{

this.Close();}}}

Листинг 2 - «MainWindow»

Листинг разметки окна «MainWn» представлен в листинге 3-4.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

Листинг 3 - «MainWn»

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using WaterSupply.Lists;

using WaterSupply.Properties;

namespace WaterSupply{

/// <summary>

/// Логика взаимодействия для MainWn.xaml

/// </summary>

public partial class MainWn : Window{

public MainWn()

{

InitializeComponent();

var mainpage = new mainPage();

MainFrame.Navigate(mainpage);}

private void ListViewItem\_MouseEnter(object sender, MouseEventArgs e){

if (Tg\_Btn.IsChecked == true)

{

tt\_home.Visibility = Visibility.Collapsed;

tt\_contacts.Visibility = Visibility.Collapsed;

tt\_settings.Visibility = Visibility.Collapsed;

tt\_signout.Visibility = Visibility.Collapsed;}

else{

tt\_home.Visibility = Visibility.Visible;

tt\_contacts.Visibility = Visibility.Visible;

tt\_settings.Visibility = Visibility.Visible;

tt\_signout.Visibility = Visibility.Visible;}}

private void BG\_PreviewMouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

Tg\_Btn.IsChecked = false;}

private void NovAvtorizachia\_Click(object sender, RoutedEventArgs e){

MainWindow main = new MainWindow();

this.Close();

main.Show();}

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

{

this.Close();}

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

{

MainFrame.Navigate(new ContractPage());}

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

{

MainFrame.Navigate(new SubscriberPage());}

private void ImgContBtn\_Click(object sender, RoutedEventArgs e){

MainFrame.Navigate(new ContractPage());}

private void ImgBtnSub\_Click(object sender, RoutedEventArgs e){

MainFrame.Navigate(new SubscriberPage());}

private void ImgBtnNovAvto\_Click(object sender, RoutedEventArgs e){

MainWindow main = new MainWindow();

this.Close();

main.Show();}

private void ImgBtnSvoistvo\_Click(object sender, RoutedEventArgs e){

MainFrame.Navigate(new Inf());}

private void Svoistvo\_Click(object sender, RoutedEventArgs e){

MainFrame.Navigate(new Inf());}}}

Листинг 4 - «MainWn»

Листинг разметки окна «WaterQualityPage» представлен в листинге 5-6.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using word = Microsoft.Office.Interop.Word;

using Microsoft.Office.Interop.Word;

using Page = System.Windows.Controls.Page;

using System.Text.RegularExpressions;

using System.IO;

using Microsoft.Win32;

namespace WaterSupply.Lists

{

/// <summary>

/// Логика взаимодействия для WaterQualityPage.xaml

/// </summary>

public partial class WaterQualityPage : Page

{

public WaterQualityPage()

{

InitializeComponent();

dgQuality.ItemsSource = DataBase.a.Drinking\_water\_quality\_indicator.ToList();

}

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

{

NavigationService.Content = new VolumePage();

}

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

{

NavigationService.Content = new ContractPage();

}

private void SubBtn\_Click(object sender, RoutedEventArgs e){

NavigationService.Content = new SubscriberPage();

}

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

{

try

{

if (dgQuality.SelectedItems.Count > 0)

{

{

SaveFileDialog sfd = new SaveFileDialog();

string source = $@"{Directory.GetCurrentDirectory()}\Показатель качества питьевой воды.docx";

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

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

doc.Activate();

try

{

Листинг 5 - «WaterQualityPage»

var x = dgQuality.SelectedItem as Drinking\_water\_quality\_indicator;

if (sfd.ShowDialog() == false){

doc.Close();

doc = null;

app.Quit();

return;}

var results = DataBase.a.Drinking\_water\_quality\_indicator.Where(P => P.Indicator\_number == x.Indicator\_number).FirstOrDefault();

word.Bookmarks wB = doc.Bookmarks;

wB["НомерПоказателя"].Range.Text = results.Indicator\_number.ToString();

wB["Показатели"].Range.Text = results.Indicators.ToString();

wB["ЕдиницыИзмерения"].Range.Text = results.Units\_measurement.ToString();

wB["Нормативы"].Range.Text = results.Regulations.ToString();

wB["ПоказательВредности"].Range.Text = results.Indicator\_harmfulness.ToString();

wB["КлассОпасности"].Range.Text = results.Hazard\_class.ToString();

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 MessageBox.Show("Выберите строку для создания word документа.", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex){

MessageBox.Show(ex.Message);}}

private void SortNumInd\_Click(object sender, RoutedEventArgs e){

dgQuality.ItemsSource = DataBase.a.Drinking\_water\_quality\_indicator.OrderBy(c => c.Indicator\_number).ToList();}

private void SortHazzard\_Click(object sender, RoutedEventArgs e){

dgQuality.ItemsSource = DataBase.a.Drinking\_water\_quality\_indicator.OrderBy(c => c.Hazard\_class).ToList();

}

private void SearchNumInd\_TextChanged(object sender, TextChangedEventArgs e){

dgQuality.ItemsSource = DataBase.a.Drinking\_water\_quality\_indicator.Where(p => p.Indicator\_number.ToString().Contains(SearchNumInd.Text)).ToList();}

private void SearchNumInd\_PreviewTextInput(object sender, TextCompositionEventArgs e){

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

e.Handled = true;}}

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

{

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

Листинг 6 - «WaterQualityPage»

Листинг разметки окна «VolumePage» представлен в листинге 7-8.

sing Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using word = Microsoft.Office.Interop.Word;

using Microsoft.Office.Interop.Word;

using Page = System.Windows.Controls.Page;

using System.Text.RegularExpressions;

using System.IO;

namespace WaterSupply.Lists

{

/// <summary>

/// Логика взаимодействия для VolumePage.xaml

/// </summary>

public partial class VolumePage : Page

{

public VolumePage()

{

InitializeComponent();

dgVol.ItemsSource = DataBase.a.Volume\_of\_drinking\_water\_consumption.ToList();

}

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

{

NavigationService.Content = new ContractPage();

}

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

{

NavigationService.Content = new WaterQualityPage();

}

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

{

NavigationService.Content = new SubscriberPage();

}

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

{

try

{

if (dgVol.SelectedItems.Count > 0)

{

{

SaveFileDialog sfd = new SaveFileDialog();

string source = [$@"{Directory.GetCurrentDirectory()}\Объём потребления питьевой воды.docx](mailto:$@%22%7bDirectory.GetCurrentDirectory()%7d\Объём%20потребления%20питьевой%20воды.docx)";

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

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

doc.Activate();

try{

Листинг 7 - «VolumePage»

var x = dgVol.SelectedItem as Volume\_of\_drinking\_water\_consumption;

if (sfd.ShowDialog() == false)

{

doc.Close();

doc = null;

app.Quit();

return;}

var results = DataBase.a.Volume\_of\_drinking\_water\_consumption.Where(P => P.Volume\_number == x.Volume\_number).FirstOrDefault();

word.Bookmarks wB = doc.Bookmarks;

wB["НомерОбъёма"].Range.Text = results.Volume\_number.ToString();

wB["ОбщийОбъём"].Range.Text = results.Total\_volume.ToString();

wB["Стоимость"].Range.Text = results.Price.ToString();

wB["ОбщиеКоличествоМесяцев"].Range.Text = results.Total\_number\_months.ToString();

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 MessageBox.Show("Выберите строку для создания word документа.", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);}}

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

{

dgVol.ItemsSource = DataBase.a.Volume\_of\_drinking\_water\_consumption.OrderBy(c => c.Price).ToList();

}

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

{

dgVol.ItemsSource = DataBase.a.Volume\_of\_drinking\_water\_consumption.OrderBy(c => c.Total\_volume).ToList();

}

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

{

dgVol.ItemsSource = DataBase.a.Volume\_of\_drinking\_water\_consumption.Where(p => p.Volume\_number.ToString().Contains(SearchNumVol.Text)).ToList();

}

private void SearchNumVol\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

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

e.Handled = true;}}

private void dgVol\_LoadingRow(object sender, DataGridRowEventArgs e){

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

Листинг 8 - «VolumePage»

Листинг разметки окна «SubcriberPage» представлен в листинге 9-10.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WaterSupply.Lists.Changeses;

namespace WaterSupply.Lists

{

/// <summary>

/// Логика взаимодействия для SubscriberPage.xaml

/// </summary>

public partial class SubscriberPage : Page

{

public SubscriberPage()

{

InitializeComponent();

dgSub.ItemsSource = DataBase.a.Subscriber.ToList();

}

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

{

NavigationService.Content = new ContractPage();

}

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

{

NavigationService.Content = new ChangesesSubscriberPage();

}

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

{

try

{

if (dgSub.SelectedItems.Count > 0)

{

var tablcont = dgSub.SelectedItem as Subscriber;

NavigationService.Content = new ChangesesSubscriberPage(tablcont);

}

else MessageBox.Show("Выберите строку для изменения.", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

NavigationService.Content = new PlaceCompletionPage();

}

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

{

Листинг 9 - «SubcriberPage»

try

{

if (MessageBox.Show("Вы уверены, что хотите удалить запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

if (dgSub.SelectedItems.Count > 0)

{

var a = dgSub.SelectedItem as Subscriber;

DataBase.a.Subscriber.Remove(a);

DataBase.a.SaveChanges();

dgSub.ItemsSource = DataBase.a.Subscriber.ToList();

}

else MessageBox.Show("Выберите строку для удаления.", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);}}

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

{

dgSub.ItemsSource = DataBase.a.Subscriber.OrderBy(c => c.Director).ToList();

}

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

{

dgSub.ItemsSource = DataBase.a.Subscriber.OrderBy(c => c.Name\_organization).ToList();

}

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

{

dgSub.ItemsSource = DataBase.a.Subscriber.Where(p => p.Subscribers\_TIN.ToString().Contains(SearchTIN.Text)).ToList();

}

private void SearchTIN\_PreviewTextInput(object sender, TextCompositionEventArgs e)

{

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

{

e.Handled = true;

}}

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

{

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

Листинг 10 - «SubcriberPage»

Листинг разметки окна «PlaceCompletionPage» представлен в листинге 11-12.

using System;

using System.Collections.Generic;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

Листинг 11 - «PlaceCompletionPage»

using System.Windows.Shapes;

using WaterSupply.Lists.Changeses;

namespace WaterSupply.Lists

{

/// <summary>

/// Логика взаимодействия для PlaceCompletionPage.xaml

/// </summary>

public partial class PlaceCompletionPage : Page

{

public PlaceCompletionPage(){

InitializeComponent();

dgPlace.ItemsSource = DataBase.a.Place\_execution.ToList();}

private void btnDob\_Click(object sender, RoutedEventArgs e){

NavigationService.Content = new ChengesPlaceComp();}

private void btnEzm\_Click(object sender, RoutedEventArgs e){

try{

if (dgPlace.SelectedItems.Count > 0){

var tablplace = dgPlace.SelectedItem as Place\_execution;

NavigationService.Content = new ChengesPlaceComp(tablplace);

}

else MessageBox.Show("Выберите строку для изменения.", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);}}

private void SubBtn\_Click(object sender, RoutedEventArgs e){

NavigationService.Content = new SubscriberPage();

}

private void ContractPG\_Click(object sender, RoutedEventArgs e){

NavigationService.Content = new ContractPage();}

private void btnDelete\_Click(object sender, RoutedEventArgs e){

try{

if (MessageBox.Show("Вы уверены, что хотите удалить запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

if (dgPlace.SelectedItems.Count > 0){

var a = dgPlace.SelectedItem as Place\_execution;

DataBase.a.Place\_execution.Remove(a);

DataBase.a.SaveChanges();

dgPlace.ItemsSource = DataBase.a.Place\_execution.ToList();}

else MessageBox.Show("Выберите строку для удаления.", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Information);}}

catch (Exception ex){

MessageBox.Show(ex.Message);}}

private void SortName\_Click(object sender, RoutedEventArgs e){

dgPlace.ItemsSource = DataBase.a.Place\_execution.OrderBy(c => c.Title).ToList();}

private void SortNameOrg\_Click(object sender, RoutedEventArgs e){

dgPlace.ItemsSource = DataBase.a.Place\_execution.OrderBy(c => c.Subscriber.Name\_organization).ToList();}

private void SearchNumPlace\_TextChanged(object sender, TextChangedEventArgs e){

dgPlace.ItemsSource = DataBase.a.Place\_execution.Where(p => p.Title.ToString().Contains(SearchNumPlace.Text)).ToList();

}

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

{

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

Листинг 12 - «PlaceCompletionPage»

Листинг разметки окна «ContractPage» представлен в листинге 12-16.

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WaterSupply.Lists.Changeses;

using word = Microsoft.Office.Interop.Word;

using excel = Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Excel;

using Microsoft.Office.Interop.Word;

using Page = System.Windows.Controls.Page;

using System.Text.RegularExpressions;

using System.IO;

namespace WaterSupply.Lists

{

/// <summary>

/// Логика взаимодействия для ContractPage.xaml

/// </summary>

public partial class ContractPage : Page

{

public ContractPage()

{

InitializeComponent();

dgCont.ItemsSource = DataBase.a.Contract.ToList();

}

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

{

NavigationService.Content = new VolumePage();

}

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

{

NavigationService.Content = new WaterQualityPage();

}

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

{

NavigationService.Content = new ChangesContract();

}

private void btnEzm\_Click(object sender, RoutedEventArgs e){

try

{

if (dgCont.SelectedItems.Count > 0)

{

var tablcont = dgCont.SelectedItem as Contract;

NavigationService.Content = new ChangesContract(tablcont);

}

else MessageBox.Show("Выберите строку для изменения.", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

Листинг 12 - «ContractPage»

{

MessageBox.Show(ex.Message);

}

}

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

{

NavigationService.Content = new SubscriberPage();

}

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

{

try

{

if (MessageBox.Show("Вы уверены, что хотите удалить запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

if (dgCont.SelectedItems.Count > 0)

{

var a = dgCont.SelectedItem as Contract;

DataBase.a.Contract.Remove(a);

DataBase.a.SaveChanges();

dgCont.ItemsSource = DataBase.a.Contract.ToList();

}

else MessageBox.Show("Выберите строку для удаления.", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

dgCont.ItemsSource = DataBase.a.Contract.OrderBy(c => c.Finish\_Date).ToList();

}

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

{

dgCont.ItemsSource = DataBase.a.Contract.OrderBy(c => c.Payment).ToList();

}

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

{

dgCont.ItemsSource = DataBase.a.Contract.Where(p => p.Payment.ToString().Contains(SearchPay.Text)).ToList();

}

private void BtnWord\_Click(object sender, RoutedEventArgs e){

try {

if (dgCont.SelectedItems.Count > 0)

{

{

SaveFileDialog sfd = new SaveFileDialog();

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

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

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

doc.Activate();

try

{

Листинг 13 - «ContractPage»

var x = dgCont.SelectedItem as Contract;

if (sfd.ShowDialog() == false){

doc.Close();

doc = null;

app.Quit();

return;}

var results = DataBase.a.Contract.Join(DataBase.a.Drinking\_water\_quality\_indicator, contract => contract.Contract\_number, ind => ind.Indicator\_number,

(contract, ind) => new{

contract.Contract\_number,

contract.Signature\_date,

contract.Payment,

contract.Finish\_Date,

contract.Indicator\_number,

contract.Volume\_number,

contract.Code\_Employee,

contract.Subscribers\_TIN,

ind.Indicators,

ind.Units\_measurement,

ind.Regulations,

ind.Indicator\_harmfulness,

ind.Hazard\_class

}).Join(DataBase.a.Volume\_of\_drinking\_water\_consumption, contract => contract.Volume\_number, vol => vol.Volume\_number,

(contract, vol) => new{

contract.Contract\_number,

contract.Signature\_date,

contract.Payment,

contract.Finish\_Date,

contract.Indicator\_number,

contract.Volume\_number,

contract.Code\_Employee,

contract.Subscribers\_TIN,

contract.Indicators,

contract.Units\_measurement,

contract.Regulations,

contract.Indicator\_harmfulness,

contract.Hazard\_class,

vol.Total\_volume,

vol.Price,

vol.Total\_number\_months

}).Join(DataBase.a.Employee, contract => contract.Code\_Employee, emp => emp.Code\_Employee,

(contract, emp) => new{

contract.Contract\_number,

contract.Signature\_date,

contract.Payment,

contract.Finish\_Date,

contract.Indicator\_number,

contract.Volume\_number,

contract.Code\_Employee,

contract.Subscribers\_TIN,

contract.Indicators,

contract.Units\_measurement,

contract.Regulations,

contract.Indicator\_harmfulness,

contract.Hazard\_class,

contract.Total\_volume,

contract.Price,

contract.Total\_number\_months,

emp.Surname,

Листинг 14 - «ContractPage»

emp.NameEmployee,

emp.Midllename,emp.Post,

emp.Number\_Phone

}).Join(DataBase.a.Subscriber, contract => contract.Subscribers\_TIN, sub => sub.Subscribers\_TIN,

(contract, sub) => new{

contract.Contract\_number,

contract.Signature\_date,

contract.Payment,

contract.Finish\_Date,

contract.Indicator\_number,

contract.Volume\_number,

contract.Code\_Employee,

contract.Subscribers\_TIN,

contract.Indicators,

contract.Units\_measurement,

contract.Regulations,

contract.Indicator\_harmfulness,

contract.Hazard\_class,

contract.Total\_volume,

contract.Price,

contract.Total\_number\_months,

contract.Surname,

contract.NameEmployee,

contract.Midllename,

contract.Post,

contract.Number\_Phone,

sub.Name\_organization,

sub.Checking\_account,

sub.Director,

sub.Address,

sub.Number,

sub.Correspondent\_account

}).Where(p => p.Contract\_number == x.Contract\_number).FirstOrDefault();

word.Bookmarks wB = doc.Bookmarks;

wB["НомерДоговора"].Range.Text = results.Contract\_number.ToString();

wB["НазваниеОрганизации"].Range.Text = results.Name\_organization.ToString();

wB["НомерИндекатора"].Range.Text = results.Indicator\_number.ToString();

wB["НомерОбъёма"].Range.Text = results.Volume\_number.ToString();

wB["Фамилия"].Range.Text = results.Surname.ToString();

wB["Имя"].Range.Text = results.NameEmployee.ToString();

wB["Отчество"].Range.Text = results.Midllename.ToString();

wB["Оплата"].Range.Text = results.Payment.ToString();

wB["НачальнаяДата"].Range.Text = results.Signature\_date.ToString();

wB["ОкончательнаяДата"].Range.Text = results.Finish\_Date.ToString();

wB["ДатаНачалаДоговор"].Range.Text = results.Signature\_date.ToString();

wB["Приложение"].Range.Text = results.Volume\_number.ToString();

doc.SaveAs2(sfd.FileName);

doc.Close();

doc = null;

app.Quit();

MessageBox.Show("Файл успешно создан");}

Листинг 15 - «ContractPage»

catch (Exception ex){

MessageBox.Show(ex.Message);

doc.Close();

doc = null;

app.Quit();}}}

else MessageBox.Show("Выберите строку для создания word документа.", "Ошибка!", MessageBoxButton.OK, MessageBoxImage.Information);}

catch (Exception ex){

MessageBox.Show(ex.Message);}}

private void SearchPay\_PreviewTextInput(object sender, TextCompositionEventArgs e){

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

e.Handled = true;}}

private void BtnChart\_Click(object sender, RoutedEventArgs e){

NavigationService.Content = new ChartPage();}

private void BtnExcel\_Click(object sender, RoutedEventArgs e){

if (dgCont.SelectedItems.Count <= 0){

MessageBox.Show("Выберите поле!", "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] = "Общий объем";

wS.Cells[4, 5] = "Стоимость";

var x = dgCont.SelectedItem as Contract;

try{

if (sfd.ShowDialog() == false){

wB.Close();

wB = null;

app.Quit();

return;}

List<Contract> contracts = DataBase.a.Contract.Where(c => c.Contract\_number == x.Contract\_number).ToList();

dgCont.ItemsSource = contracts;

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

wS.Cells[i + 5, 1].Value = contracts[i].Contract\_number;

wS.Cells[i + 5, 2].Value = contracts[i].Signature\_date;

wS.Cells[i + 5, 3].Value = contracts[i].Finish\_Date;

wS.Cells[i + 5, 4].Value = contracts[i].Volume\_of\_drinking\_water\_consumption.Total\_volume;

wS.Cells[i + 5, 5].Value = contracts[i].Payment;}

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();}}}}}

Листинг 16 - «ContractPage»

Листинг разметки окна «ChangesContractPage» представлен в листинге 17-18.

using Microsoft.Win32;

using System;

Листинг 17 - «ChangesContractPage»

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WaterSupply.Lists.Changeses{

/// <summary>

/// Логика взаимодействия для ChangesContract.xaml

/// </summary>

public partial class ChangesContract : Page{

public ChangesContract(){

InitializeComponent();

NameOrg.ItemsSource = DataBase.a.Subscriber.ToList();

NumInd.ItemsSource = DataBase.a.Drinking\_water\_quality\_indicator.ToList();

NumVol.ItemsSource = DataBase.a.Volume\_of\_drinking\_water\_consumption.ToList();

CbName.ItemsSource = DataBase.a.Employee.ToList();

CbLast.ItemsSource = DataBase.a.Employee.ToList();

CbOtch.ItemsSource = DataBase.a.Employee.ToList();

\_con = new Contract();}

public ChangesContract(Contract contract){

InitializeComponent();

NameOrg.ItemsSource = DataBase.a.Subscriber.ToList();

NumInd.ItemsSource = DataBase.a.Drinking\_water\_quality\_indicator.ToList();

NumVol.ItemsSource = DataBase.a.Volume\_of\_drinking\_water\_consumption.ToList();

CbName.ItemsSource = DataBase.a.Employee.ToList();

CbLast.ItemsSource = DataBase.a.Employee.ToList();

CbOtch.ItemsSource = DataBase.a.Employee.ToList();

\_con = contract;}public Contract \_con;

private void NOk\_Click(object sender, RoutedEventArgs e){

NavigationService.Content = new ContractPage();}

private void Ok\_Click(object sender, RoutedEventArgs e){

try{

if (MessageBox.Show("Вы уверены, что хотите изменить запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

if (\_con.Contract\_number == 0){

DataBase.a.Contract.Add(\_con);

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

DataBase.a.SaveChanges();

NavigationService.Navigate(new ContractPage());}}

catch (Exception){

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

private void NumCon\_PreviewTextInput(object sender, TextCompositionEventArgs e){

if (!Char.IsDigit(e.Text, 0)){e.Handled = true;}}

private void Pay\_PreviewTextInput(object sender, TextCompositionEventArgs e){if (!Char.IsDigit(e.Text, 0)){e.Handled = true;}}

private void Page\_Loaded(object sender, RoutedEventArgs e){

this.DataContext = \_con;}}}

Листинг 18 - «ChangesContractPage»

Листинг разметки окна «ChangesSubcribertPage» представлен в листинге 19.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WaterSupply.Lists.Changeses{

/// <summary>

/// Логика взаимодействия для ChangesesSubscriberPage.xaml

/// </summary>

public partial class ChangesesSubscriberPage : Page{

public ChangesesSubscriberPage(){

InitializeComponent();

\_sub = new Subscriber();}

public ChangesesSubscriberPage(Subscriber subscriber){

InitializeComponent();

\_sub = subscriber;

TINSub.Visibility = Visibility.Hidden;

TN.Visibility = Visibility.Hidden;}

public Subscriber \_sub;

private void NOk\_Click(object sender, RoutedEventArgs e){

NavigationService.Content = new SubscriberPage();}

private void Ok\_Click(object sender, RoutedEventArgs e) {

try{

if (MessageBox.Show("Вы уверены, что хотите изменить запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes){

if (\_sub.ID\_Sub == 0){

DataBase.a.Subscriber.Add(\_sub);

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

DataBase.a.SaveChanges();

NavigationService.Navigate(new SubscriberPage());}}

catch (Exception){

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

private void TINSub\_PreviewTextInput(object sender, TextCompositionEventArgs e){

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

e.Handled = true;}}

private void Dir\_PreviewTextInput(object sender, TextCompositionEventArgs e){

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

if (!inp.IsMatch(e.Text) && (e.Text == ",") && (e.Text == ",")

&& (!Dir.Text.Contains(","))){

e.Handled = true;}}

private void Page\_Loaded(object sender, RoutedEventArgs e){

this.DataContext = \_sub;

}}

}

Листинг 19 - «ChangesSubcribertPage»

Листинг разметки окна «ChangesPlacePage» представлен в листинге 20.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WaterSupply.Lists.Changeses{

/// <summary>

/// Логика взаимодействия для ChengesPlaceComp.xaml

/// </summary>

public partial class ChengesPlaceComp : Page

{

public ChengesPlaceComp()

{

InitializeComponent();

NameOrg.ItemsSource = DataBase.a.Subscriber.ToList();

\_plac = new Place\_execution();}

public ChengesPlaceComp(Place\_execution place)

{

InitializeComponent();

NameOrg.ItemsSource = DataBase.a.Subscriber.ToList();

\_plac = place;}

public Place\_execution \_plac;

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

{

NavigationService.Content = new PlaceCompletionPage();}

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

{

try

{

if (MessageBox.Show("Вы уверены, что хотите изменить запись?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

if (\_plac.Place\_execution\_number == 0)

{

DataBase.a.Place\_execution.Add(\_plac);

MessageBox.Show("Место исполнения добавлен!", "Внимание!", MessageBoxButton.OK, MessageBoxImage.Information);}

DataBase.a.SaveChanges();

NavigationService.Navigate(new PlaceCompletionPage());}}

catch (Exception){

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

private void NumPlace\_PreviewTextInput(object sender, TextCompositionEventArgs e){

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

e.Handled = true;}}

private void Page\_Loaded(object sender, RoutedEventArgs e){

this.DataContext = \_plac;

}}

}

Листинг 20 - «ChangesPlacePage»

## 2.5.Разработка тестовых данных

Информация по тесту №1 представлена рисунке 18.

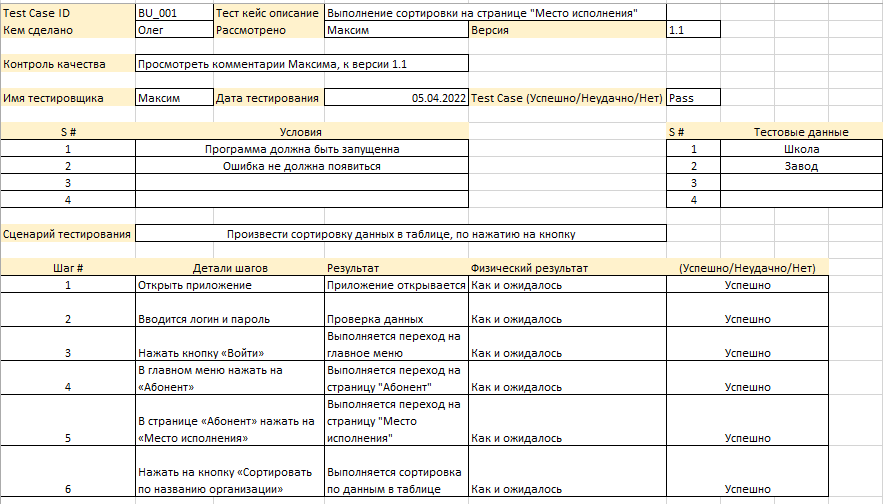


Рис. 18 – Тест № 1

Информация по тесту №2 представлена рисунке 19.

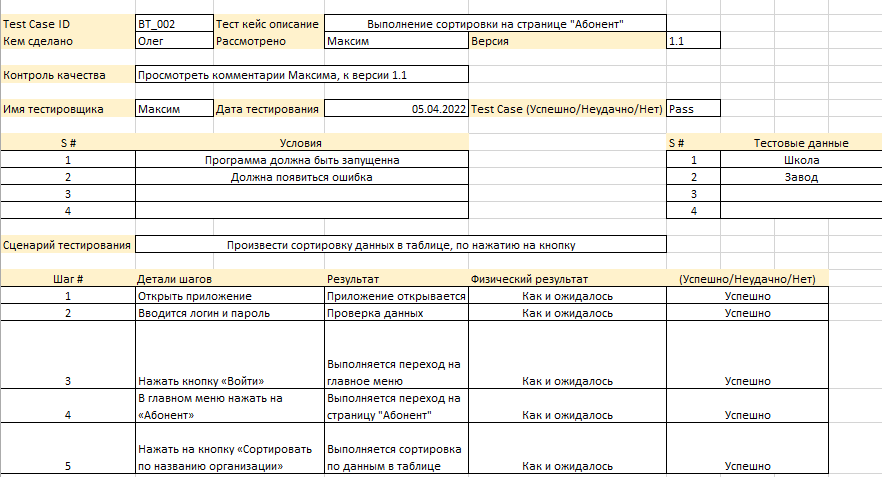


Рис. 19 – Тест № 2

Информация по тесту №3 представлена рисунке 20.

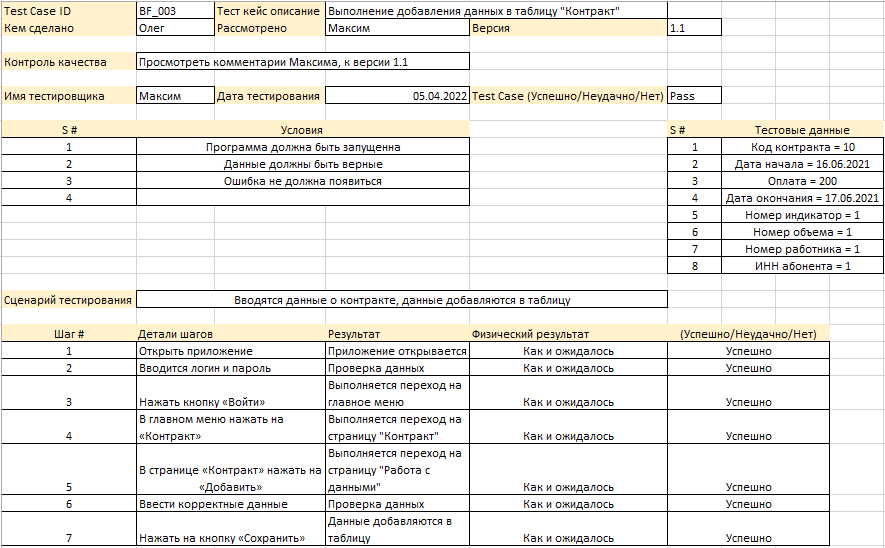


Рис. 20 – Тест № 3

Информация по тесту №4 представлена рисунке 21.

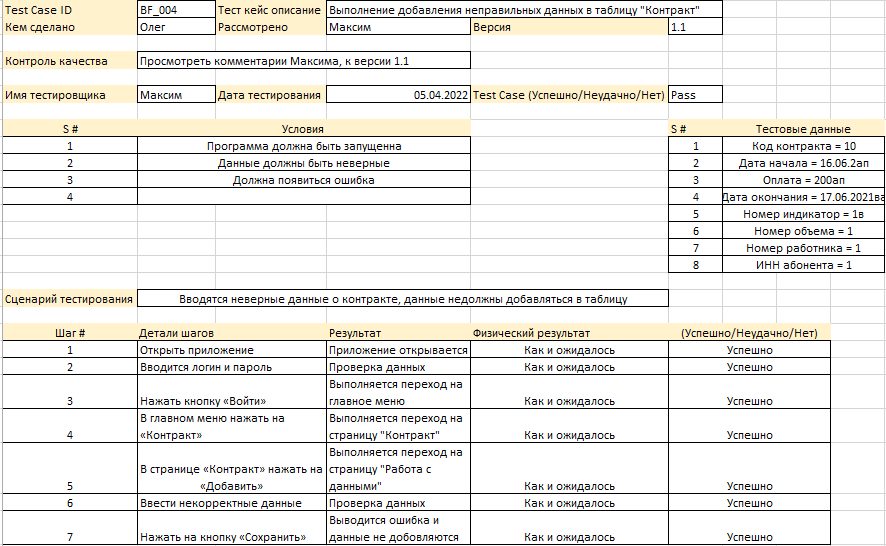


Рис. 21 – Тест № 4

Информация по тесту №5 представлена рисунке 22.

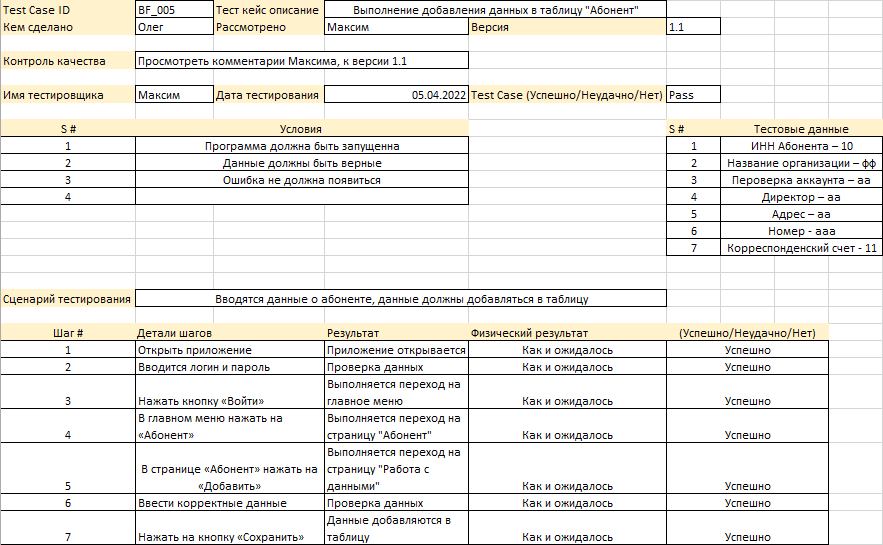


Рис. 22 – Тест № 5

Информация по тесту №6 представлена рисунке 23.

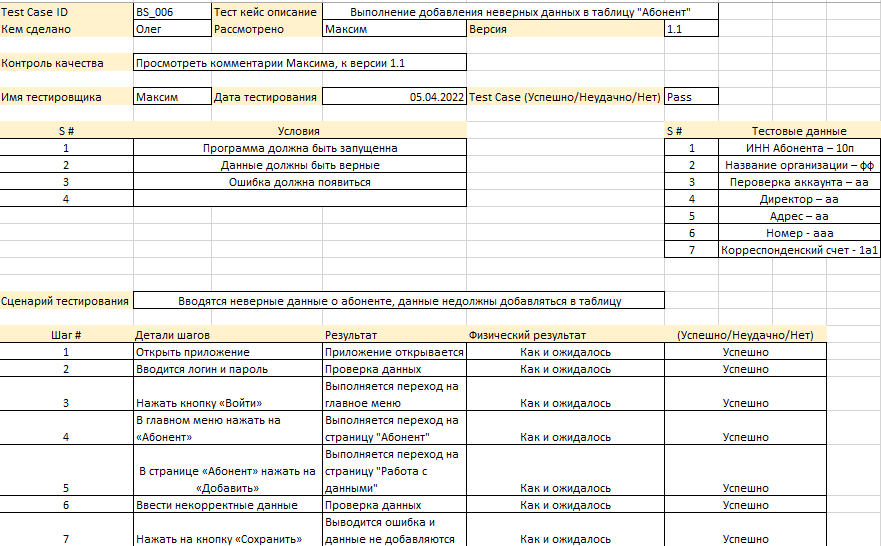


Рис. 23 – Тест № 6

Информация по тесту №7 представлена рисунке 24.

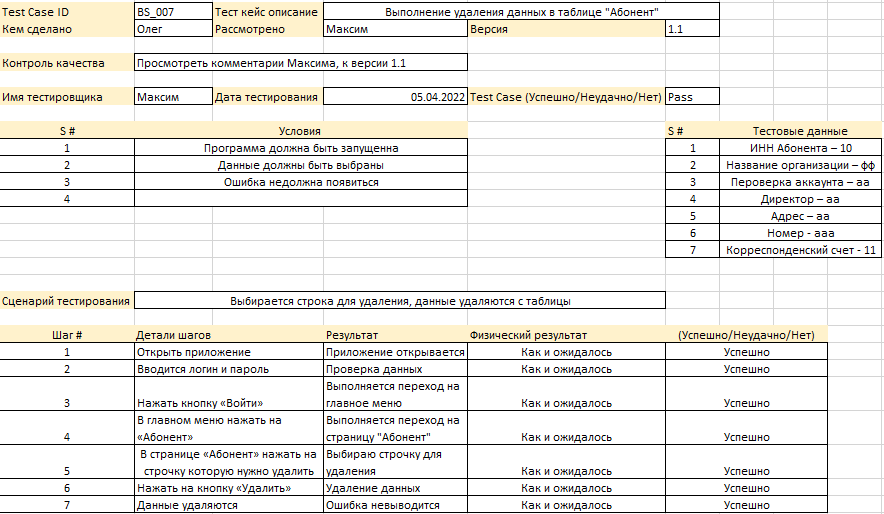


Рис. 24 – Тест № 7

Информация по тесту №8 представлена рисунке 25.

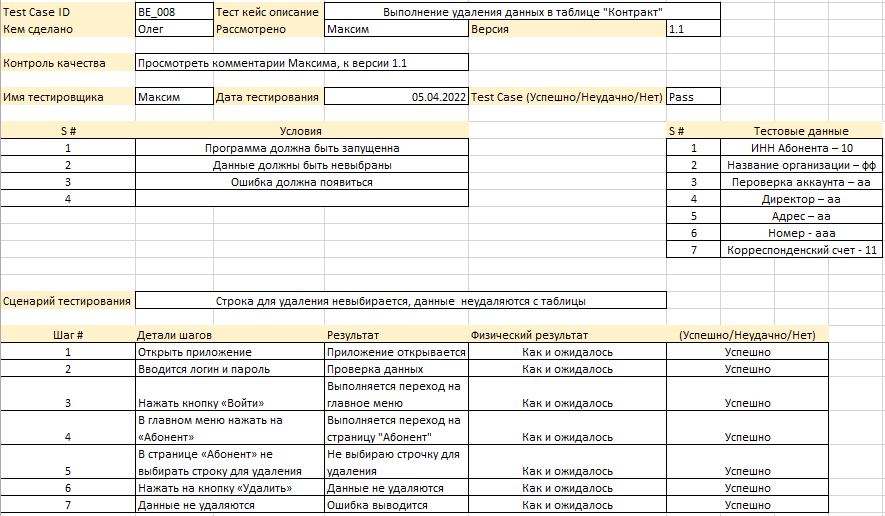


Рис. 25 – Тест № 8

Информация по тесту №9 представлена рисунке 26.

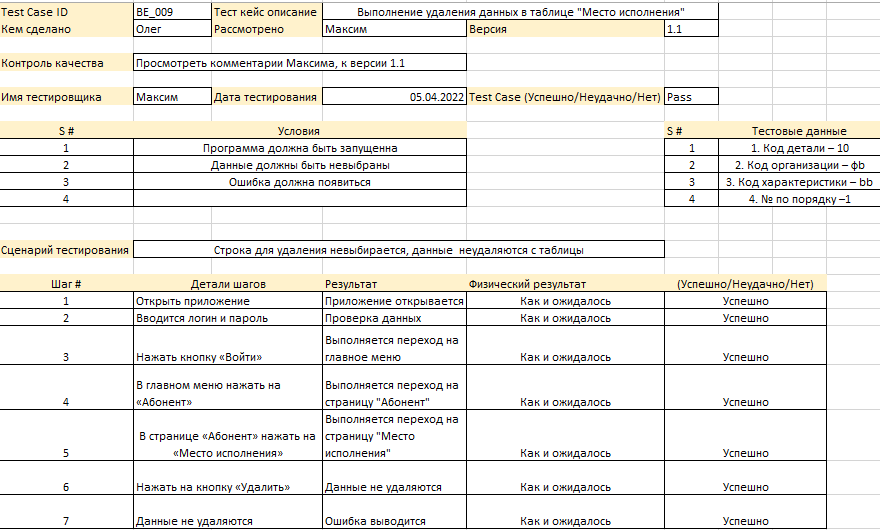


Рис. 26 – Тест № 9

Информация по тесту №10 представлена рисунке 27.

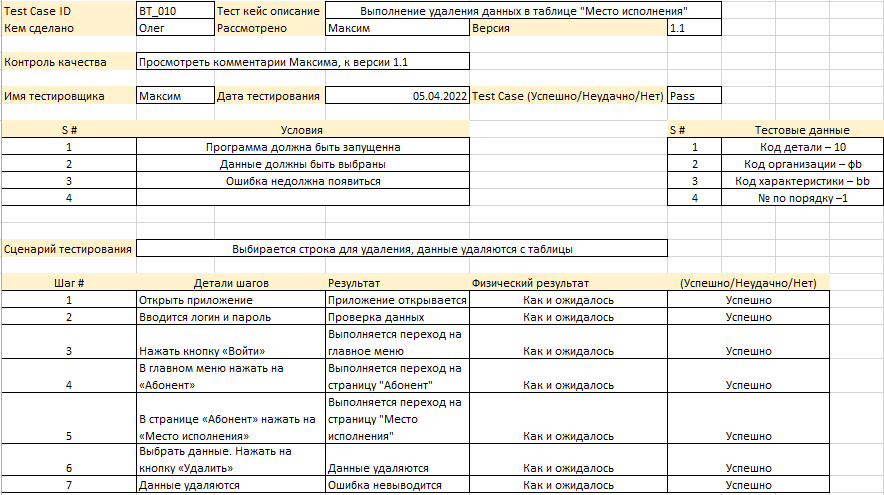


Рис. 27 – Тест № 10

Информация по тесту №11 представлена рисунке 28.

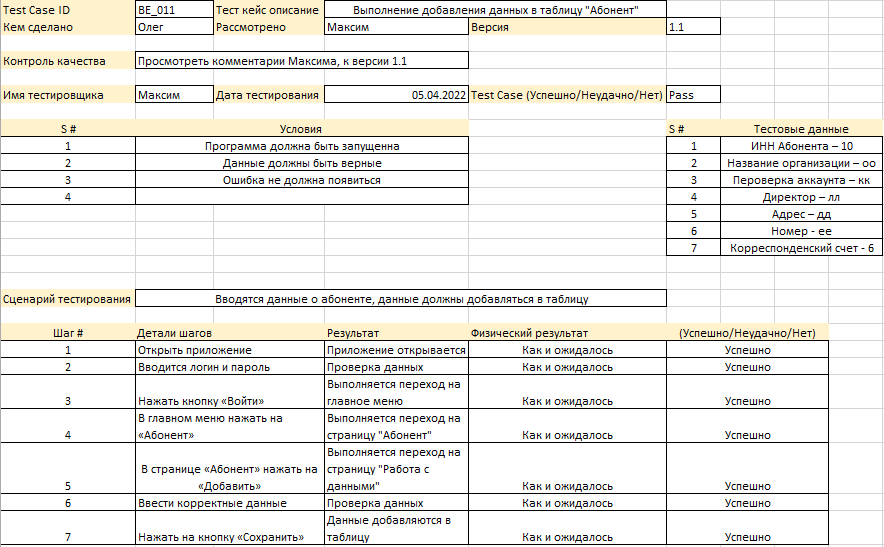


Рис. 28 – Тест № 11

Информация по тесту №12 представлена рисунке 29.

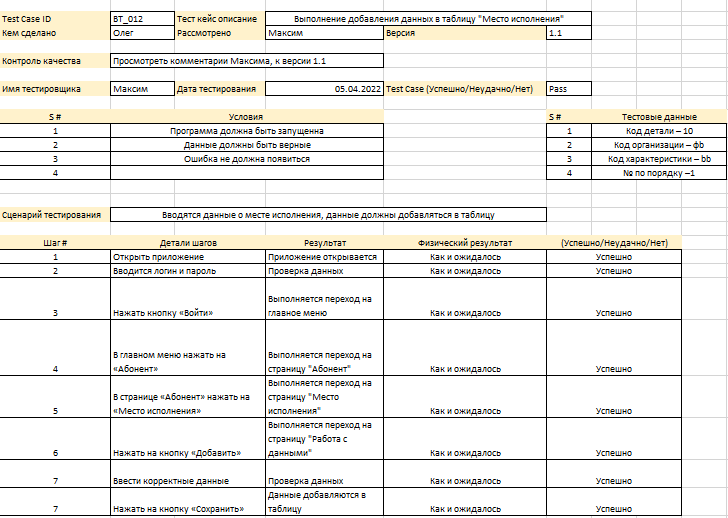


Рис. 29 – Тест № 12

Информация по тесту №13 представлена рисунке 30.

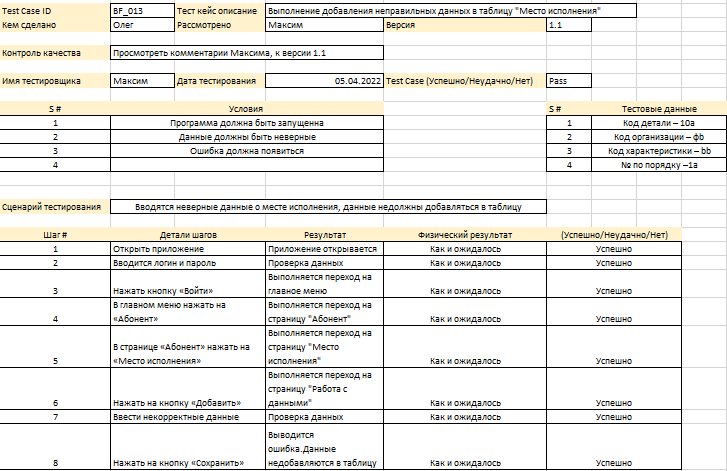


Рис. 30 – Тест № 13

Информация по тесту №14 представлена рисунке 31.

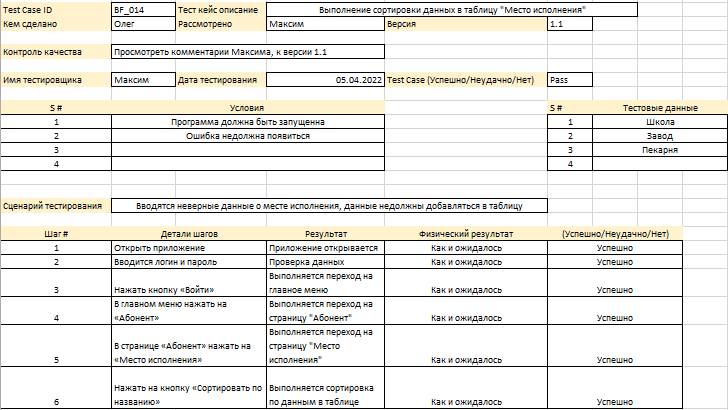


Рис. 31 – Тест № 14

Информация по тесту №15 представлена рисунке 32.

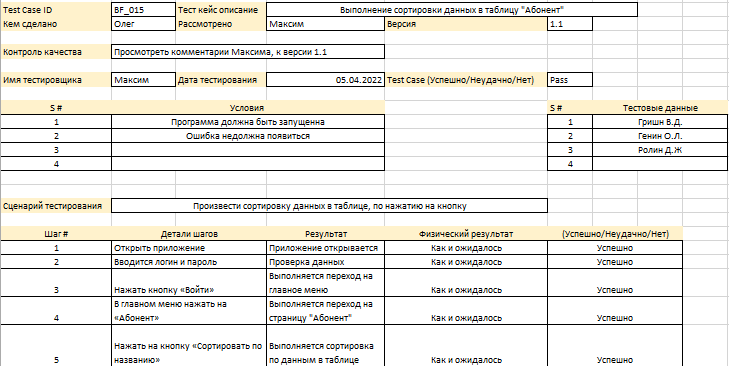


Рис. 15 – Тест № 32

Информация по тесту №16 представлена рисунке 33.

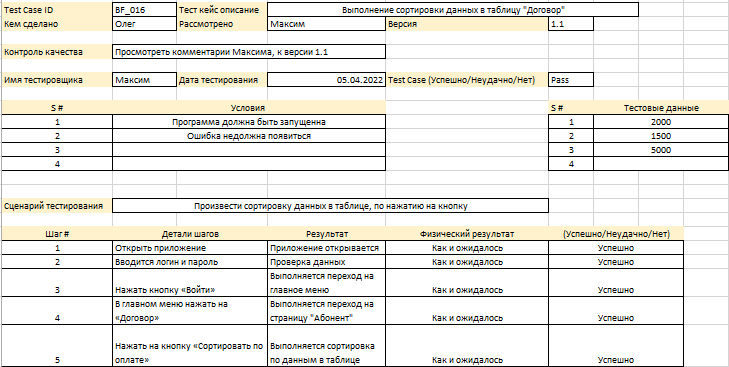


Рис. 33 – Тест № 16

Информация по тесту №17 представлена рисунке 34.

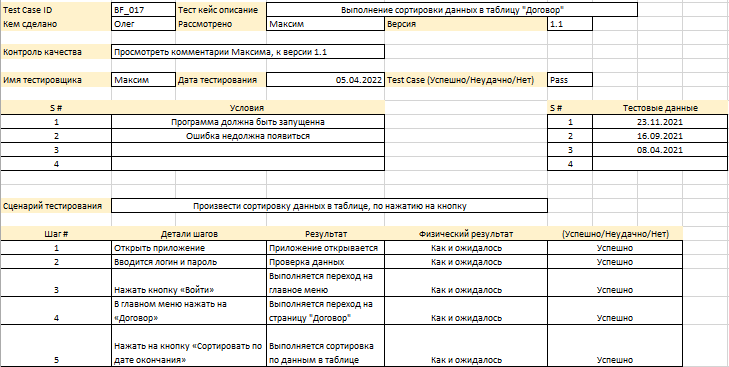


Рис. 34 – Тест № 17

Информация по тесту №18 представлена рисунке 35.

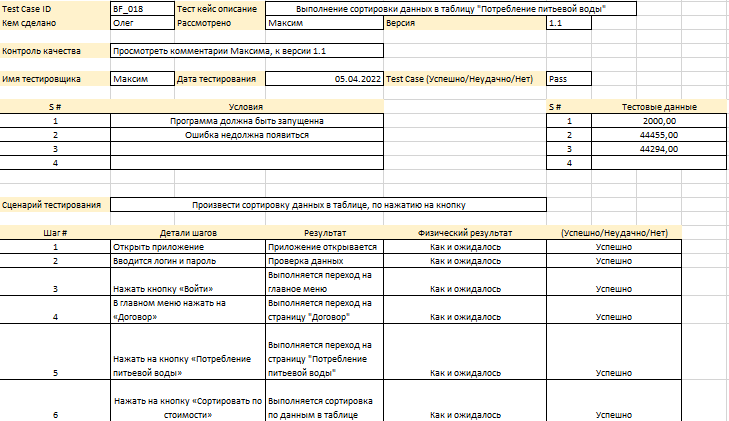


Рис. 35 – Тест № 18

Информация по тесту №19 представлена рисунке 36.

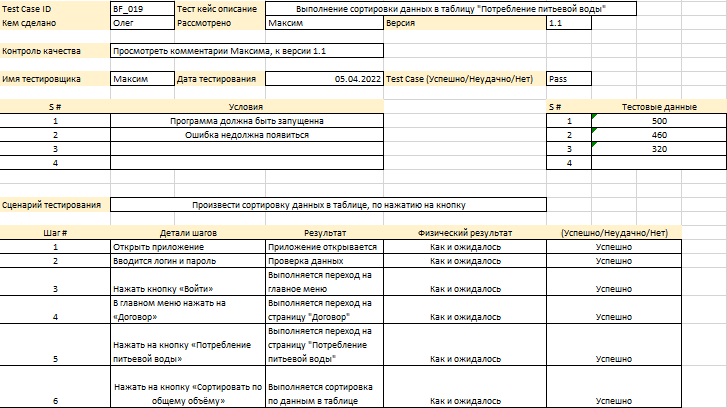


Рис. 36 – Тест № 19

Информация по тесту №20 представлена рисунке 37.

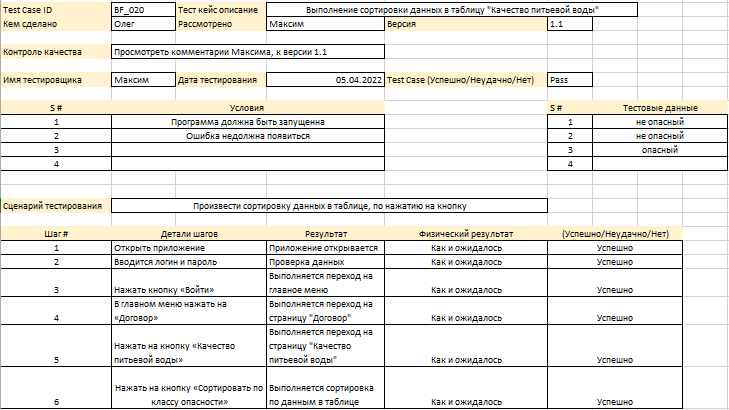


Рис. 37 – Тест № 20

Результат выполнения тестов представлене на рисунке 38

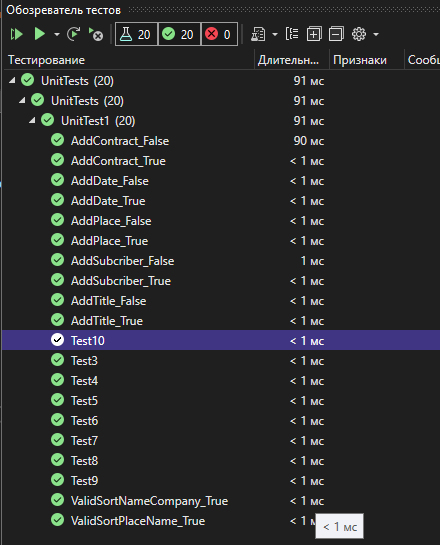


Рис. 38 – результат выполнения тестов

## 2.6. Результат работы программы

Результат работы программы представлен на рисунках 39-42



Рис. 39 – результат работы программы

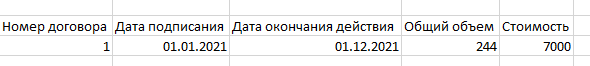


Рис. 40 – результат работы программы

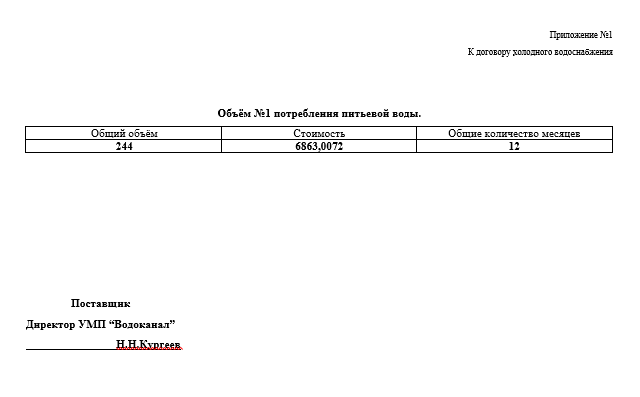


Рис. 41 – результат работы программы

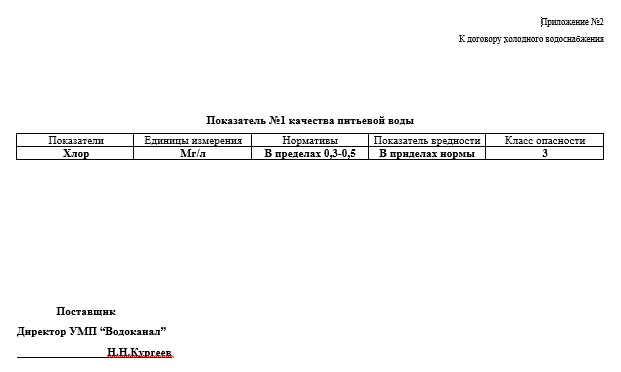


Рис. 42 – результат работы программы

## 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 21 – 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 одной роли «Сотрудник»

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 22 – 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 43.



Рис. 43 – Окно «Авторизация»

После успешной авторизации Вам откроет главное меню приложения. Интерфейс окна «Главное меню» представлен на рисунке 44



Рис. 44 – Окно «Главное меню»

Кнопка «Договор» открывает страницу с договорами.

Кнопка «Абонент» открывает страницу с данными о абонентах.

Кнопка «Свойства» открывает страницу с информацией о свойствах.

Кнопка «К авторизации» позволяет нам попасть на страницу авторизации.

**Лист «Договор»**

Для того чтобы перейти к данной странице, необходимо в главное меню нажать на кнопку «Договор». Интерфейс данного листа представлен на рисунке 45.

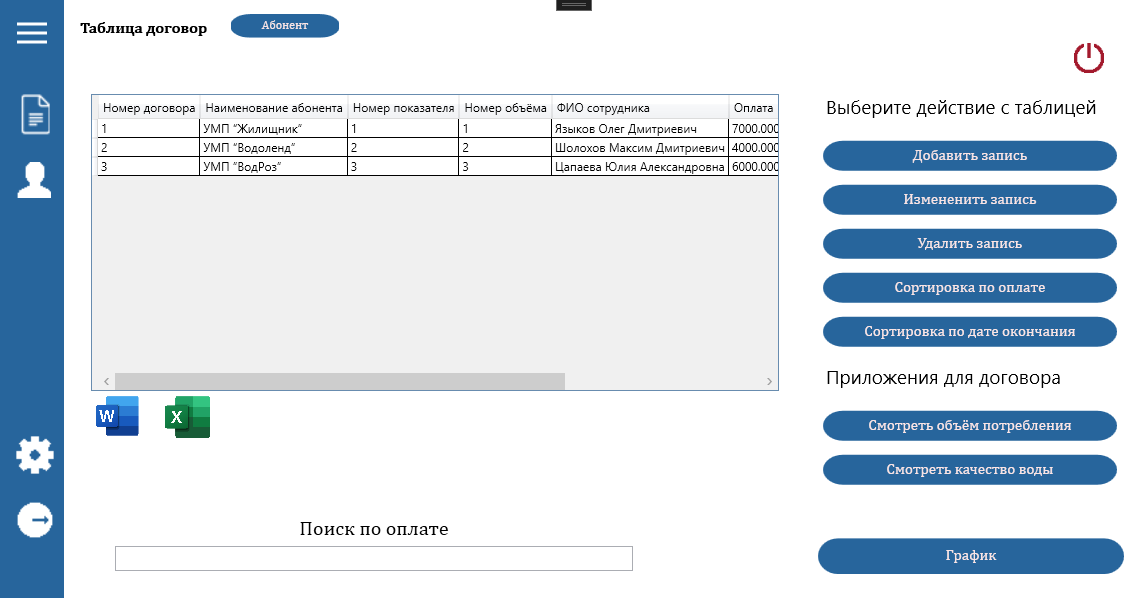


Рис. 45 – Лист «Договор»

Здесь Вы можете добавлять информацию о договоре, сортировать по оплате и дате окончания, выполнять фильтрацию, а также выводить документ в формате .docx, .xlsx.

**Лист «Добавление/Изменение договора»**

Для того чтобы перейти к данной странице, необходимо на странице «Договор» нажать на кнопку «Добавить». Интерфейс данного листа представлен на рисунке 46.

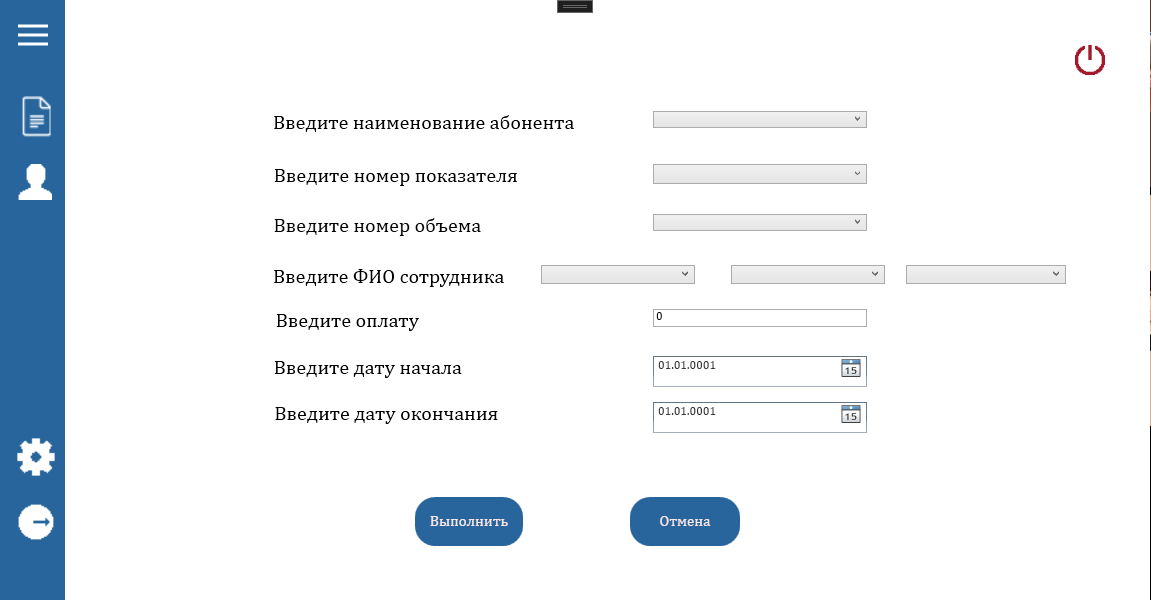


Рис. 46 – Лист «Добавление/Изменение договора»

Здесь Вам необходимо указать наименование абонента. Номер показателя и объема, ввести ФИО сотрудника. Сумму опалыт, дату начала и окончания. Для того чтобы данные были успешно сохранены, необходимо нажать на кнопку «Выполнить». Если хотите выйти с данной страницы, то нажмите на кнопку «Отмена».

**Лист «Объем потребления питьевой воды»**

Для того чтобы перейти к данному листу, необходимо на листе «Договор» нажать на кнопку «Смотреть объем потребления».

Интерфейс данного листа представлен на рисунке 47.



Рис. 47 – лист «Объем потребления питьевой воды»

Здесь вы можете просматривать информацию по объему потребления питьевой воды, так же отсортировать данные или произвести по ним поиск. На данному листе, так же возможно создать приложение для договора.

**Лист «Показатель качества питьевой воды»**

Для того чтобы перейти к данному листу, необходимо на листе «Договор» нажать на кнопку «Смотреть качество воды».

Интерфейс данного листа представлен на рисунке 48.

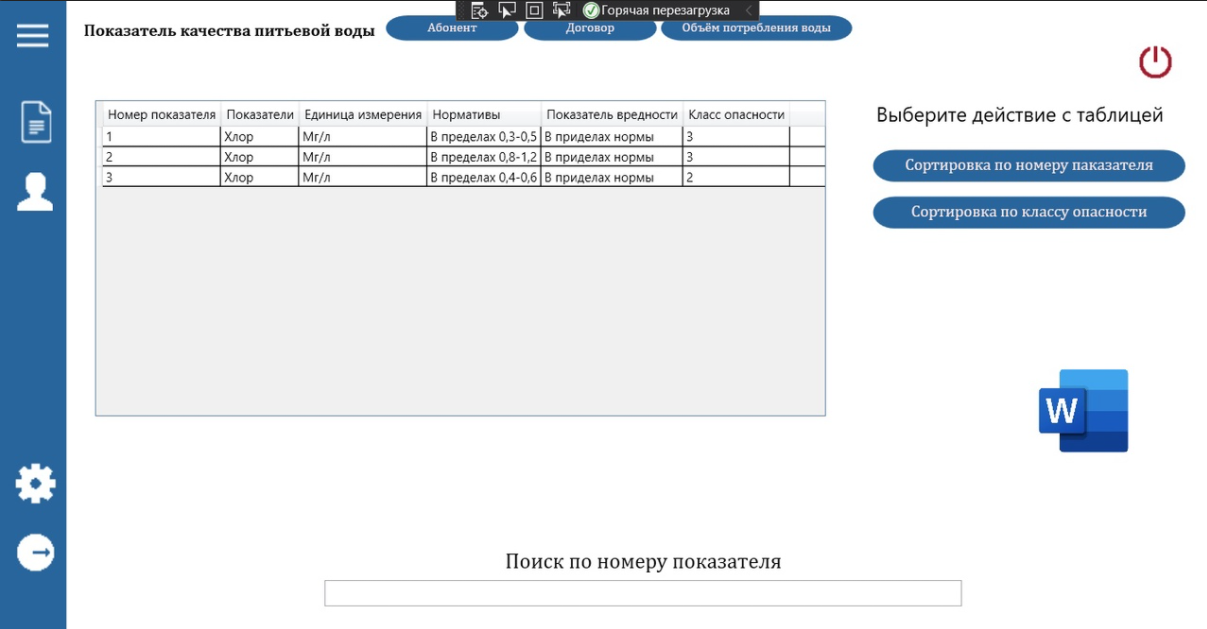


Рис. 48 – лист «Показатель качества воды»

На данном листе можно посмотреть данные таблицы, так же отсортировать эти данные и осуществить поиск по данным. И так же создать второе приложение для договора.

**Лист «Абонент»**

Для того чтобы перейти к данному листу, необходимо на главном окне нажать на кнопку «Абонент».

Интерфейс данного листа представлен на рисунке 49.

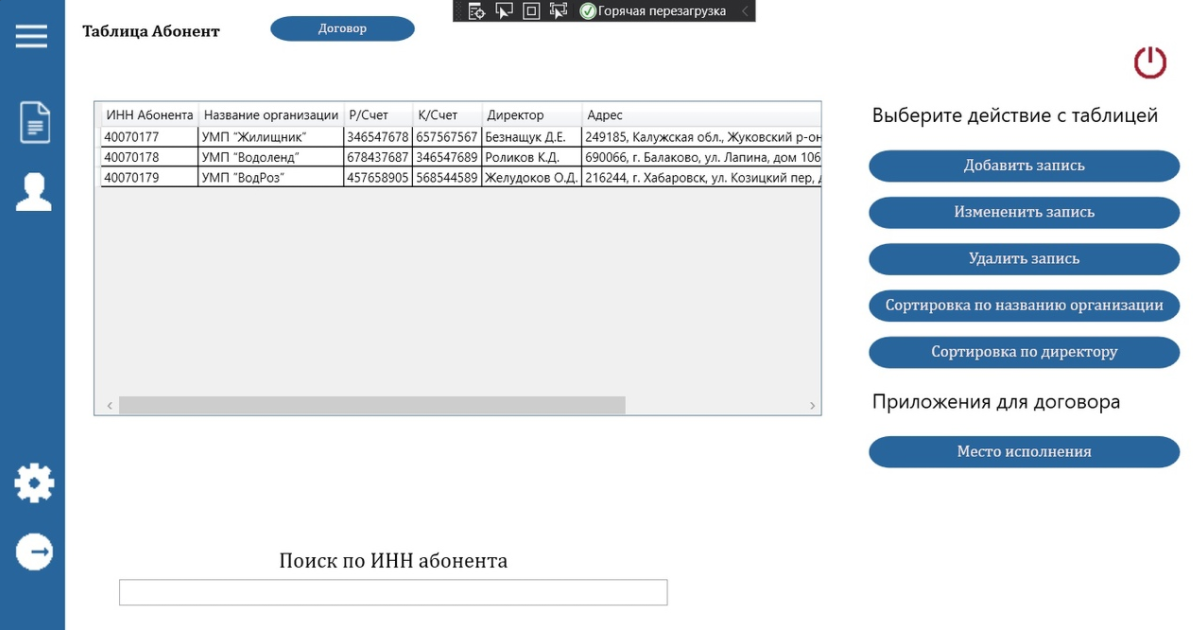


Рис. 49 – лист «Абонент»

Здесь вы можете увидеть информацию о абонентах, так же совершить добавление/изменение/удаление данных. Произвести сортировку данных и поиск по ним.

**Лист «Добавления/Изменения абонент»**

Для того чтобы перейти к данному листу, необходимо в листе «Абонент» нажать на кнопку «Добавить запись». Если необходимо изменить данные, то выбираете нужного абонента и затем нажимаете на кнопку «Изменить запись».

Интерфейс данного листа представлен на рисунке 50.

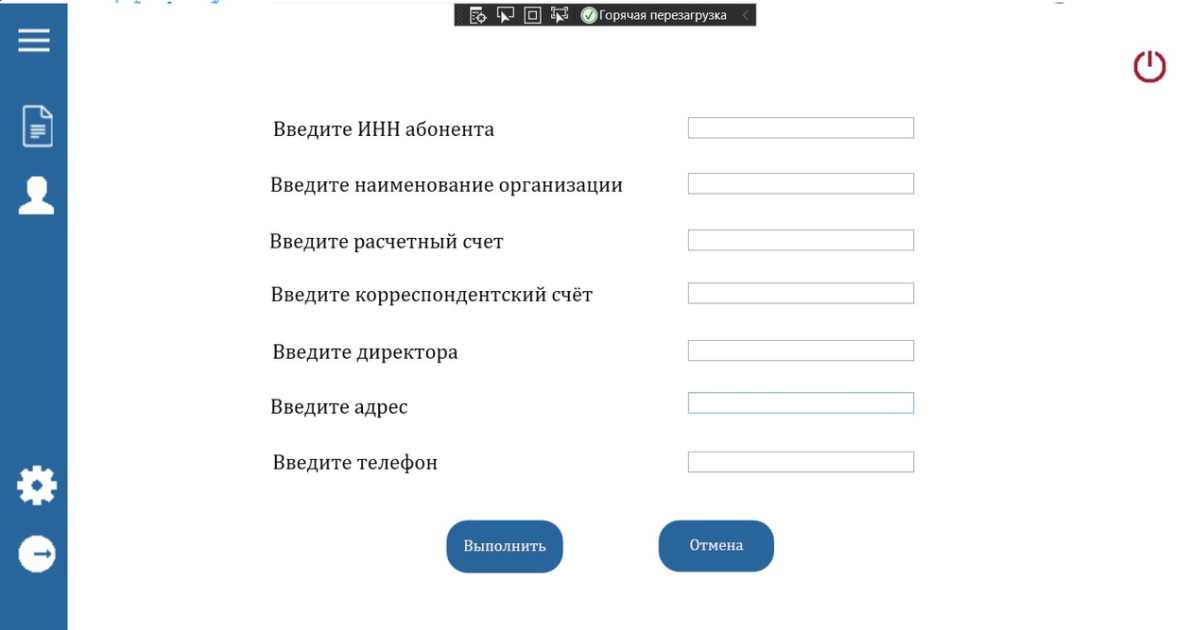


Рис. 50 – лист «Добавление/Изменение данных абонента»

Здесь Вам необходимо ввести данные абонента. Для сохранения данных необходимо нажать кнопку «Сохранить» после чего появиться уведомление. Если хотите выйти с данного листа необходимо нажать на кнопку «Назад».

**Лист «Место исполнения»**

Для того чтобы перейти к данному листу, необходимо на листе «Абонент» нажать на кнопку «Место исполнения».

Интерфейс данного листа представлен на рисунке 51.

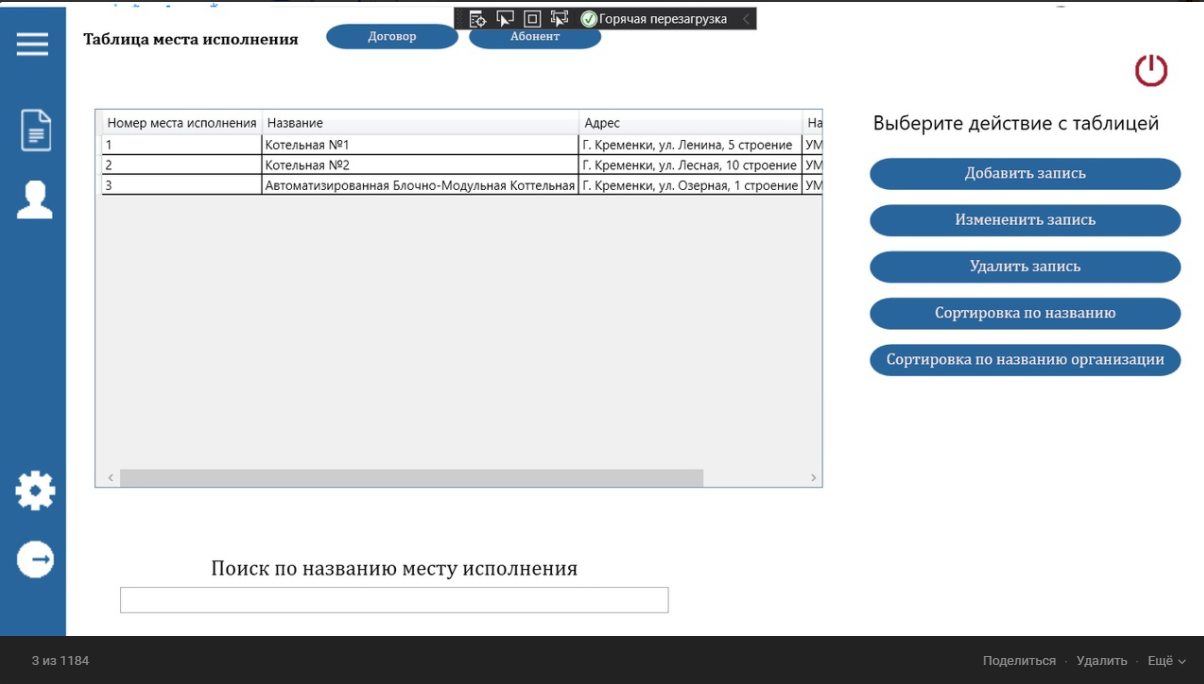


Рис. 51 – лист «Место исполнения»

Здесь вы можете увидеть информацию о месте исполнения, так же совершить добавление/изменение/удаление данных. Произвести сортировку данных и поиск по ним.

**Лист «Добавления/Изменения места исполнения»**

Для того чтобы перейти к данному листу, необходимо в листе «Место исполнения» нажать на кнопку «Добавить запись». Если необходимо изменить данные, то выбираете нужное место и затем нажимаете на кнопку «Изменить запись».

Интерфейс данного листа представлен на рисунке 52.

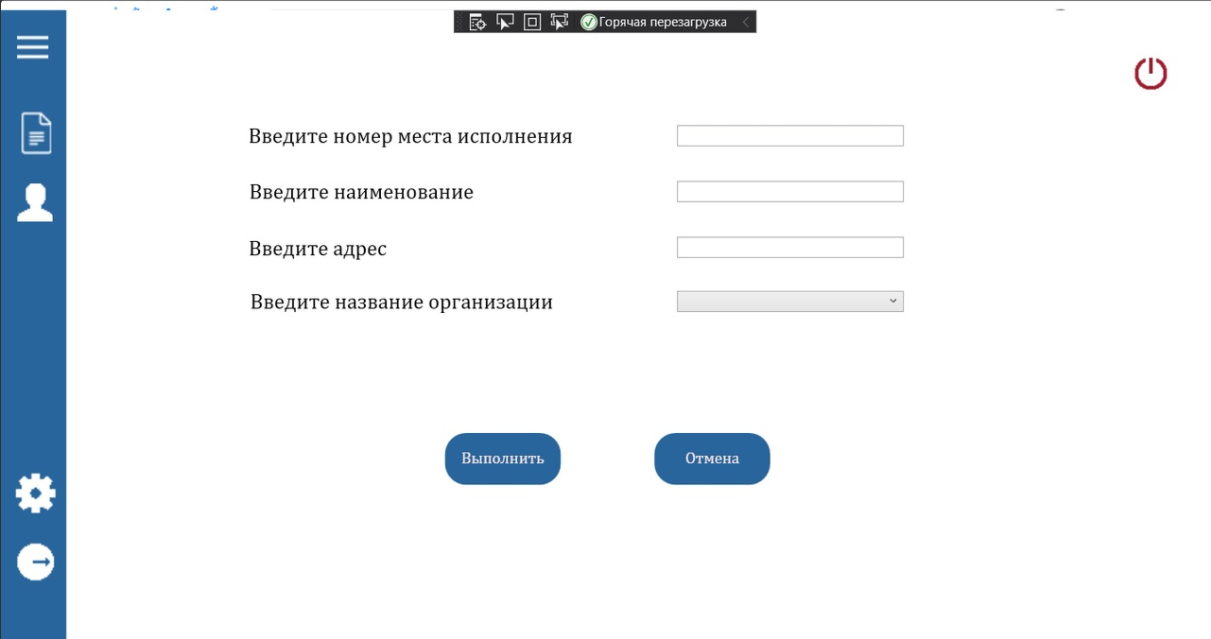


Рис. 52 – лист «Добавление/Изменение данных место исполнения»

Здесь Вам необходимо ввести данные места исполнения и выбрать наименование организации. Для сохранения данных необходимо нажать кнопку «Сохранить» после чего появиться уведомление. Если хотите выйти с данного листа необходимо нажать на кнопку «Назад».

**Лист «График»**

В программе так же реализован график, на котором показывается информация на месяц, где указаны прогнозы на поставку определенного объема воды, в определенную организацию.

Интерфейс данных листов представлен на рисунках 53.

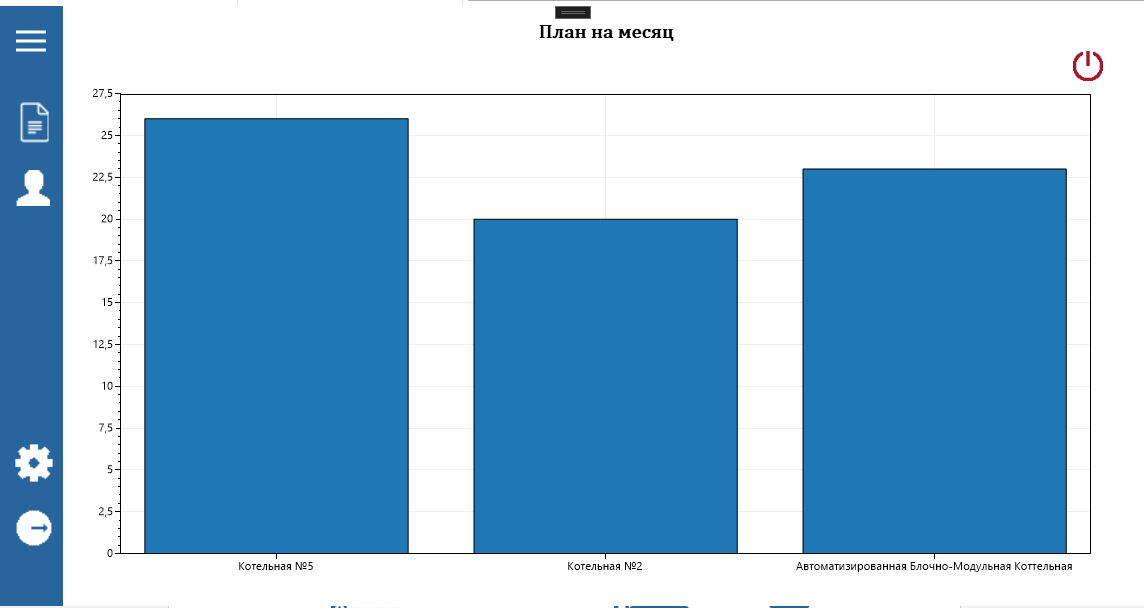


Рис. 53 – листов «График»

**Лист «Свойства»**

В программе так же реализованы листы с информацие о программе и так же руководство пользователя. Нажав на шестеренку в левой нижней часте програмы вы поподете в это меню.

Интерфейс данных листов представлен на рисунках 54-55.

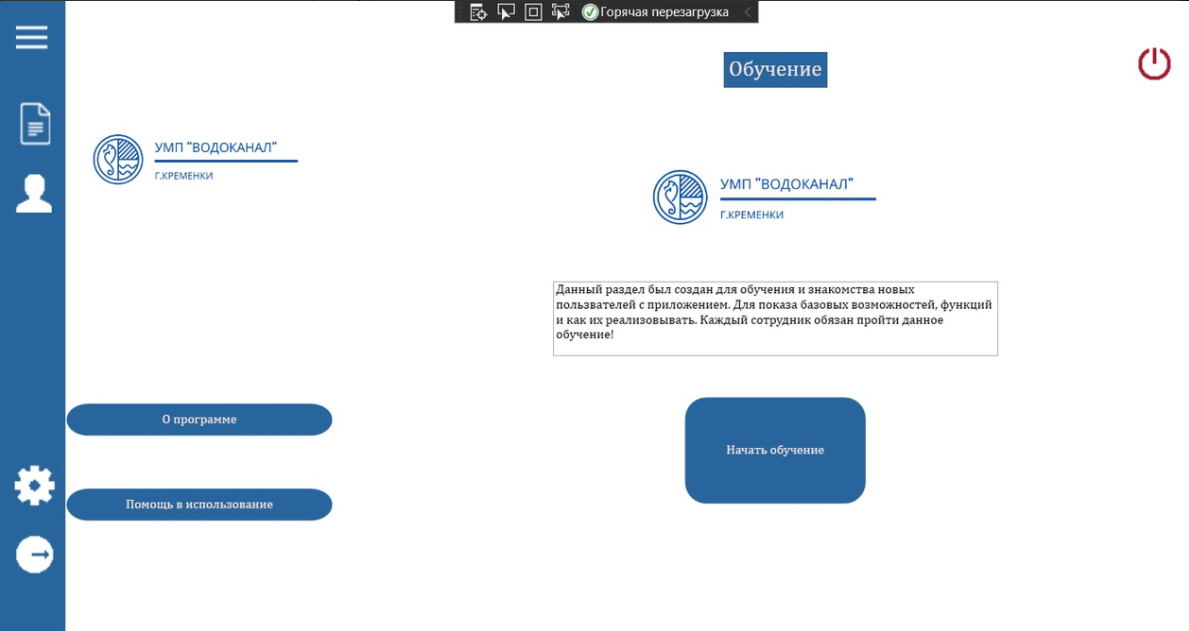
 

Рис. 54-55 – листов «Свойства»

Здесь вы можете узнать информацию о программе и так же обучиться работе с ней с помошью записанных гифф файлов и текстовых подсказок.

## 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 21.

public static string Encrypt(string value){

StringBuilder Sb = new StringBuilder();

using (SHA256 hash = SHA256Managed.Create()){

Encoding enc = Encoding.UTF8;

Byte[] result = hash.ComputeHash(enc.GetBytes(value));

foreach (Byte b in result)

Sb.Append(b.ToString("x2"));}

return Sb.ToString();

}

Листинг 21 – шифрование пароля

## 2.9. Эксплуатация и развитие программы.

Данное приложение было разработано непосредственно для организации УК «Жилищник» для улучшения автоматизации процесса.  
До этого момента работа по введению и заключению договоров осуществлялась в ручной форме, но теперь, благодаря программе “WaterSupply” 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 догворов водоснабжения.

Первым этапом был произведен анализ актуальности темы и сравнения уже существующих разработок по данной теме, а именно «СборРос», «Zalondor», «РосПост». 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 только одним пользователем. Согласно роли – сотрудником, ему доступен следующий функционал:

– просмотр информации по абонентам;

– добавление новой информации по абонентам

– возможность редактирования и удаления данных;

– просмотр информации по договору;

– добавление нового договора;

– возможность удаления информации;

– формирования договора;

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

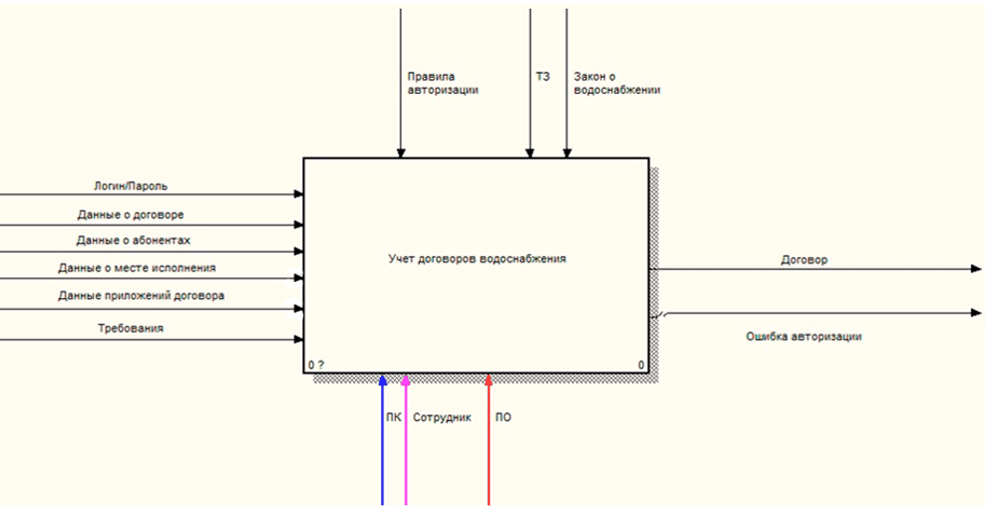


Рис.56 - IDEF0

ОКОНЧАНИЕ ПРИЛОЖЕНИЕ 1 – IDEF0

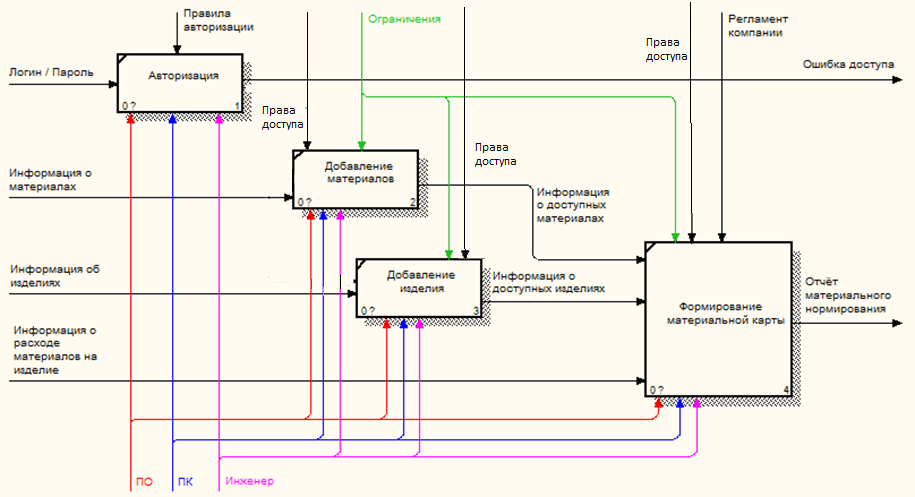


Рис.57 - IDEF0

# ПРИЛОЖЕНИЕ 2 – DFD

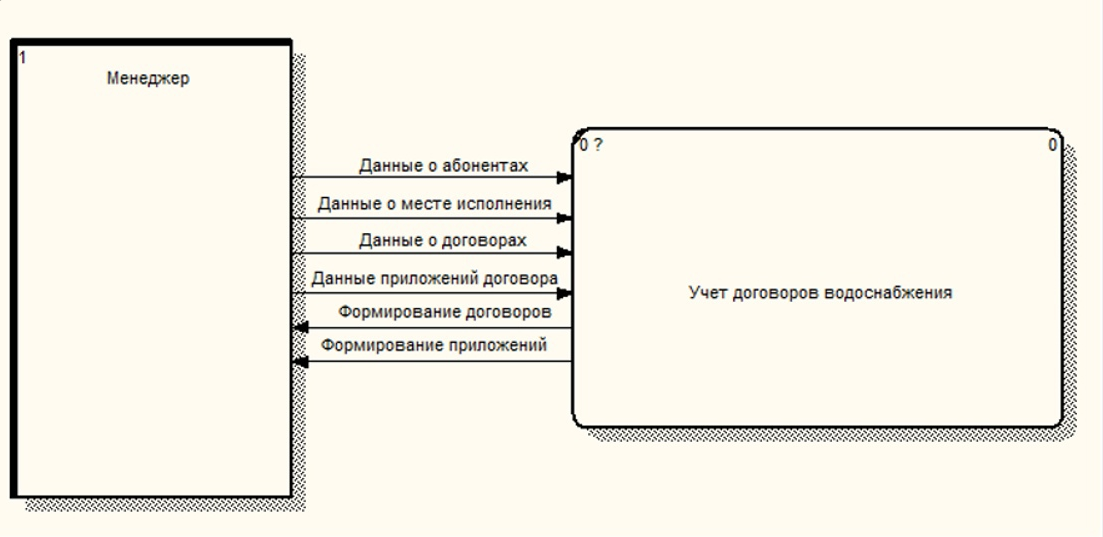


Рис.58 – DFD

ОКОНЧАНИЕ ПРИЛОЖЕНИЕ 2 – DFD

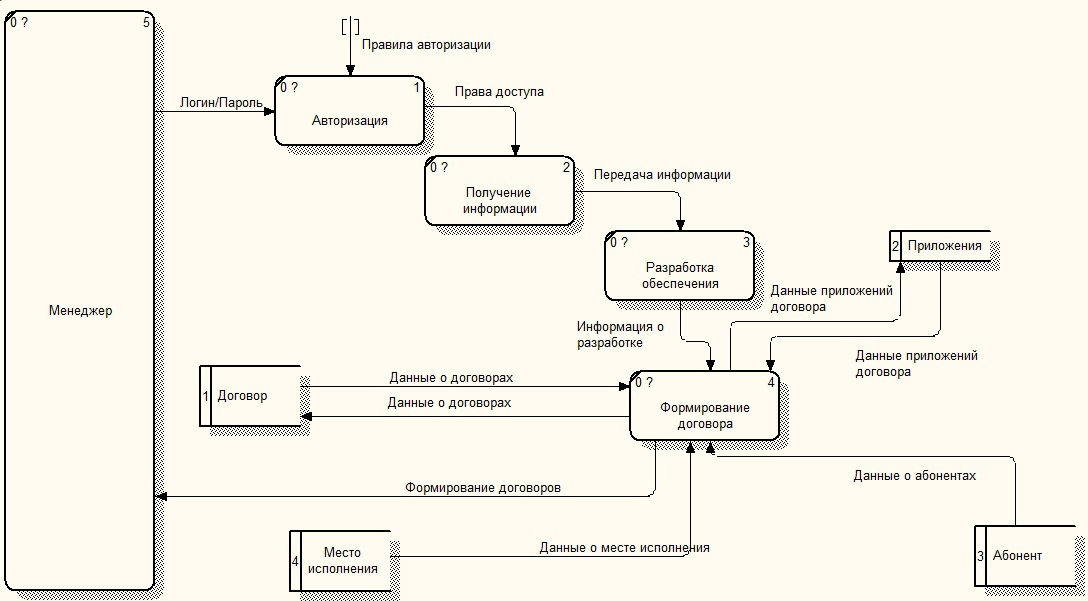


Рис. 59- DFD

# ПРИЛОЖЕНИЕ 3 – Диаграмма классов

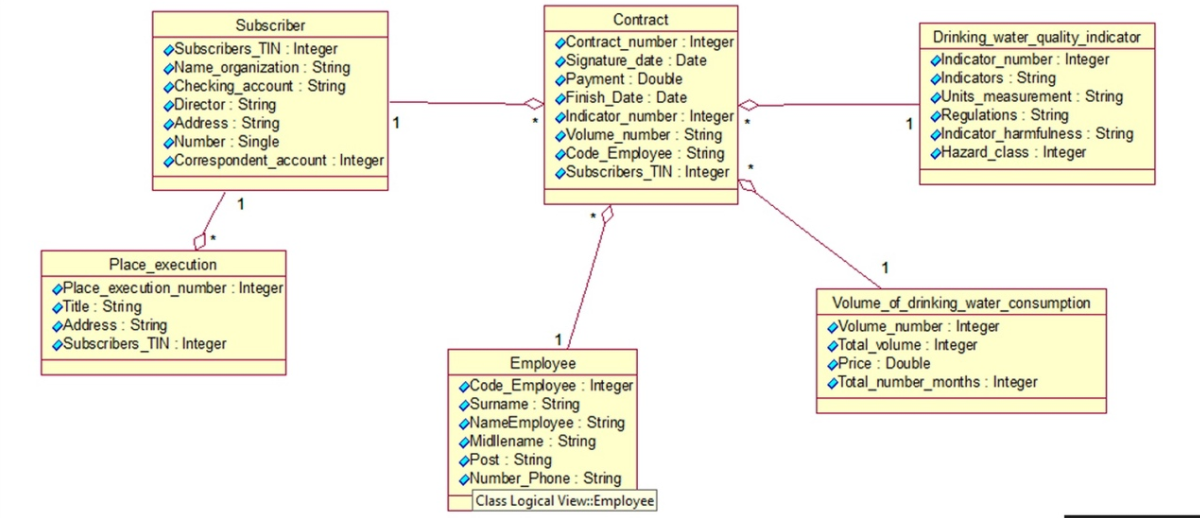


Рис. 60 – Диаграмма классов

# ПРИЛОЖЕНИЕ 4 – ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ

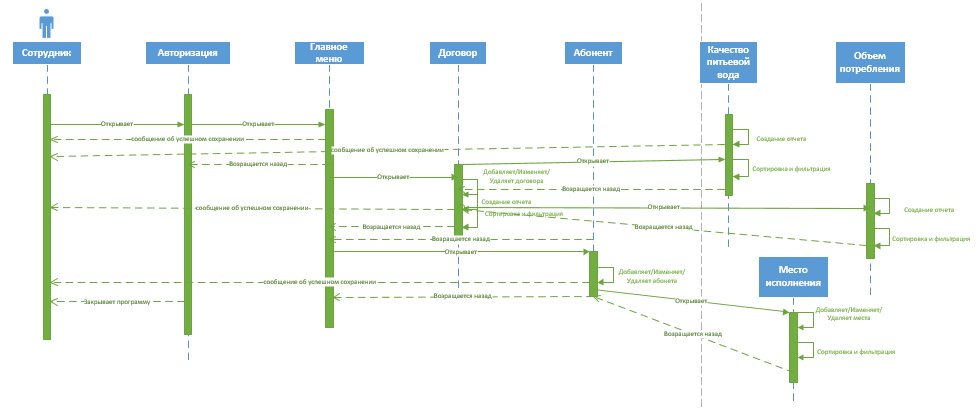


Рис. 61 – Диаграмма последовательности

# ПРИЛОЖЕНИЕ 5 – ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

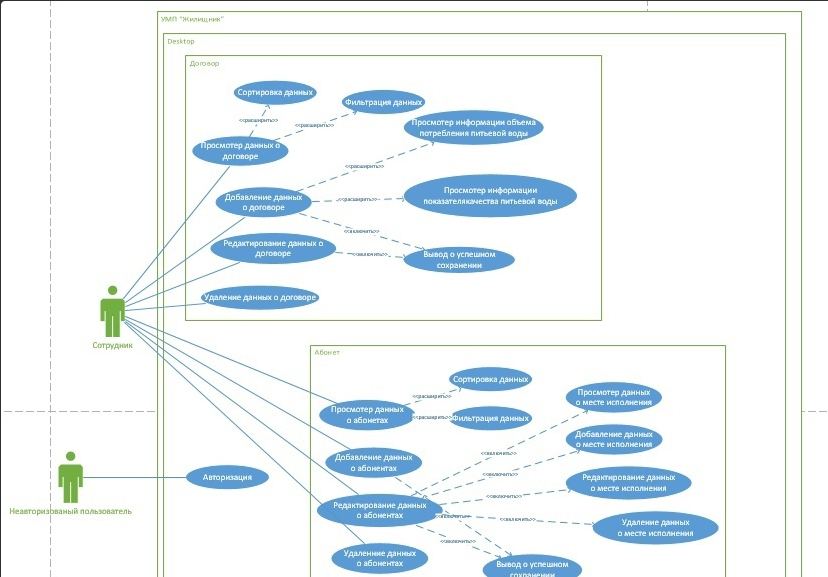


Рис.62 – Диаграмма вариантов использования.

# ПРИЛОЖЕНИЕ 6 – ДИАГРАММА ДЕЯТЕЛЬНОСТИ

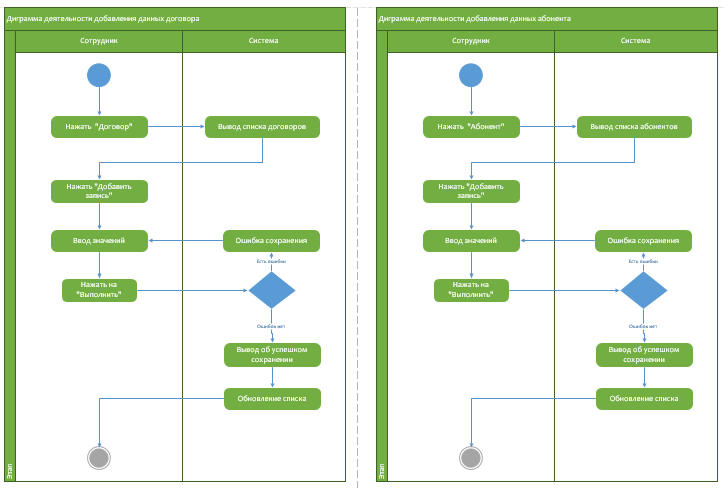


Рис. 63 – Диаграмма деятельности

ПРОДОЛЖЕНИЕ ДИАГРАММЫ 6

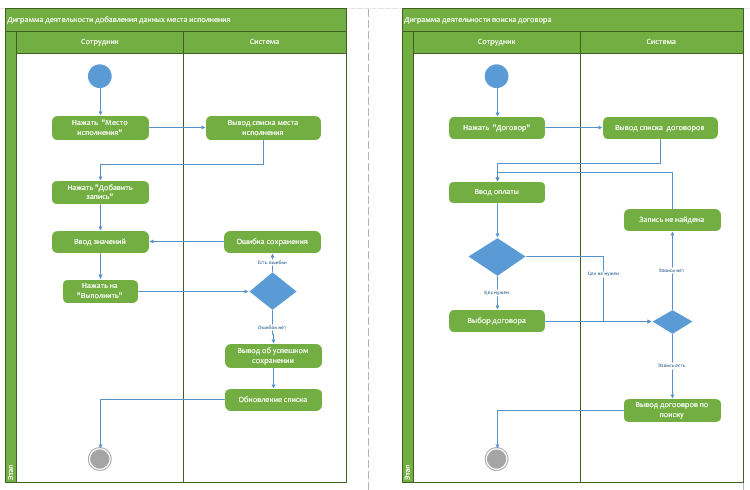


Рис. 64 – Диаграмма деятельности

ПРОДОЛЖЕНИЕ ДИАГРАММЫ 6

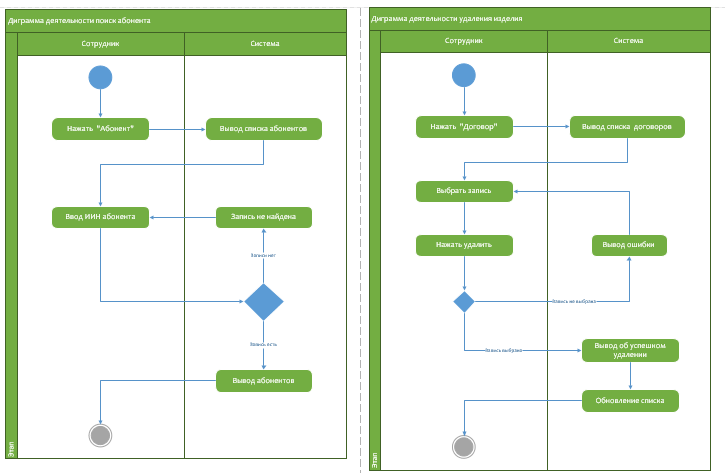


Рис. 65 – Диаграмма деятельности

ОКОНЧАНИЕ ДИАГРАММЫ 6

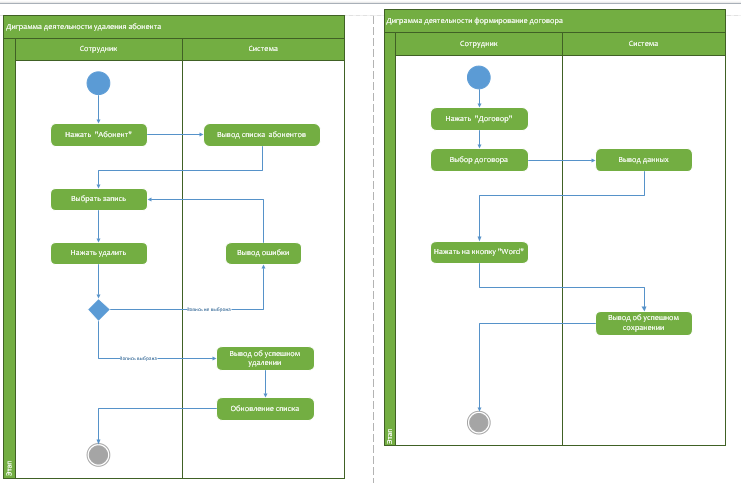


Рис. 66 – Диаграмма деятельности

# ПРИЛОЖЕНИЕ 7 – ДИАГРАММА РАЗВЕРТЫВАНИЯ

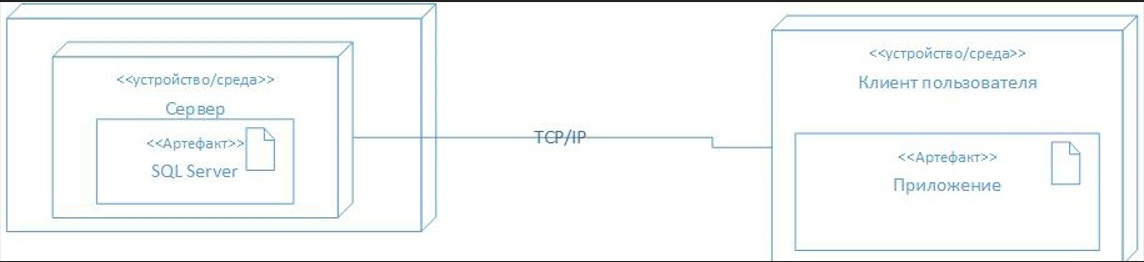


Рис.67 – Диаграмма развертывания.

# ПРИЛОЖЕНИЕ 7 – ЛОГИЧЕСКАЯ МОДЕЛЬ

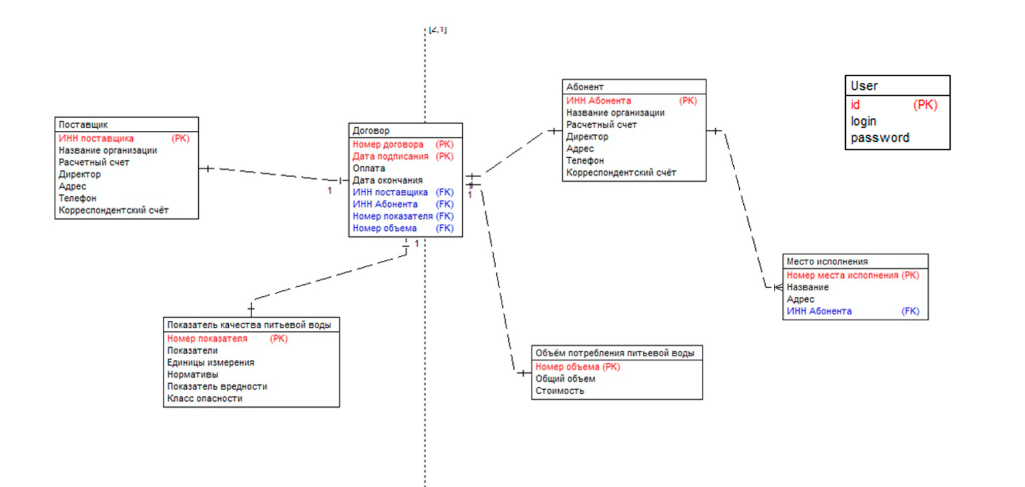


Рис. 68 – Логическая модель.

# ПРИЛОЖЕНИЕ 8 – ФИЗИЧЕСКАЯ МОДЕЛЬ

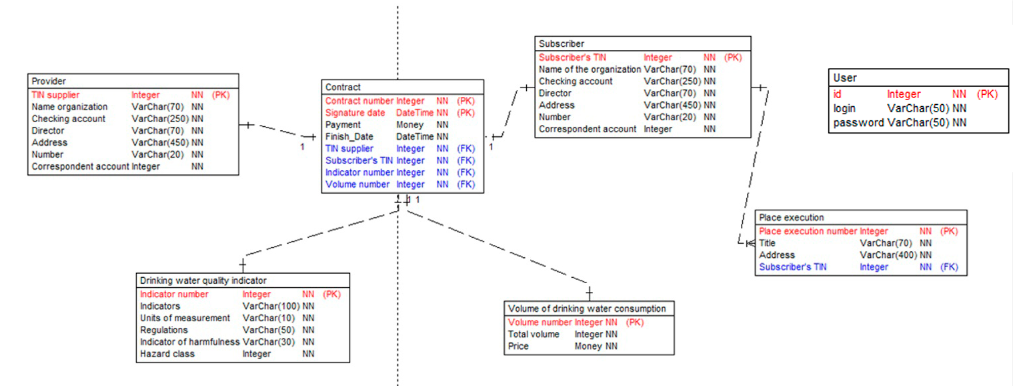


Рис. 69 – Физическая модель

# ПРИЛОЖЕНИЕ 9 – ВХОДНЫЕ ДАННЫЕ

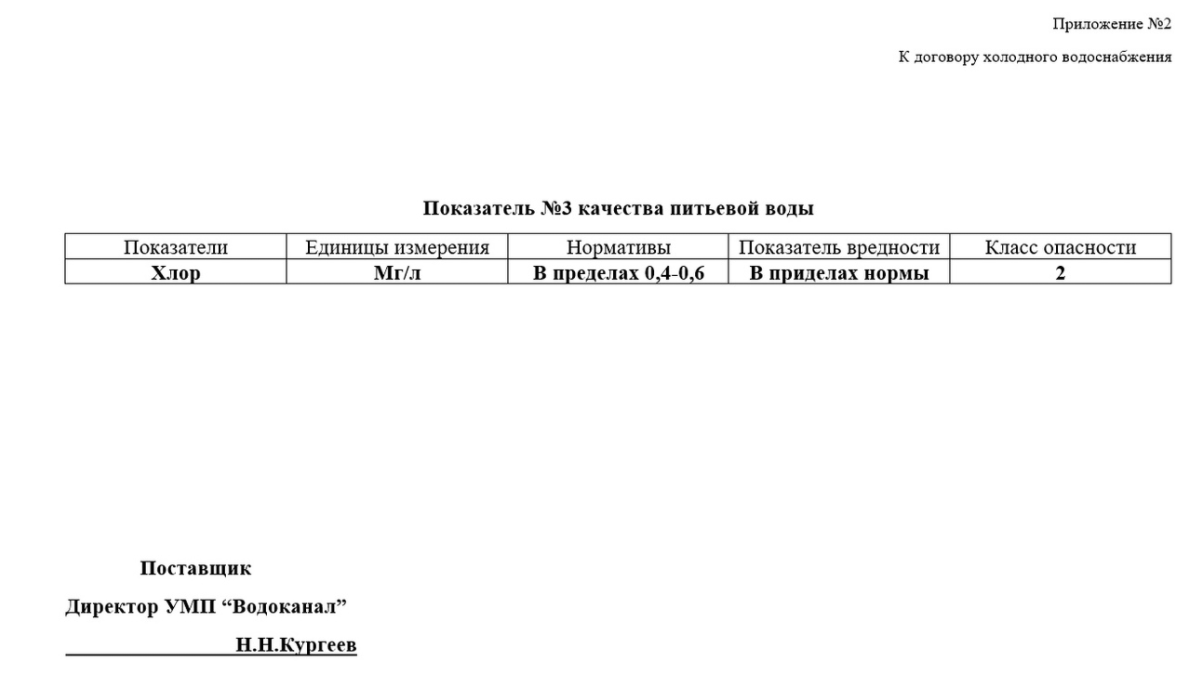
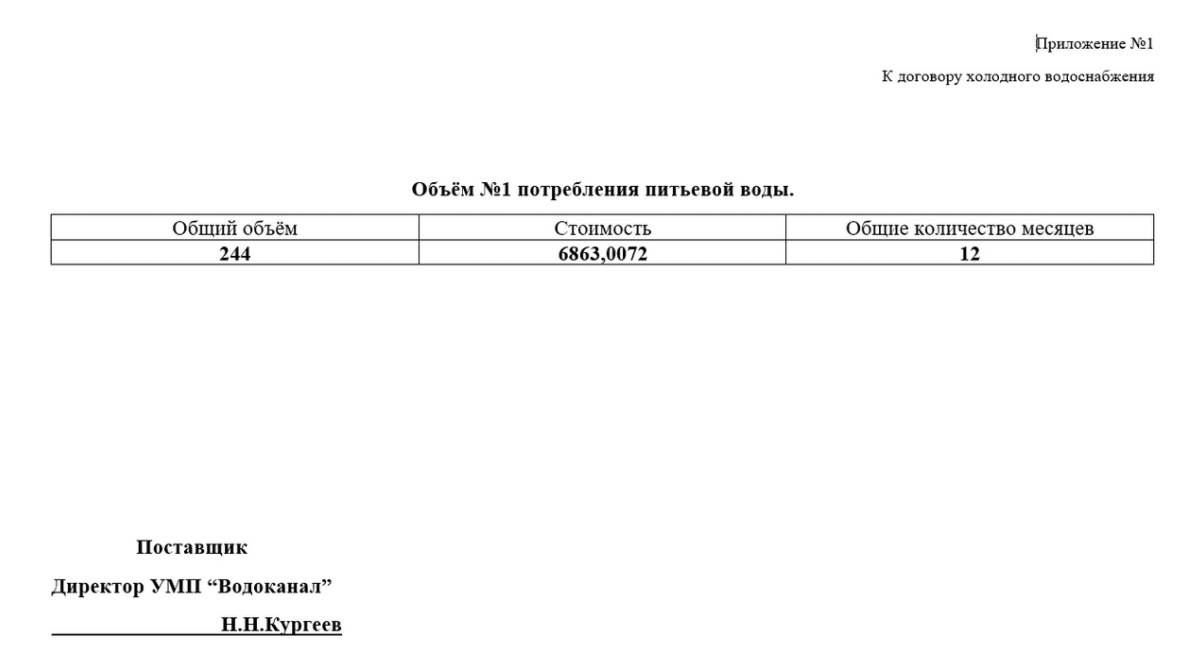
 

Рис. 70 – Входные данные

# ПРИЛОЖЕНИЕ 10 – ВЫХОДНЫЕ ДАННЫЕ

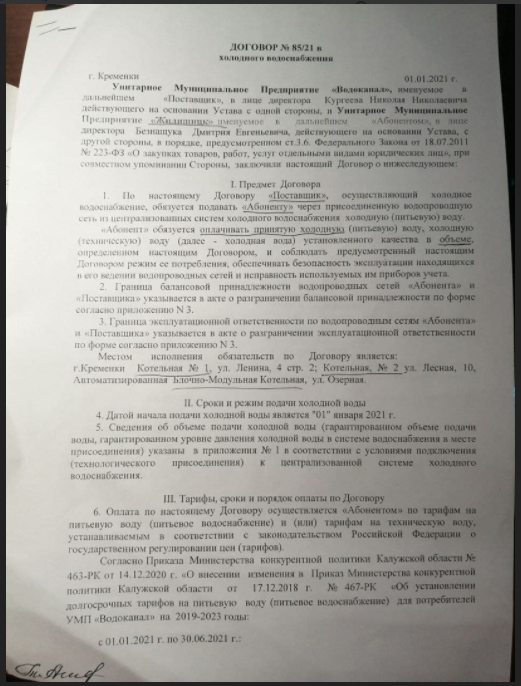
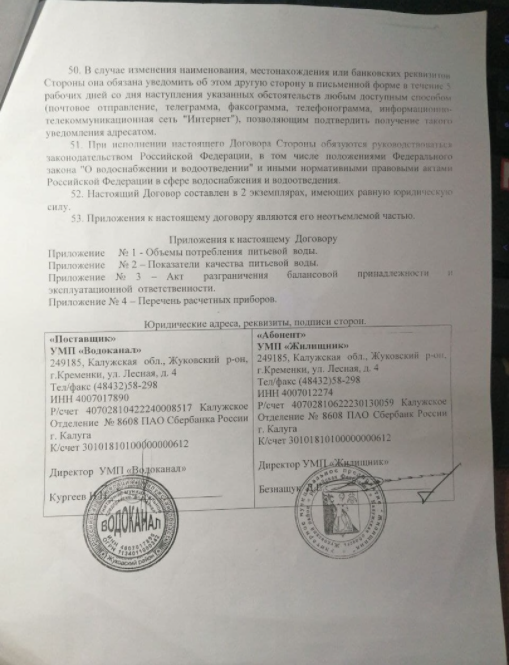
 

Рис. 71 – Выходные данные