

ДИПЛОМНЫЙ ПРОЕКТ

по специальности 09.02.07 Информационные системы и программирование
код и наименование специальности

по теме: Разработка программного продукта для автоматизации
документооборота ООО «НефтеГазоТехнологии»

Разработал _____ группы _____
студент _____

_____ *подпись* _____ *И.О. Фамилия*
_____ 20 _____

Руководитель дипломного проекта _____
_____ *подпись* _____ *И.О. Фамилия*
_____ 20 _____

Задание принял к исполнению _____
_____ *подпись* _____ *И.О. Фамилия*
_____ 20 _____

Нормоконтролер _____
_____ *подпись* _____ *И.О. Фамилия*
_____ 20 _____

Рецензент _____
_____ *подпись* _____ *И.О. Фамилия*
_____ 20 _____

Самара, 2022 г.

toptem.ru
8 (800) 100-67-87
info@toptem.ru

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1 ОБЩАЯ ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	7
1.1 Анализ объекта, процесса, предметной области как объекта информатизации	7
1.2 Анализ информационного обеспечения объекта информатизации	8
1.3 Организационная и функциональная схемы объектов информатизации	9
1.4 Описание категорий пользователей и потоков данных объекта информатизации	12
ГЛАВА 2 ПРЕДПРОЕКТНАЯ ЧАСТЬ	13
2.1 Анализ функциональных требований программного продукта	13
2.2 Анализ входных и выходных данных, определение структуры данных	17
2.3 Разработка тестовых сценариев, программы и методики испытаний	18
2.4 Разработка технического задания программного продукта	23
2.5 Разработка (оптимизация) схемы данных программного продукта	29
2.6 Описание клиент-серверного взаимодействия	30
2.7.Выбор программного обеспечения для реализации (разработки) программного продукта	31
2.8 Выбор программного обеспечения для разработки и эксплуатации удаленных базы данных	32
ГЛАВА 3 ПРОЕКТНАЯ ЧАСТЬ	33
3.1 Разработка интерфейса программного продукта	33
3.2 Разработка удаленной базы данных и необходимых запросов	37
3.3 Реализация функциональных требований	40
3.4.Интеграция программных модулей	47
3.5 Отладка программного продукта	48
3.6 Тестирование программного продукта	51
ГЛАВА 4 ДОКУМЕНТАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММНОГО ПРОДУКТА	52

4.1 Документированный программный код	52
4.2 Руководство пользователя программного продукта	54
4.3 Руководство администратора (включая инструкцию по установке)	63
ГЛАВА 5 ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ПРОГРАММНОГО ПРОДУКТА	65
5.1 Разработка мер защиты информации от несанкционированного доступа	65
5.2 Мероприятия по обеспечению техники безопасности и пожарной безопасности	66
ЗАКЛЮЧЕНИЕ	70
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	72
ПРИЛОЖЕНИЕ А UML-диаграммы	77
ПРИЛОЖЕНИЕ Б Руководство пользователя (на английском языке)	80
ПРИЛОЖЕНИЕ В Руководство администратора (на английском языке)	89
ПРИЛОЖЕНИЕ Г Документированный код программного продукта	91
ПРИЛОЖЕНИЕ Д Флешка с материалами	131

8 (800) 100-67-87

info@toptem.ru

ВВЕДЕНИЕ

Вопрос о необходимости автоматизации документооборота давно перешёл в практическую плоскость, и все больше российских предприятий внедряют у себя системы электронного документооборота, позволяя организациям уже на собственном опыте оценить преимущества новой технологии работы с документами.

Сотрудники могут оперативно получать любую необходимую информацию по клиентам компании. В условиях современного развития бизнеса автоматизация документооборота должна решать задачи, связанные с оптимизацией затрат, с возможностью экономии внутренних ресурсов предприятия. Деятельность любого современного предприятия сопровождается накоплением огромных объёмов данных, которые часто имеют разнородную структуру и не поддаются простой агрегации.

В результате сотрудникам трудно делать значимые выводы на основе имеющейся информации, что неизбежно усложняет процесс принятия управленческих решений.

Актуальность данной темы заключается в перемещении большого количества документов и ведение журналов, что ведёт к большим объёмам непроизводительных трудозатрат, следовательно, автоматизация документооборота устранил проблемы, сэкономит время на формирование документации, также позволит максимально быстро и точно выполнить все эти операции и повысить эффективность работы функциональных специалистов, создающих документы и использующих их в повседневной работе.

Цель дипломного проекта: Разработка программного продукта для автоматизации документооборота для строительной компании «НефтеГазoТехнологии».

Объект исследования: Разработка программного продукта для автоматизации документооборота для строительной компании ООО «НефтеГазоТехнологии».

Предмет исследования: Средства и методы разработки программного продукта для автоматизации документооборота для строительной компании ООО «НефтеГазоТехнологии».

Для достижения поставленной цели необходимо выполнить следующие задачи:

- провести анализ объекта, процесса, предметной области как объекта информатизации;
- провести анализ информационного обеспечения объекта информатизации;
- построить организационные и функциональные схемы объектов информатизации;
- описать категории пользователей и потоков данных объекта информатизации;
- провести анализ функциональных требований программного продукта;
- провести анализ входных и выходных данных;
- построить структуру данных;
- разработать тестовые сценарии, программы и методики испытаний;
- разработать техническое задание для программного продукта;
- разработать и оптимизировать схемы данных для программного продукта;
- выбрать программное обеспечение для реализации (разработки) программного продукта;
- выбрать программное обеспечение для разработки и эксплуатации удаленных базы данных;
- разработать интерфейс на русском и английском языках для программного продукта;

- разработать удаленную базу данных и необходимые запросы;
- реализовать функциональные требования;
- описать интеграцию программных модулей;
- выполнить отладку программного продукта;
- провести тестирование программного продукта;
- выполнить документирование программного кода;
- разработать руководство пользователя для программного продукта;
- разработать меры защиты информации от несанкционированного доступа;
- описать мероприятия по обеспечению техники безопасности и пожарной безопасности.

Дипломный проект основан на гипотезе, согласно которой разработанный программный продукт будет максимально эффективным, если:

- правильно описаны функциональные требования программного продукта;
- правильно подобрано программное обеспечение для разработки программного продукта;
- разработан оптимальный, удобный и понятный интерфейс пользователя;
- реализованы и протестированы все функции программного продукта;
- техническая документация к программе будет разработана в соответствии с ГОСТ;
- уменьшится вероятность потери информации;
- уменьшится нагрузка на сотрудников;
- будут соблюдены все нормы и правила безопасности при работе.

ГЛАВА 1 ОБЩАЯ ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Анализ объекта, процесса, предметной области как объекта информатизации

Процесс анализа предметной области начинается с определения концептуальных требований ряда пользователя.

Концептуальное проектирование включает описания объектов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области и выявляемых в результате анализа данных, который может проводиться с помощью традиционных методик – анкетирования и интервьюирования. На этом этапе выявляются заинтересованные лица (все те, кто имеет прямое или косвенное отношение к процессу, автоматизация которого производится), их потребности и соответственно функции программного продукта.

Компания ООО «НефтеГазоТехнологии» работает в сфере строительства. Основным направлением деятельности является строительство стратегически важных объектов повышенной опасности в нефтяной и газовой промышленности. Она предоставляет услуги в области добычи нефти и природного газа, производство земляных работ, работы строительные отделочные, разведочное бурение, подготовка строительной площадки, производство электромонтажных, санитарно-технических и прочих строительного-монтажных работ.

Одним из структурных подразделений ООО «НефтеГазоТехнологии» является Отдел материально технического снабжения (ОМТС). В состав управления входят: начальник отдела, инженер по электрооборудованию, инженер по общестроительным материалам, инженер по заказной продукции.

Отдел материально технического снабжения занимается нахождением: поставщиков, стройматериалов для проектов, заводов. Заключают договоры, контролируют поставку материалов, нанимают транспорт для перевозки.

Отдел имеет в своем распоряжении множество документов, которые хранят в себе информацию о поставках, договорах, ГОСТах, сертификатов материалов.

Преимущества:

- 1) Осуществляют собственными силами предподготовку к проведению работ.
- 2) Компания отвечает за качество выполняемых работ и несет гарантийные обязательства перед своим заказчиком. Общество имеет все необходимые лицензии для выполнения строительно-монтажных работ на особо опасных и технически сложных объектах России.
- 3) Долгое время работ на рынке (9 лет).

1.2 Анализ информационного обеспечения объекта информатизации

8 (800) 100-67-87

Сотрудники ООО «ИТ» используют такие программные продукты, как:

- Операционная система Windows 10;
- Microsoft Office 2019;
- Битрикс-24.

А также можно описать аппаратное обеспечение предприятия общими словами, как:

- Видеокарта NVIDIA GeForce GTX 1050 Ti;
- Жесткий диск Samsung V-NAND SSD 970 EVO Plus;
- Оперативная память CORSAIR Vengeance LPX 16Гб (2x8 Гб) DDR4;
- Блок питания Aerocool KCAS RGB 850W [KCAS-850G];
- Материнская плата PRIME Z370-P;

- Процессор Intel® Core™ i5-9400F CPU @ 2.90GHz;
- Кулер Deepcool GAMMAXX 300.

1.3 Организационная и функциональная схемы объектов информатизации

Организационная схема показывает взаимодействие основных отделов и служб организации, представлена на рисунке 1.

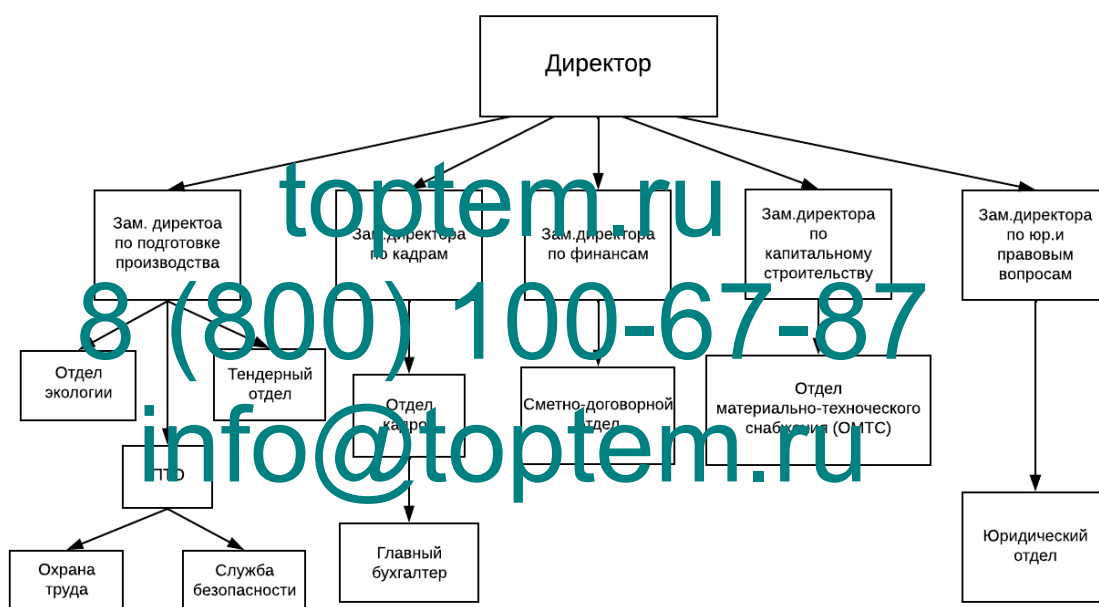


Рисунок 1 – Организационная схема

Для получения информации об объекте автоматизации, поиска процессов, которые требуют автоматизации, определения потребностей непосредственно заказчика и пользователей будущей информационной системы были использованы различные способы:

- беседа с сотрудниками объекта информатизации (интервью);
- анкетирование сотрудников объекта информатизации;
- наблюдение за выполнением рабочих процессов сотрудниками.

В результате обработки результатов, полученных в ходе анализа, получены следующие данные:

- функциональные обязанности сотрудников структурного подразделения ОМТС;
- список заинтересованных в информационной системе лиц;
- список потребностей заинтересованных лиц в разрабатываемой информационной системе;
- описание объекта информатизации и рабочих процессов;
- модель объекта информатизации.

При изучении функциональных обязанностей сотрудника данного структурного подразделения было выявлено строгое подчинение Уставу, должностным инструкциям, положением о структурном подразделении, в котором работает сотрудник, положениями в компании и различными документированными процедурами.

Функциональные обязанности инженера:

- Занимается поиском поставщиков;
- Ведет коммерческие переговоры с поставщиками;
- Осуществляет заключение договоров на поставку материалов;
- Отслеживают поставку товара;
- Оформляют договор о покупке товаров;

Было проведено анкетирование инженера по продажам компании, на основании которого сделан вывод, что в первую очередь сотрудник заинтересован в автоматизации документооборота. Во вторую – автоматизация процесса получения некоторой отчетной документации, даже не по форме установленного шаблона, а как сам факт, с целью дальнейшей обработки информации.

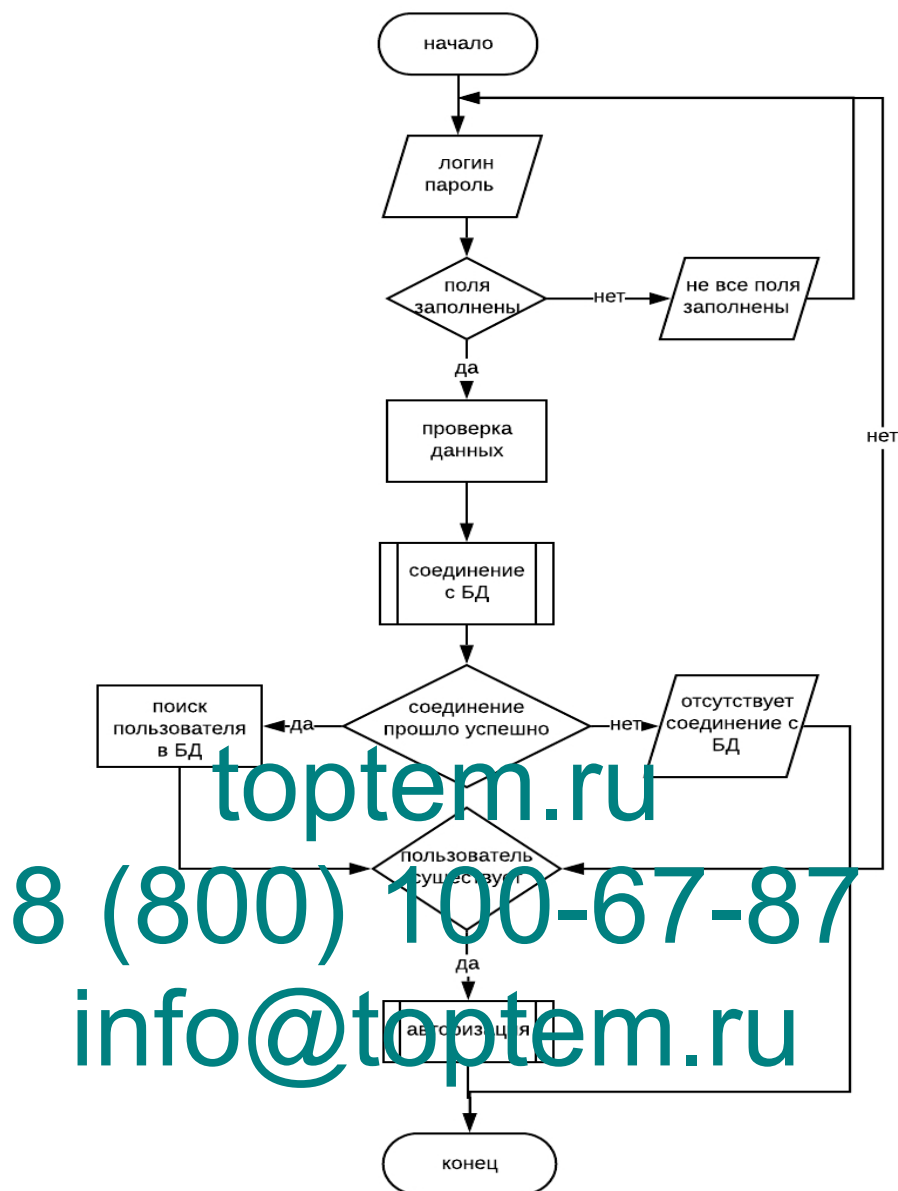


Рисунок 2 – Функциональная схема

В самом начале пользователь должен авторизоваться в программе, если он вводит неверные данные, то появляется ошибка и пользователю нужно повторить авторизацию. Если пользователь всё же прошёл авторизацию, то у него открывается главная форма программы.

1.4 Описание категорий пользователей и потоков данных объекта информатизации

На основе проведенного интервью были выявлены следующие категории пользователей и потоков данных объекта информатизации, расположенных в таблице 1.

Таблица 1

Категории пользователей

Категория пользователей	Функциональные возможности
Администратор	Управление списком договоров Управление списком материалов Управление списком строительной документации Управление списком строительных объектов Управление списком поставщиков Добавление информации о пользователях Настройка подключения базы данных Управление и изменение информации Заполнение документов на основе шаблона Экспортирование информации Импортирование информации
Инженер	Управление списком договоров Управление списком материалов Управление списком поставщиков Добавление информации о договорах, материалах и поставщиках Изменение информации о договорах, материалах и поставщиках Заполнение документов на основе шаблона Экспортирование информации Импортирование информации

ГЛАВА 2 ПРЕДПРОЕКТНАЯ ЧАСТЬ

2.1 Анализ функциональных требований программного продукта

Для определения функциональных требований к программному продукту необходимо провести интервью с сотрудником, непосредственно заинтересованным в автоматизации данного процесса.

В чем специфика интервьюирования? **Интервьюирование** – это разновидность опросного метода. При интервьюировании бланк интервью заполняет не сам респондент, а интервьюер, само же интервью напоминает беседу. Под **интервью** понимается непосредственное устное общение с респондентом. Специфика интервью заключается в следующем. Во-первых, высока роль интервьюера: человека, проводящего беседу, что повышает требования к его подготовке, а во-вторых, высока степень контроля над условиями получения информации, в которых находится респондент [43].

Анкета представляет собой специально разработанный опросный лист для получения определенных сведений о том, кто его заполняет. Анкеты позволяют составить грубое представление о деятельности предприятия, что позволит спланировать первоначальное распределение работ группы аналитиков. Шаблон анкеты для интервьюирования представлен в таблице 2.

Таблица 2

Карта тем для интервью

№п/п	Тема	Вопросы	Примечание
1	Информация о проекте	1) Какова основная предметная область? 2) Каковы основные функциональные обязанности сотрудника? 3) Какие основные проблемы существуют? 4) Как решаются данные проблемы в настоящее	Оценить отношение к проекту

		<p>время?</p> <p>5) Как хотелось бы решать их в дальнейшем?</p>	
2	Понимание пользовательской среды	<p>1) Какие пользователи системы планируются?</p> <p>2) Какое образование и навыки работы с компьютером?</p> <p>3) Имеют ли пользователи опыт работы с таким типом приложений?</p> <p>4) Какие платформы используются?</p> <p>5) Каковы планы на будущее относительно платформ?</p> <p>6) Используется ли ПО для решения проблемы на текущий момент? Каков его функционал?</p> <p>7) Необходимо ли обучение пользователей?</p> <p>8) Необходимы ли справочные руководства, и в каком виде?</p>	Степень заинтересованности, вовлеченности
3	Описание бизнес-процесса	<p>1) Опишите процесс внесения данных о материалах</p> <p>2) Какие данные вносятся?</p> <p>3) Каким образом следует осуществлять их хранение? Необходимо ли шифрование?</p>	
4	Проблемное поле	<p>1) Часто ли бывают ошибки в данных? Если да, то какие.</p> <p>2) Сколько времени уходит на обработку данных одной партии стройматериалов?</p>	
5	Подведение итогов	<p>1) Можно ли будет обратиться к Вам в будущем для получения дополнительной информации?</p>	

Результат интервью приведен в таблице 3.

Таблица 3

Ответы на интервью

№ п/п	Вопрос	Ответ	Примечание
1.	Какова основная предметная область?	Отдел материально технического снабжения, по работе с оформлением документации на материалы	Посмотреть работу ОМТС для полного понимания алгоритмов
2.	Каковы основные функциональные обязанности сотрудника?	Основные функциональные обязанности изложены в документированной процедуре.	Посмотреть документированную процедуру
3.	Как решаются данные проблемы в настоящее время?	В настоящее время используется ПО «Битрикс-24»	Посмотреть функционал программы
4.	Как хотелось бы решать их в дальнейшем?	Хотелось бы более удобную программы для работы	
5.	Какие пользователи системы планируются?	Планируется два пользователя системы Инженер и Администратор	
6.	Какое образование и навыки работы с компьютером?	У сотрудников хороший опыт работы с компьютером	
7.	Имеют ли пользователи опыт работы с таким типом приложений?	Да, похожие приложения есть.	
8.	Какие платформы используются?	Используется ОС Windows, офисный пакет	Посмотреть компьютер сотрудника
9.	Каковы планы на будущее относительно платформ?	Изменяться не будет	
10.	Используется ли ПО для решения проблемы на текущий момент? Каков его функционал?	Используется ПО «Битрикс-24», но хотелось бы более удобную программу для работы	Посмотреть функционал программы
11.	Необходимо ли обучение пользователей?	Не помешает	
12.	Необходимы ли справочные руководства, и в каком виде?	В ходе ознакомления, возможно, будут нужны	
13.	Опишите процесс внесения данных о документах	Создается запись о документе, где заполняются нужные данные	
14.	Какие данные вносятся?	Наименование, номер документа, дата, статус документа	

15.	Каким образом следует осуществлять их хранение? Необходимо ли шифрование?	Централизованное хранение, чтобы не набирать данные вручную, а переносить их из одной программы в другую	
16.	Сколько времени уходит на обработку данных одной партии стройматериалов?	Около 20 минут, иногда больше	
17.	Можно ли будет обратиться к Вам в будущем для получения дополнительной информации?	Да, конечно	Встретиться еще раз с Лисицким С.В.

В результате анкетирования и интервьюирования всех заинтересованных лиц были сформулированы потребности заказчика относительно разрабатываемого ПП. Далее необходимо провести аналогию между выявленными потребностями, структурой и требованиями ТЗ в соответствии с ГОСТ. Таким образом, потребности заказчика в ТЗ могут быть описаны в разделе «Назначение и цели создания системы» [6].

В процессе анализа зоны проектирования и предметной области были выявлены следующие потребности:

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

- начальное формирование данных о документах в компании в виде списка (ввод с клавиатуры или из файла);
- работа с уже имеющимися данными.

Автоматизация осуществляемых на предприятии процессов позволит хранить информацию в одной базе, информация в которую вводится с помощью удобного интерфейса.

Создание собственной автоматизированной системы позволит учесть все особенности учета на предприятии, то есть разработать только то, что нужно для данного предприятия.

Программный продукт должен выполнять следующие функции/соответствовать следующим функциональным требованиям:

- Разделение доступа пользователей к информации;
- Возможность просмотра и добавления информации, доступной инженерам;
- Облегчение работы с документацией;
- Автоматизация заполнения документов.

2.2 Анализ входных и выходных данных, определение структуры данных

Входными данными разрабатываемой информационной системы можно считать:

- информация о поставщике: название компании, адрес, телефон, изображение;
- информация с материалами: название, цена, производитель, количество, изображение, описание, история продажи, типы;
- информация о поставке: дата, количество, поставщик, материал, цена;
- информация о договорах: наименование, дата заключения;
- информация об объектах: наименование, изображение;
- информация о строительной документации: наименование, изображение, статус.

Указанную информацию можно назвать основной входной информацией. В процессе работы программы, от пользователя, для работы некоторых функций, могут быть запрошены дополнительные данные: даты, примечания и др.

К выходным данным информационной системы можно отнести

составленные по запросу пользователя отчеты, а также представление имеющейся ранее входной информации в базе данных:

- результаты внесенных изменений;

2.3 Разработка тестовых сценариев, программы и методики испытаний

Цель тестирования

Целью тестирования проекта является проверка всех функциональных возможностей программного продукта и совместимости с различными операционными системами, проведение серии стресс-тестов для выявления уязвимостей и поиска багов решения [7].

План тестирования

Функциональное тестирование:

Цель: Выявление функциональных ошибок, несоответствий ТЗ и ожиданиям пользователя путем имитации стандартных действий пользователя.

Классификация функций:

1. Авторизация:
 - 1.1.Открытие настроек;
 - 1.1.1. Работа с программой;
2. Работа с меню:
 - 2.1.Вывод списка материалов;
 - 2.2.Вывод списка договоров;
 - 2.3.Вывод списка строительных объектов;
 - 2.4.Вывод списка документации на объект;
 - 2.5.Вывод списка поставщиков.
3. Работа со страницей «Договоры»:

- 3.1. Вывод списка договоров;
 - 3.2. Добавление договора;
 - 3.2.1. Возможность выбора поставщика из системы, отображение дополнительных полей ввода.
 - 3.2.2. Возможность заменить скан договора.
 - 3.3. Заполнение договора по шаблону;
 - 3.3.1. Заполнение данных по меткам.
 - 3.3.2. Данные сохранены в формате .docx
 - 3.4. Удаление договора.
 - 3.4.1. Запрос на подтверждение удаления.
 - 3.5. Импортирование списка;
 - 3.5.1. Данные сохранены в формате .xlsx.
 - 3.6. Экспортирование списка
 - 3.6.1. Данные сохранены в формате .xlsx.
4. Работа со страницей «Материалы»
 - 4.1. Вывод списка материалов.
 - 4.2. Добавление материала.
 - 4.3. Изменение материала.
 - 4.4. Возможность просмотра списка материалов по выбранному типу.
 5. Работа со страницей «Строительная документация»
 - 5.1. Вывод списка строительной документации.
 - 5.2. Добавление документа.
 - 5.2.1. Возможность выбора наименования строительного объекта, на который оформлены документы.
 - 5.3. Изменение документа.
 - 5.3.1. Возможность менять наименование строительного объекта.
 - 5.3.2. Возможность замены изображения документа.
 - 5.4. Удаление строительной документации.
 - 5.4.1. Подтверждение удаления пользователем.

6. Работа со страницей «Строительный объект»
 - 6.1. Вывод списка.
 - 6.2. Добавление объекта.
 - 6.2.1. Возможность выбора клиента.
 - 6.2.2. Возможность выбора договора.
 - 6.2.3. Возможность замены изображения объекта.
 - 6.3. Изменение объекта.
 - 6.3.1. Возможность изменения клиента.
 - 6.3.2. Возможность изменение договора.
 - 6.3.3. Возможность изменения изображения объекта.
 - 6.4. Удаление документа.
 - 6.4.1. Подтверждение удаления пользователем

7. Работа со страницей «Поставщики»

- 7.1. Вывод списка поставщиков.

- 7.2. Добавление поставщика.

- 7.2.1. Возможность ввода данных.

- 7.2.2. Возможность загрузки изображения.

- 7.3. Изменение документа

- 7.3.1. Возможность изменения данных.

- 7.3.2. Возможность изменения изображения.

- 7.4. Удаление документа.

Тестирование в определенной среде

Цель: проверить работу всех модулей программного продукта и корректность отображения форм в различных операционных системах и при различных разрешениях монитора.

Стресс-тестирование

Цель: выявить уязвимости программного продукта посредством имитации чрезмерной нагрузки на информационную систему, ввода данных большой длины и некорректных данных.

Этапы:

1. Авторизация пользователя;
2. Добавление нового пользователя;
3. Удаление пользователя;
4. Поиск.

План тестирования для последующего процесса тестирования представлен в таблицах 4, 5, 6.

Таблица 4

Функциональное тестирование

Функция	Действие	Ожидаемый результат	Полученный результат	Фатальные ошибки
Авторизация пользователей	Ввод данных для входа в поля Логин и Пароль, последующее нажатие кнопки «Войти»	Переход на форму главного меню, в соответствии с ролью пользователя		
Добавление данных	Ввод новых данных в соответствующие поля и нажатие кнопки «Сохранить»	Успешное сохранение данных		
Изменение данных.	Ввод новых данных в соответствующие поля и нажатие кнопки «Сохранить»	Успешное сохранение измененных данных.		
Сохранение информации	Отправка запроса к БД	Успешное сохранение		
Использование кнопок главного меню	Нажатие на кнопку	Переход на соответствующую страницу		
Поиск	Ввод данных в поле	Вывод данных согласно поисковому запросу		
Экспорт	Нажатие кнопки «Экспорт» на странице	Данные экспортированы в формате .xlsx		
Импорт	Нажатие кнопки «Импорт» на странице	Данные импортированы в список		

Отмена	Нажатие на кнопку «Отмена», на странице Добавления/ Редактирования	Возвращает на начальную страницу		
Заполнение договоров по шаблону	Заполнение документа данными	Данные сохранены в формате .docx		
Сортировка и фильтрация	Раскрытие списка с данными	Изменения списка согласно выбранным данным		

Таблица 5

Тестирование в определенной среде

Система	Разрешение экрана	Ошибки/несоответствия
Windows 7	1366x768	
Windows 8	1280x1024	
Windows 10	3840x2160	
Windows 10	1920x1080	

Таблица 6

8 (800) 100-67-87

Стресс-тестирование

Функция	Действие	Ожидаемый результат	Полученный результат	Фатальные ошибки
1 Авторизация пользователя	Ввод некорректных данных для входа в поле Логин и Пароль	Сообщение о неверном вводе данных		
2 Добавление данных	Ввод некорректных данных	Сообщение о неверном вводе данных		
3 Поле поиска	Ввод некорректных данных	Сообщение об отсутствии данных		

2.4 Разработка технического задания программного продукта

Введение

Данное техническое задание предназначено на разработку программного продукта для автоматизации документооборота ООО «НефтеГазоТехнологии», предназначенного для использования инженером компании [12].

Наименование

Программный продукт для автоматизации документооборота ООО «НефтеГазоТехнологии».

Краткая характеристика области применения программы

Область применения данной системы – отдел материально-технического снабжения. Данная разработка предназначена для автоматизации документооборота. Программа будет работать с базой данных, считывая оттуда различные данные, необходимые для работы автоматизированной информационной системы.

Основание для разработки

Основанием для проведения разработки является установленная тема ВКР «Разработка программного продукта для автоматизации документооборота ООО «НефтеГазоТехнологии»».

1. Общие положения

1 Назначение документа

В настоящем документе приводится набор требований к ИС, необходимых для ее правильной реализации и работы.

При реализации необходимо выполнить работы в объеме, указанном в настоящем Техническом задании.

Все неоднозначности, выявленные в настоящем Техническом задании после его согласования и подписания, подлежат двухстороннему согласованию между Сторонами.

2 Цели создания

Данный программный продукт создан с целью повышения

эффективности работы с внутренними документами предприятия.

3 Основные функциональные возможности

- Учет и хранение документов;
- Систематизация документов;
- Контроль исполнения документов по срокам;
- Оперативный доступ к документам;
- Заполнение документов по шаблону.

4 Пользователи системы

Основными пользователями системы являются:

Администратор – имеет полный доступ к системе, добавляет и удаляет пользователей, выдает права доступа, возможность подключать новую базу данных.

Инженер – имеет доступ к системе, может добавлять, редактировать, просматривать данные о поставщиках, договорах и материалах.

Требования к составу выполняемых функций
Разрабатываемый программный продукт должен обеспечивать возможность выполнения перечисленных ниже функций:

- Ввод данных;
- Добавление данных;
- Редактирование данных;
- Осуществление поиска информации;
- Удаление данных;
- Заполнение документов по шаблону;
- Экспорт и импорт данных;
- Сортировка и фильтрация.

Требования к организации входных данных

Входные данные программы должны быть организованы в виде вводимого в специальную форму текста или файла, соответствующего

определенному шаблону.

Файлы указанного формата должны размещаться (храниться) на локальных или съемных носителях, отформатированных согласно требованиям операционной системы.

Требования к организации выходных данных

Выходные данные программы должны быть организованы в виде отчетов или таблиц.

Файлы указанного формата должны храниться на локальных или съемных носителях, отформатированных согласно требованиям операционной системы. Отчеты формируются в режиме реального времени и передаются пользователю в формате «.docx», «.xlsx». При желании любой отчет можно сохранить отдельно.

Описание требований к системе

1 Требования к системе в целом

Система включает в себя 2 уровня доступа. Пользователь должен функционировать в рамках своего информационного пространства.

Для получения доступа в пространство персонифицированного доступа, пользователь должен пройти процедуру аутентификации. В случае успешной аутентификации пользователя все его действия в системе будут однозначно ассоциироваться с его учетной записью.

Процедура аутентификации пользователя Системы должна выполняться с помощью логина и пароля пользователя.

2 Требования к информационному обеспечению системы

Состав, структура и способы организации данных в системе должны быть определены на этапе технического проектирования.

Уровень хранения данных в системе должен быть построен на основе современных реляционных или объектно-реляционных СУБД. Для обеспечения целостности данных должны использоваться встроенные механизмы СУБД.

Средства СУБД, а также средства используемых операционных систем должны обеспечивать документирование и протоколирование обрабатываемой в системе информации.

Структура базы данных должна поддерживать кодирование хранимой и обрабатываемой информации в соответствии с общероссийскими классификаторами (там, где они применимы).

Доступ к данным должен быть предоставлен только авторизованным пользователям с учетом их служебных полномочий, а также с учетом категории запрашиваемой информации.

Структура базы данных должна быть организована рациональным способом, исключающим одновременную полную выгрузку информации, содержащейся в базе данных системы.

Технические средства, обеспечивающие хранение информации, должны использовать современные технологии, позволяющие обеспечить повышенную надежность хранения данных и оперативную замену оборудования (распределенная избыточная запись считывание данных; зеркалирование; независимые дисковые массивы; кластеризация).

В состав системы должна входить специализированная подсистема резервного копирования и восстановления данных.

Данные должны быть защищены от разрушений при авариях и сбоях в электропитании системы путем создания резервных копий.

При проектировании и развертывании системы необходимо рассмотреть возможность использования накопленной информации из уже функционирующих информационных систем.

3 Требования к лингвистическому обеспечению системы

Все обозначения, названия элементов управления Системы, тексты должны быть изложены на русском языке без применения терминов, непонятных пользователю.

Разработка системы должна вестись на языке программирования

высокого уровня PHP.

4 Требования к программному обеспечению системы

Системные программные средства свободно распространяемая операционная система Windows.

Программное обеспечение:

- База данных (СУБД) MySQL Workbench;
- ОС Windows 10;
- Apache HTTP Server версии 2.2.16 (или выше);
- PHP версии 5.1 (или выше);
- Система управления контентом с открытым кодом.

5 Требования к техническому обеспечению системы

Система должна функционировать на ПК, удовлетворяющих следующим требованиям:

1. Тип процессора: процессор типа Intel Core i3 и выше (или эквивалент);
2. Базовая тактовая частота процессора: минимум: 3,0 ГГц;
3. Оперативная память: минимум: 4 ГБ;
4. Дисковое пространство: минимум: 500 ГБ;
5. Источник бесперебойного питания - Обеспечивает работоспособность в условиях кратковременного сбоя электроснабжения;
6. Операционная система: Windows 7 и выше;
7. Файловый менеджер (в составе операционной системы или др.);
8. Браузеры, один из предложенных ниже:
 - a. Internet Explorer версии 10 или более поздней;
 - b. Mozilla Firefox версии 5 или более поздней;
 - c. Google Chrome версии 13 или более поздней;
 - d. Opera версии 11 или более поздней;
9. Почтовый клиент (входит в состав операционных систем или др.);
10. Антивирусная программа (клиентская лицензия);
11. Комплект общеупотребимых офисных программ, включающий:

- a. текстовый редактор,
- b. программу разработки презентаций,
- c. электронные таблицы;

12. Программа для просмотра статических изображений;

Необходимые для работы периферийные устройства и их характеристики:

1. Принтер лазерный сетевой - Формат А4 Быстродействие не ниже 25 стр./мин, разрешение не ниже 600×600 dpi;
2. Сканер - Оптическое разрешение не менее 1200×2400 dpi;
3. Цифровая фотокамера;
4. Расходные материалы – бумага А4, картриджи и тонер для лазерного принтера;

Внутренняя сеть и средства коммуникации, должны обладать как минимум следующими характеристиками:

1. скорость передачи данных подключаемого канала к публичным сетям не менее 2 Мб/с;
2. оборудование узла должно оставаться работоспособным при кратковременных отключениях электропитания (на время не менее 15 минут);
3. оборудование узла должно обеспечивать коммутируемое подключение всех устройств со скоростью до 100 Мбит/с.

6 Требования к организационному обеспечению системы

Организационное обеспечение системы должно быть достаточным для эффективного выполнения персоналом возложенных на него обязанностей при осуществлении автоматизированных и связанных с ними неавтоматизированных функций системы.

Заказчиком должны быть определены структура и функция подразделений, участвующих в функционировании системы, должностные лица, ответственные за:

- обработку информации АС;
- администрирование АС;
- обеспечение безопасности информации АС;
- управление работой персонала по обслуживанию АС.

К работе с системой должны допускаться сотрудники, имеющие навыки работы на персональном компьютере, ознакомленные с правилами эксплуатации и прошедшие обучение работе с системой.

2.5 Разработка и оптимизация схемы данных программного продукта

На основе анализа предметной области и определения функционала, строится инфологическая модель, создаются таблицы в базе данных по этой инфологической модели (рисунок 3).

toptem.ru
8 (800) 100-67-87
info@toptem.ru

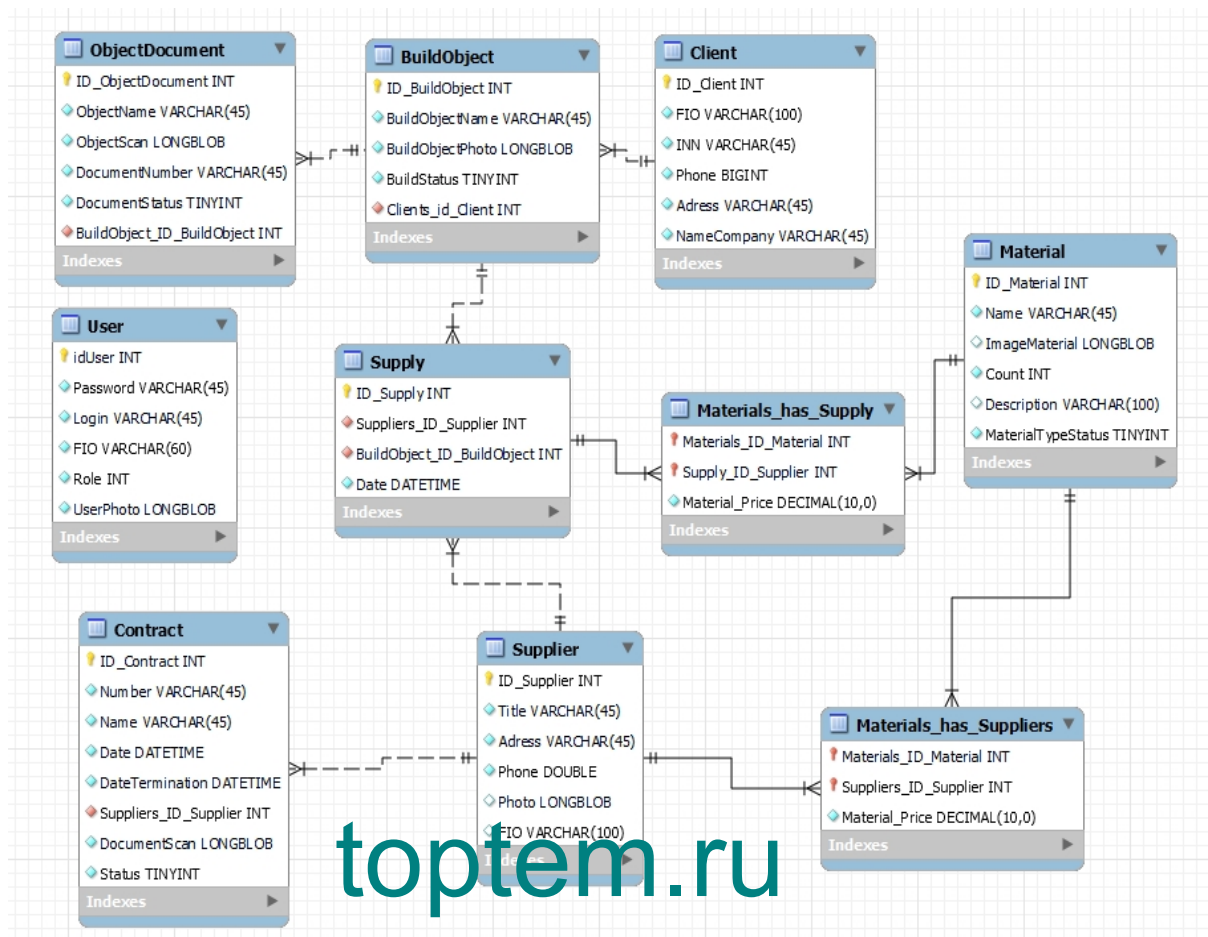


Рисунок 8 – Реляционная схема базы данных (ER-диаграмма)

Основными таблицами являются клиентское поле, которое является первичным ключом (PRIMARY KEY), со свойством AUTO_INCREMENT, необходимые для построения корректных связей между ними. Таблицы, обеспечивающие связь данных, имеют несколько ключевых полей для обеспечения удобства взаимодействия с данными с обеих сторон связей. Все таблицы в базе данных имеют связи для обеспечения целостности базы данных.

2.6 Описание клиент-серверного взаимодействия

Клиент-серверное взаимодействие необходимо в любом современном программном продукте, оно представляет из себя взаимодействие

приложения – клиента и серверного приложения. В данной связке клиент посылает команды удаленному серверу, а удаленный сервер отвечает на них.

В данном программном продукте, в качестве сервера выступает, сервер базы данных MySQL, а в качестве клиента разработанное приложение. Взаимодействие осуществляется модуля Entity Framework, преобразующего высокоуровневые команды LINQ, в низкоуровневые запросы к базе данных [32].

2.7.Выбор программного обеспечения для реализации (разработки) программного продукта

Для выбора интегрированной среды разработки и языка программирования для реализации функционала, сравнивались три системы Visual Studio, VS Code и IntelliJ Idea. Также был проведен сравнительный анализ трех языков программирования, представленный в таблице 7.

8 (800) 100-67-87
info@toptem.ru

Таблица 7

Сравнительный анализ языков программирования

Критерии	C#	C++	Python
Бесплатная IDE	+	+	+
Наличие большого количества библиотек и шаблонов	+	+	+
Подключение к удаленной БД	+	+	+
Кроссплатформенность	+	-	-
Удобство разработки	+	-	+
Быстрая сборка проекта	-	+	+

На основе анализа программный продукт будет реализован на языке C#, так как является актуальным и имеет множество обновляющихся библиотек, в том числе EntityFramework, которая используется для работы с проектом. Для разработки кода выбрана интегрированная среда Visual Studio

[30].

2.8 Выбор программного обеспечения для разработки и эксплуатации удаленных базы данных

При выборе системы управления базами данных (далее - СУБД) для разработки базы данных сравнивались три системы MySQL, SQL Server, Oracle Database. Сравнительный анализ приведен в таблице 8.

Таблица 8

Сравнительный анализ СУБД

Критерии	MySQL	SQL Server	Oracle Database
Шифрование данных	-	-	+
Объектно-ориентированный подход к проектированию базы данных	-	+	+
Наличие схем данных	+	+	+
Хранимые процедуры	+	-	+
Наличие открытых ключей	+	+	-
Кроссплатформенность	+	-	+

Для создания базы данных был выбран сервер MySQL. Это одна из наиболее популярных клиентских программ. Данная СУБД является одной из самых быстрых и высокопроизводительных, её достаточно просто администрировать. Её сервер позволяет подключаться одновременно неограниченному количеству пользователей, что весьма удобно на предприятии. СУБД MySQL имеет довольно маленький размер и имеет хорошую аппаратную совместимость. Благодаря данной СУБД обеспечивается высокий уровень безопасности.

ГЛАВА 3 ПРОЕКТНАЯ ЧАСТЬ

3.1 Разработка интерфейса программного продукта

Программная часть АИС работает на базе WPF – часть экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов. Все страницы программы взаимосвязаны между собой: это позволяет пользователю переходить с одной рабочей страницы на другую, а в самой программе появляется возможность работы с данными, созданными на одной странице, в другой странице. Также одной из важных особенностей является использование языка декларативной разметки интерфейса XAML, основанного на XML: есть возможность создавать насыщенный графический интерфейс [31].

При загрузке программы пользователю открывается окно аутентификации, изображённое на рисунке 4.

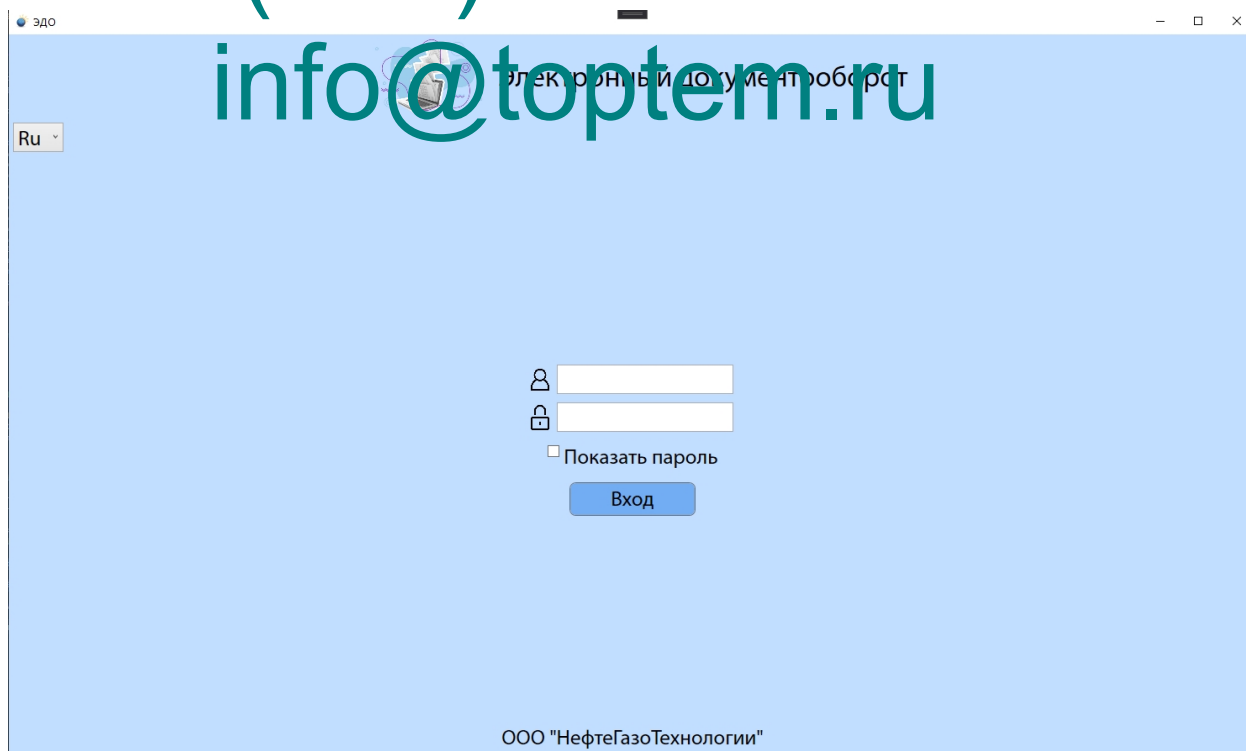


Рисунок 4 – Окно аутентификации пользователя

После аутентификации пользователя с правами администратора открывается главное меню, изображённое на рисунке 5.

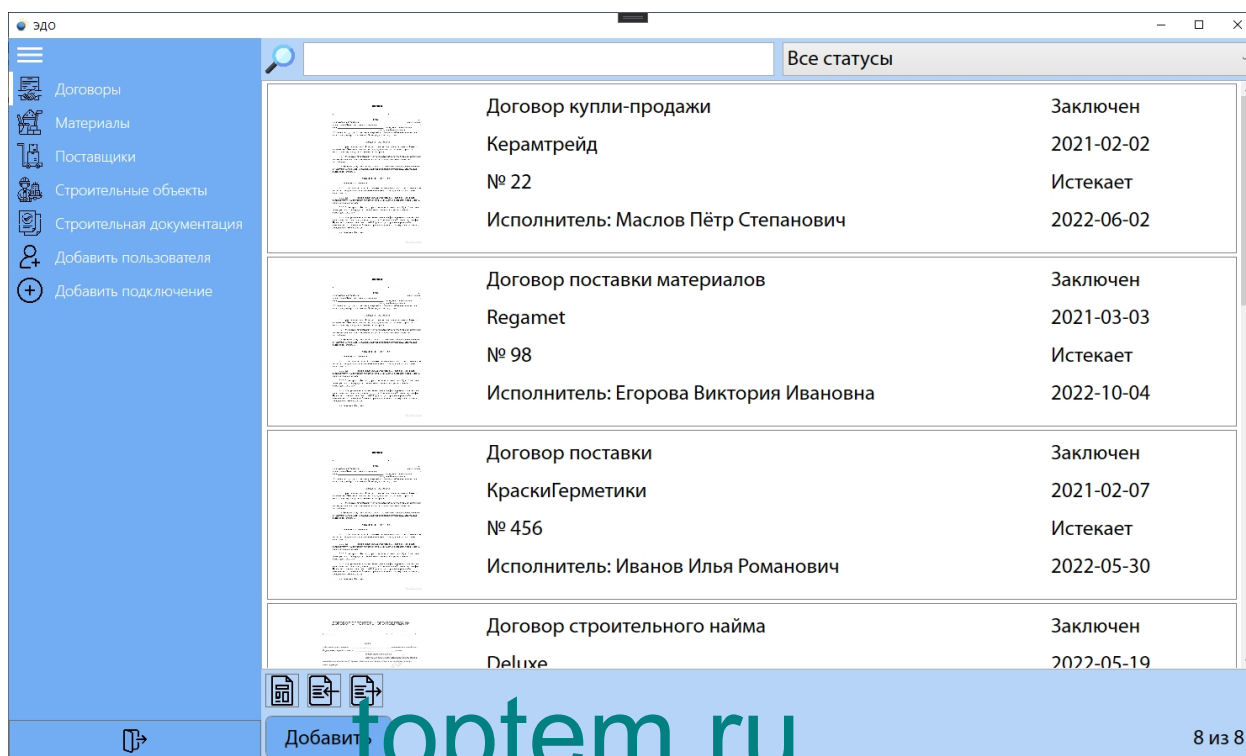


Рисунок 5 – Главное меню программного продукта

8 (800) 100-67-87

Страницы, открывающиеся по нажатию кнопок «Договоры», «Материалы», «Поставщики», «Строительные объекты», «Строительная документация» достаточно схожи между собой: они включают список, изображения, а также кнопки, необходимые для выполнения различных операций. Например, страницу, открывающуюся по нажатию вкладки «Материалы», можно увидеть на рисунке 6, а страницу «Поставщики», соответственно, на рисунке 7.

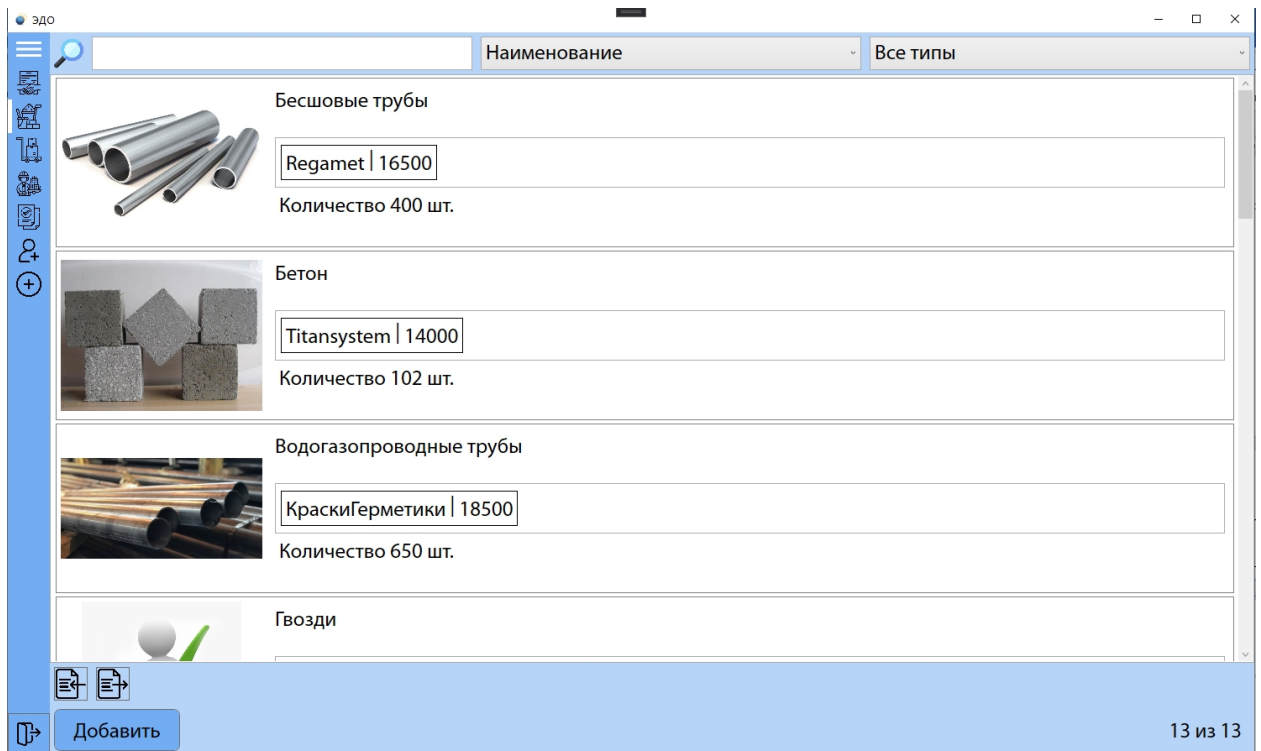


Рисунок 6 – Страница «Материалы»

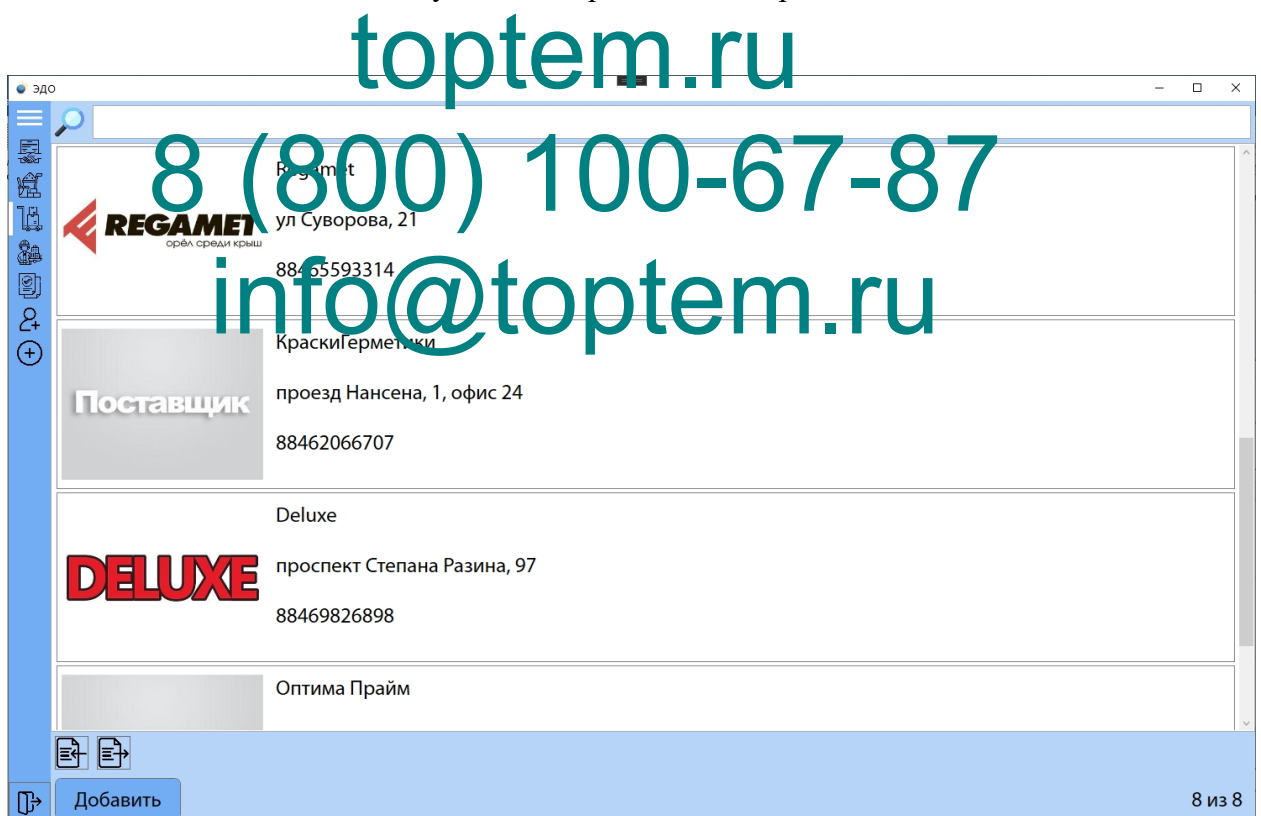


Рисунок 7 – Страница «Поставщики»

Для просмотра информации о строительных объектах используется страница, изображённая на рисунке 8.

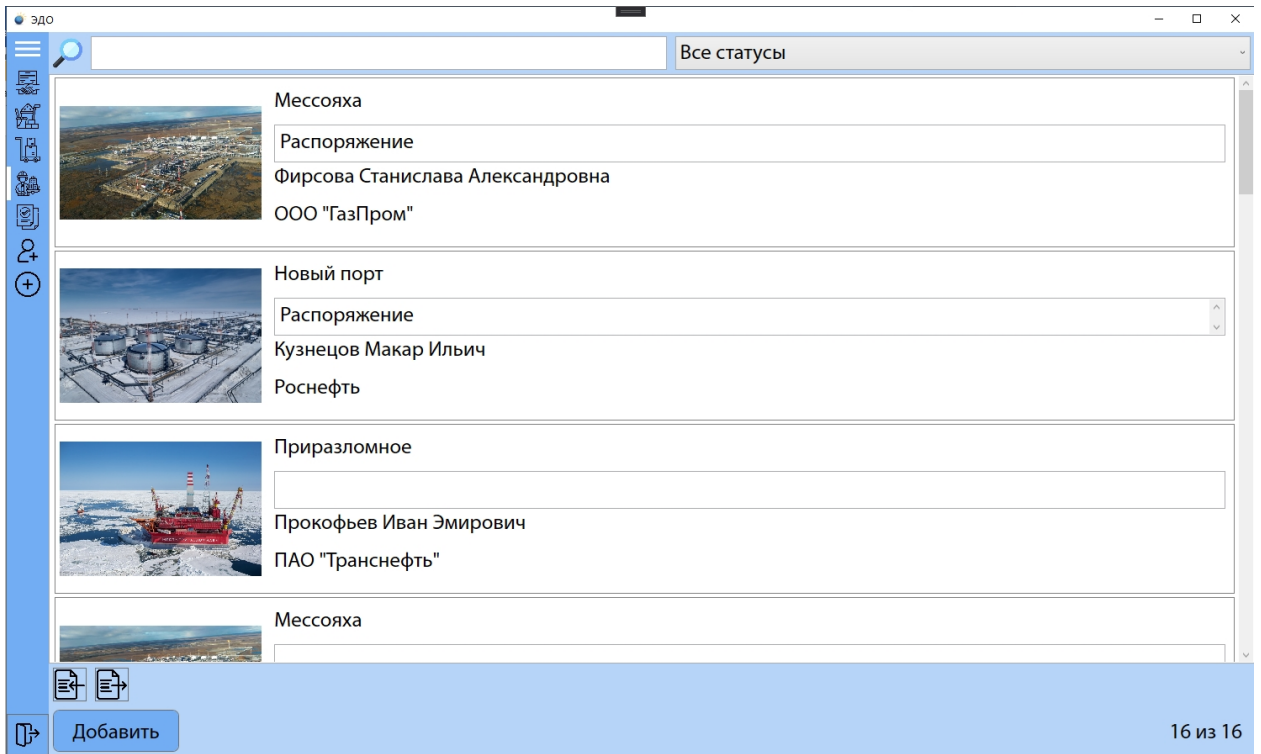


Рисунок 8 – Страница «Строительные объекты»

Для просмотра информации в таблице «Строительная документация»
используется страница, изображенная на рисунке 9.

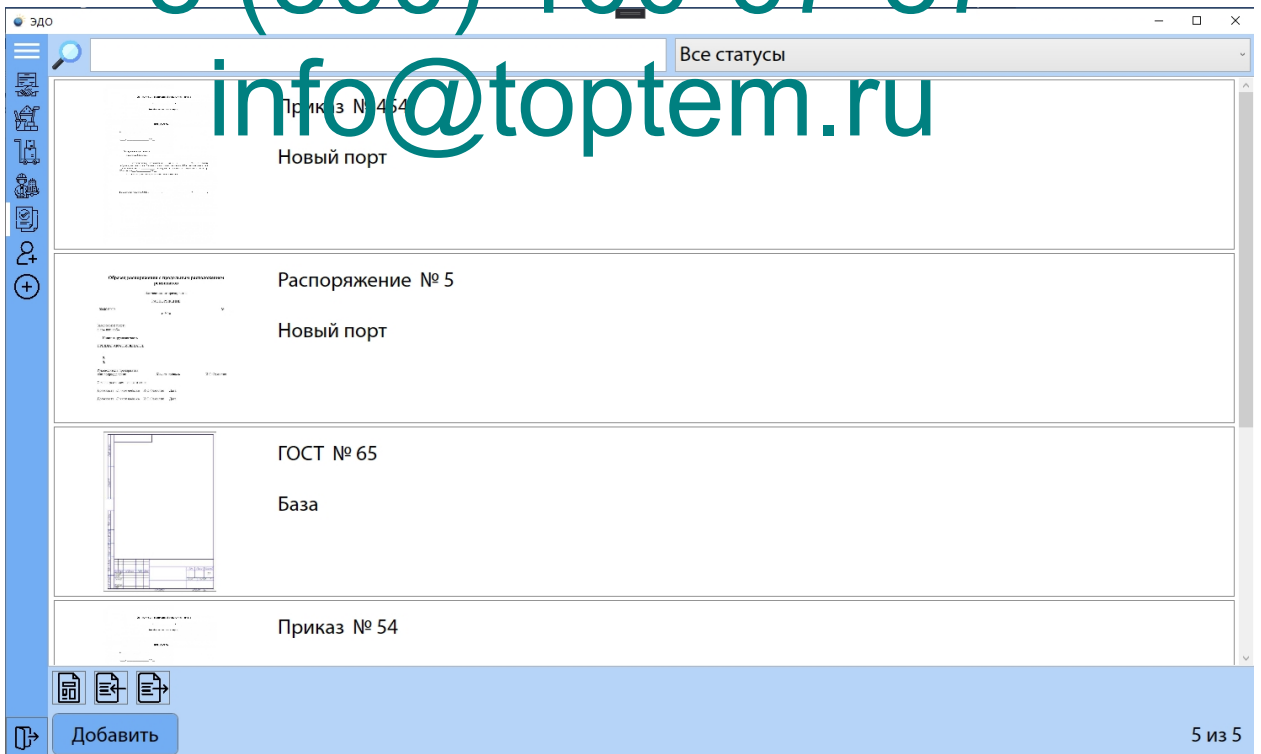


Рисунок 9 – Страница «Строительная документация»

3.2 Разработка удаленной базы данных и необходимых запросов

Для реализации функций программного продукта, было необходимо создать и подключить базу данных MySQL [32]. В появившейся странице, в левом меню нажав правую кнопку мыши «Create Schema», введите имя создаваемой базы.

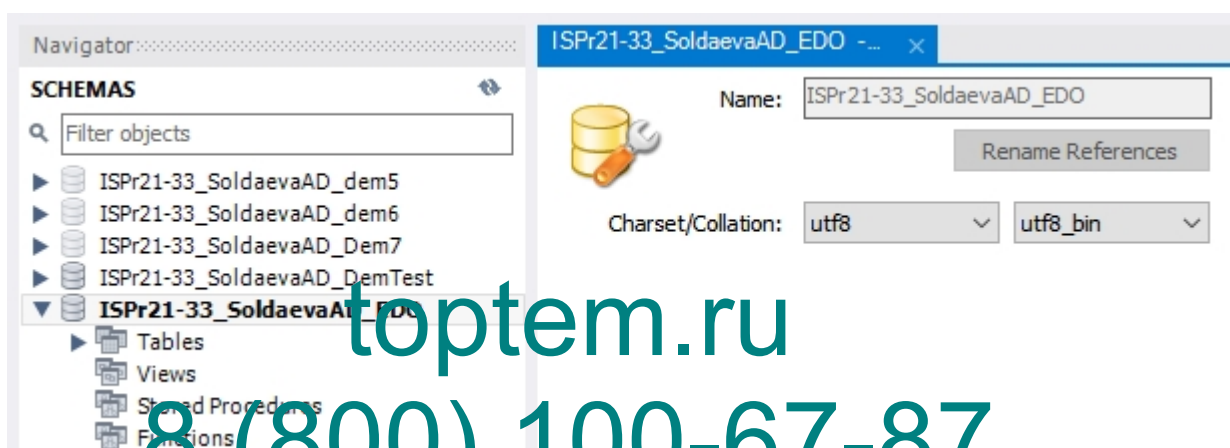


Рисунок 10 – Создание базы данных

Для установки соединения с базой данных посредством Entity Framework и обеспечения их работы необходимо использовать библиотеку `MySQL.Data.EntityFramework`, затем создать модель для взаимодействия Entity Framework и базы данных.

Структура таблиц представлена на рисунках 11 - 20.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID_BuildObject	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
BuildObjectName	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
BuildObjectPhoto	LONGBLOB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
BuildStatus	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Clients_id_Client	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 11 – Структура таблицы «BuildObject»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID_Client	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
FIO	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
INN	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Phone	BIGINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Adress	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
NameCompany	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 12 – Структура таблицы «Client»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID_Contract	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Number	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Date	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DateTermination	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Suppliers_ID_Supplier	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DocumentScan	LONGBLOB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Status	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 13 – Структура таблицы «Contract»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID_Material	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
ImageMaterial	LONGBLOB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Description	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Рисунок 14 – Структура таблицы «Material»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Materials_ID_Material	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Suppliers_ID_Supplier	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Material_Price	DECIMAL(10,0)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 15 – Структура таблицы «Materials_has_Suppliers»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
Materials_ID_Material	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Supply_ID_Supplier	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Material_Count	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Material_Price	DECIMAL(10,0)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 16 – Структура таблицы «Materials_has_Supply»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID_ObjectDocument	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
ObjectName	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
ObjectScan	LONGBLOB	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DocumentNumber	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
DocumentStatus	TINYINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
BuildObject_ID_BuildObject	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 17 – Структура таблицы «ObjectDocument»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID_Supplier	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Title	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Adress	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Phone	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Photo	LONGBLOB	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Рисунок 18 – Структура таблицы «Supplier»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
ID_Supply	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Suppliers_ID_Supplier	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
BuildObject_ID_BuildObject	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Date	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 19 – Структура таблицы «Supply»

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idUser	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Password	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Login	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
FIO	VARCHAR(60)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Role	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 20 – Структура таблицы «User»

Подключение удаленной БД происходит с помощью модуля Entity Framework. Строка подключения расположена в коде класса модели данных, а ее параметры расположены в параметрах программы, предоставленных средой разработки Visual Studio [30].

Для подключения программного продукта к удалённой базе данных требуется создание модели с помощью конструктора Entity Framework в Visual Studio. Для чего в проекте был создан новый элемент Entity

Framework, в разделе Data (ADO.NET Entity Data Model). После чего выбрана модель Code First from Database, которая автоматически создала аннотацию с конфигурацией и передала в конструктор базового класса строку соединения с базой следующего вида:

```
"<add name="EfModel" connectionString="server=cfif31.ru;
user id=ISPr21-33_SoldaevaAD; password=ISPr21-33_SoldaevaAD;
persistsecurityinfo=True;database=ISPr21-33_SoldaevaAD_EDO"
providerName="MySql.Data.MySqlClient"/>"
```

3.3 Реализация функциональных требований

Главное меню различно для всех ролей пользователей, но все страницы пользователей имеют общее меню. Администратор, и инженер могут добавлять, удалять или изменять данные. Меню администратора (рисунок 3.22) содержит вкладки на страницы «Договоры», «Материалы», «Поставщики», «Строительные объекты», «Строительная документация», «Добавить пользователя» и «Добавить подключение». При желании пользователь может скрыть панель меню и при наведении на иконку будет показываться подсказка в виде названия вкладки (рисунок 21).

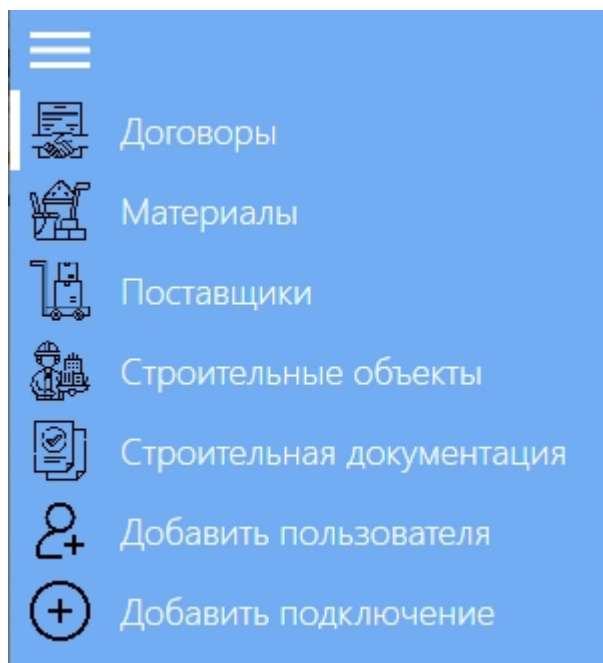


Рисунок 21 - Главное меню администратора

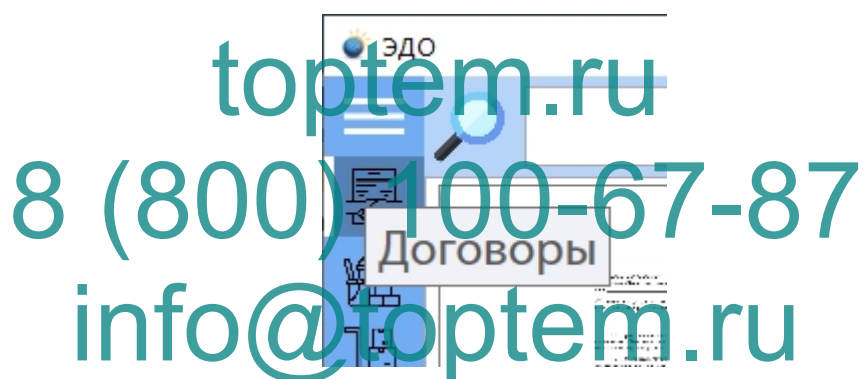


Рисунок 22- Подсказка меню

Главное меню инженера содержит вкладки таких страниц, как «Договоры», «Материалы», «Поставщики» (рисунок 23).

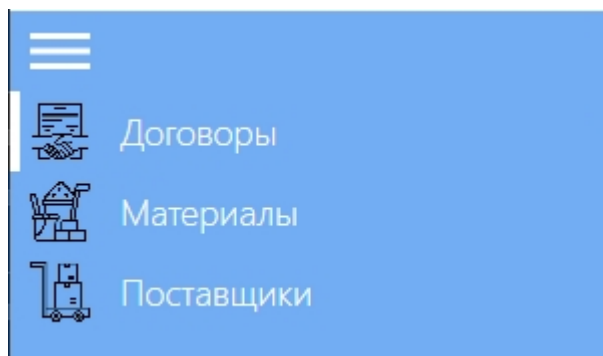


Рисунок 23 – Главное меню инженера

Список пользователей содержит информацию, такую как ФИО, фотография, логин, пароль и роль. На странице присутствует поиск и кнопки добавления, удаления и редактирования для выполнения операций над пользователями (рисунок 24).

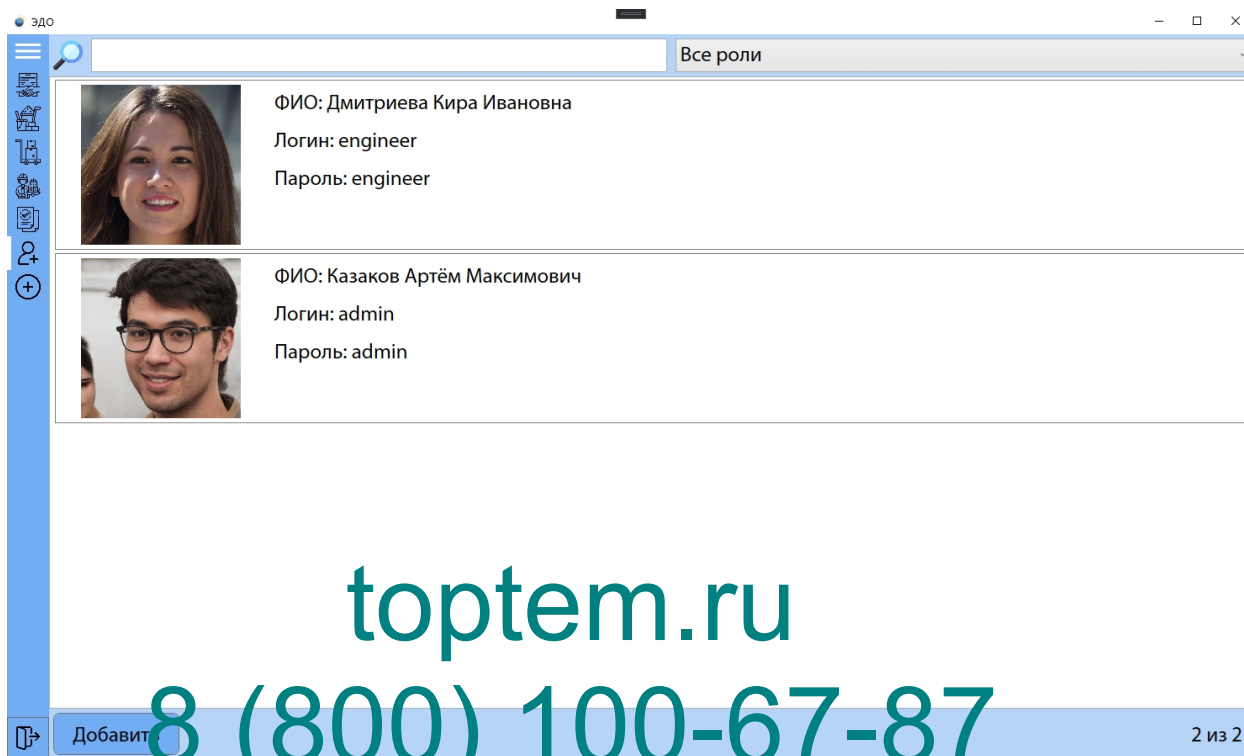


Рисунок 24 – Список пользователей

Страница добавления и изменения пользователя, содержит поля для ввода всех необходимых данных о пользователе, возможность загрузки фотографии в базу, а также поле для выбора роли пользователя в системе (рисунок 25).

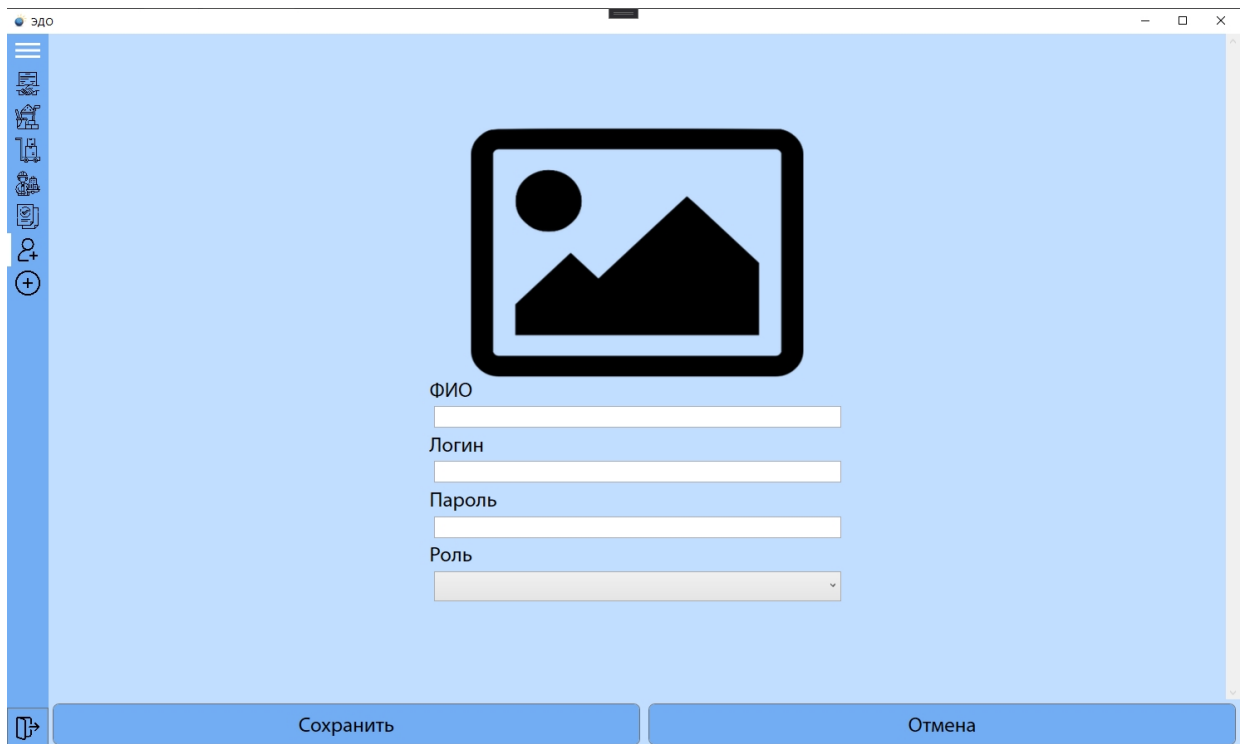


Рисунок 25- Страница добавления/изменения пользователей

toptem.ru
 Страница списка материалов (рисунок 26) содержит информацию о
 поступившем товаре, строку поиска, сортировку по наименованию и
 возрастанию/убыванию цены, фильтрацию по типу материала, кнопки
 добавления, удаления и изменения, а также кнопки импорта и экспорта.
info@toptem.ru

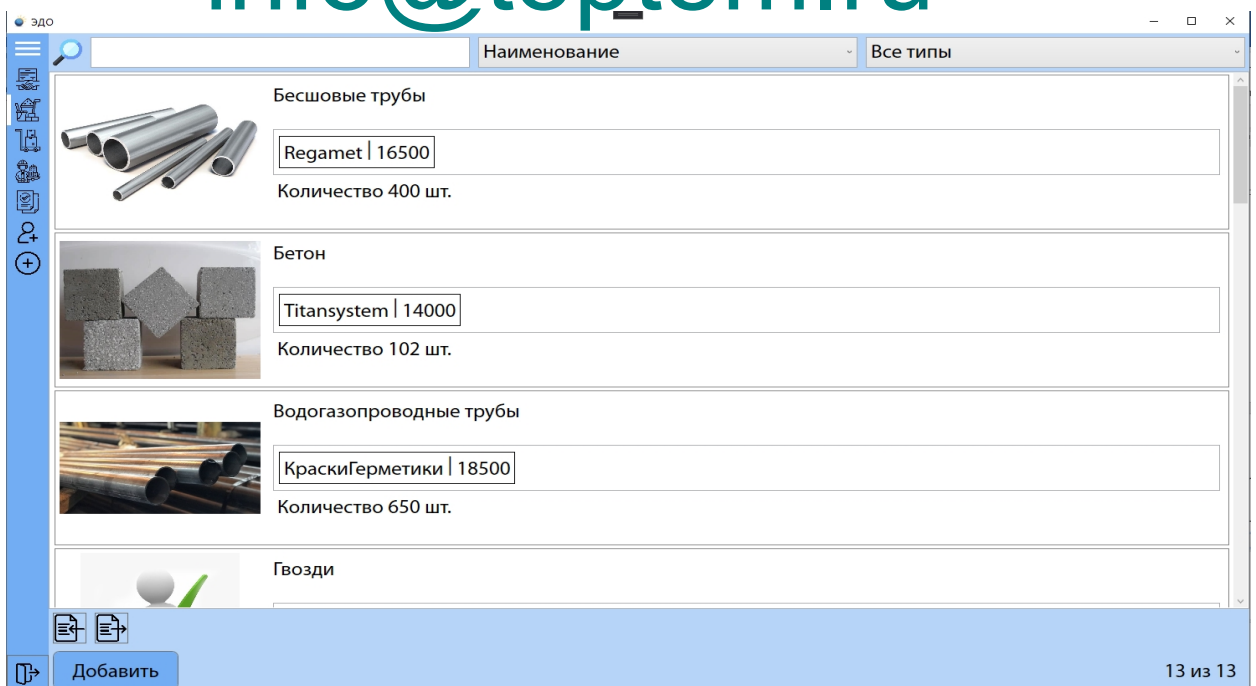


Рисунок 26– Страница списка материалов

Страница добавления и изменения данных о материале запрашивает ввод названия материала, описания, если оно требуется, выбор поставщика из выпадающего списка и ввод цены товара. После, данные добавляются в отдельный дополнительный список и сохраняются уже в основной список (рисунок 27).

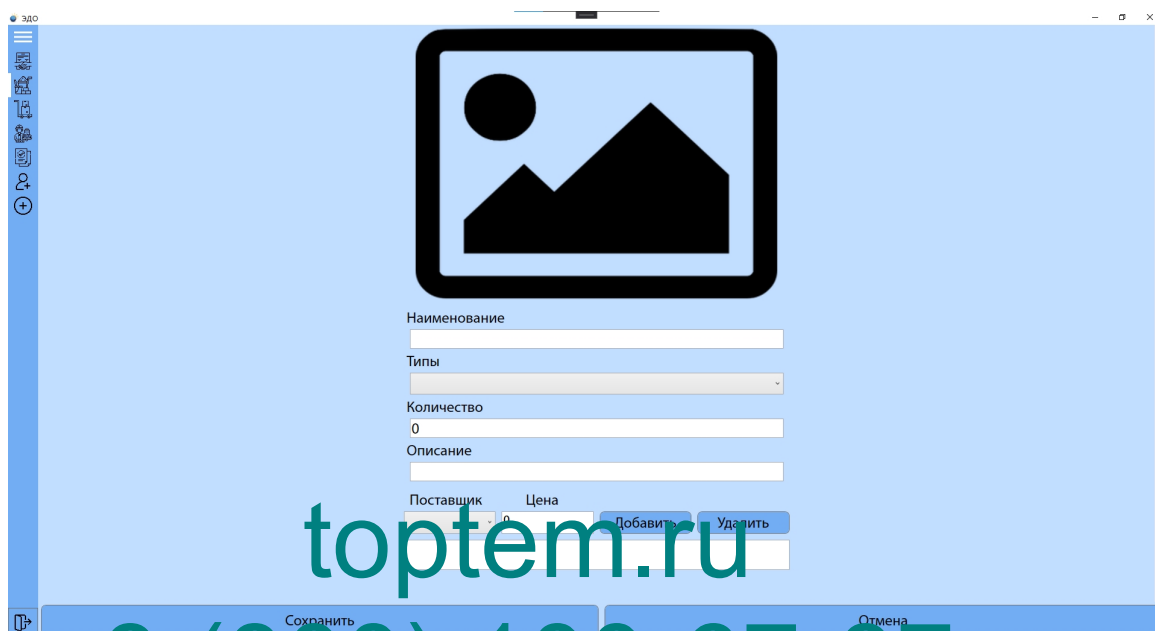


Рисунок 27 – Страница добавления и изменения данных о материале

Список договоров содержит следующие данные: наименование договора, наименование поставщика, номер договора, дата заключения и расторжения договора. Функционал данной страницы представляет собой: строку поиска, кнопки добавления, изменения, удаления, импорта, экспорта и добавление шаблона (рисунок 28). Также на странице присутствует фильтрация по статусу. Данная функция есть и на страницах «Строительная документация» (рисунок 29) и «Строительные объекты» (рисунок 30).

		Все статусы
	<p>Договор купли-продажи Керамтрейд № 22 Исполнитель: Маслов Пётр Степанович</p>	<p>Заключен 2021-02-02 Истекает 2022-06-02</p>
	<p>Договор поставки материалов Regamet № 98 Исполнитель: Егорова Виктория Ивановна</p>	<p>Заключен 2021-03-03 Истекает 2022-10-04</p>
	<p>Договор поставки КраскиГерметики № 456 Исполнитель: Иванов Илья Романович</p>	<p>Заключен 2021-02-07 Истекает 2022-05-30</p>
	<p>Договор строительного найма Deluxe</p>	<p>Заключен 2022-05-19</p>

Добавить 8 из 8

Рисунок 28- Список договоров

toptem.ru

8 (800) 100-07-87

info@toptem.ru

		Все статусы
	<p>Приказ № 44 Новый порт</p>	<p>Все статусы Обработанные Необработанные</p>
	<p>Распоряжение № 5 Новый порт</p>	
	<p>ГОСТ № 65 База</p>	
	<p>Приказ № 54</p>	

Добавить 5 из 5

Рисунок 29- Страница «Строительная документация»

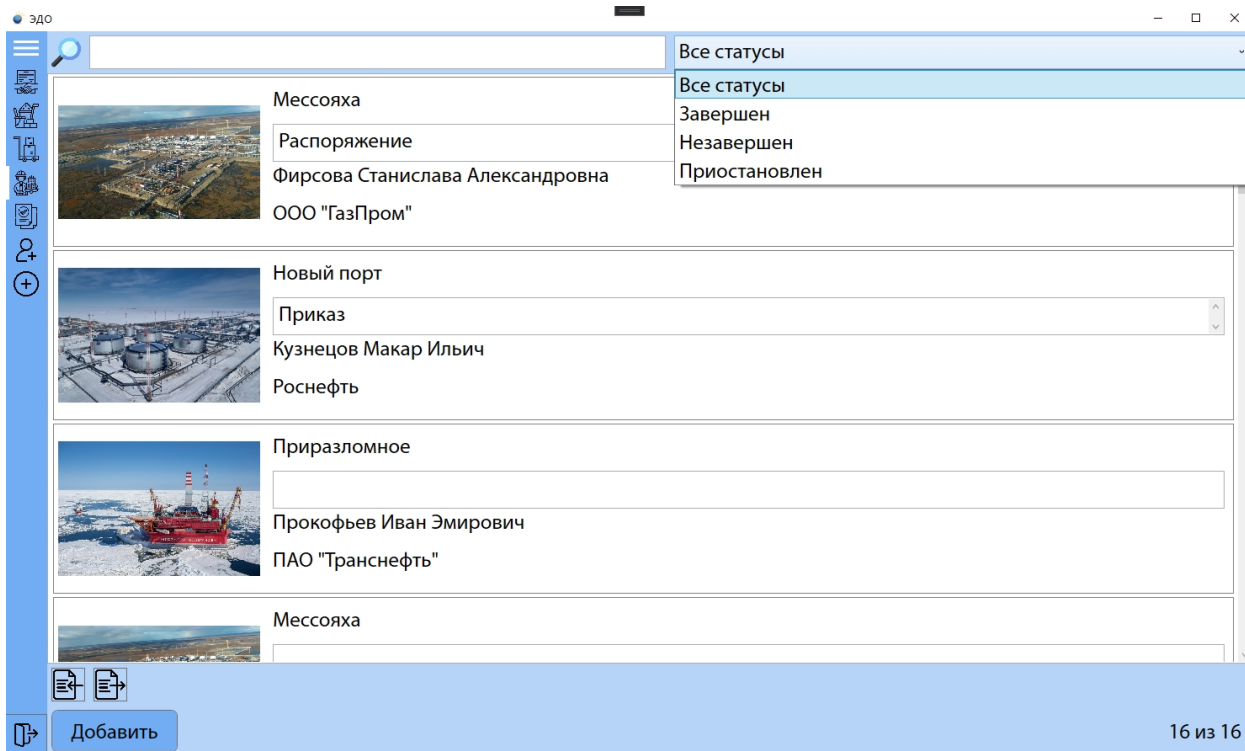


Рисунок 30 - Страница «Строительные объекты»

toptem.ru

Страница заполнения шаблона. Содержит поля для ввода данных:
 наименование договора, город, дата заключения, наименование компаний
 заказчика и поставщика, ФИО директоров компаний и постановления
 (рисунок 31) **8 (800) 100-67-87**
info@toptem.ru

Рисунок 31 - Страница заполнения шаблона

toptem.ru

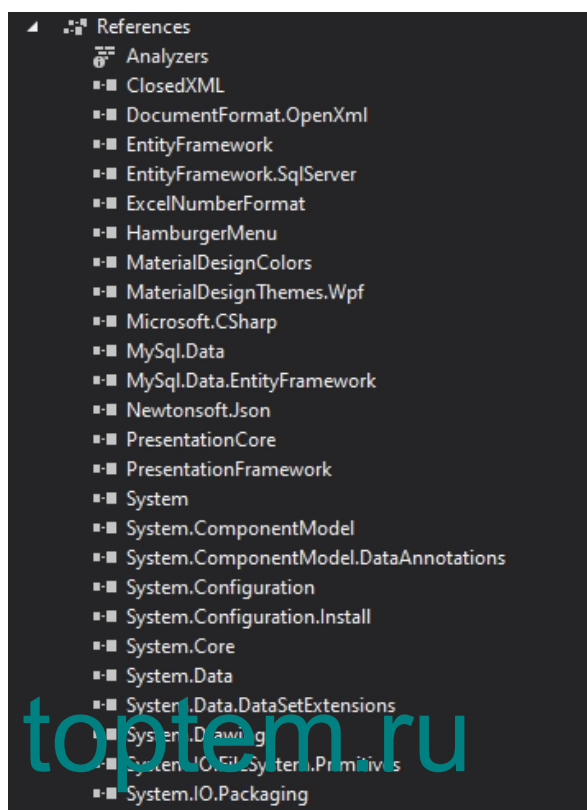
3.4. Интеграция программных модулей
8 (800) 100-67-87

info@toptem.ru

Для организации работы программного продукта с базой данных используется EntityFramework, для его использования вместе с MySQL требуется библиотека MySql.Data.Entity. Была использована модель разработки Code First from Database, которая автоматически перенесла данные из БД и создала готовые таблицы в коде. Обращение к сущностям базы данных осуществляется посредством обращения к полям модели базы с использованием методов LINQ [34].

Для генерации файла формата word так же используется сторонняя библиотека – OpenXML SDK, предоставляющая интерфейс для взаимодействия с файлами Word без использования com объектов. И позволяет использовать программный продукт без необходимости установки Microsoft Word на компьютер. При создании шаблонизатора использовалась

библиотека TemplateEngine.Docx (рисунок 32).



8 (800) 100-67-87

info@toptem.ru

3.5 Отладка программного продукта

Тестирование программного продукта нацелено на проверку основного функционала программного продукта: просмотр, удаление, добавление и редактирование информации о пользователях, студентах, менеджерах компетенций, самих компетенций, расписания, изменений в расписание, переключение языка программного продукта (таблицы 9-11).

Таблица 9

Функциональное тестирование

Функция	Действие	Ожидаемый результат	Полученный результат	Фатальные ошибки
Авторизация пользователей	Ввод данных для входа в поля Логин и Пароль, последующее нажатие кнопки «Войти»	Переход на страницу главного меню, в соответствии с ролью пользователя	Переход на страницу главного меню, в соответствии с ролью пользователя	Отсутствуют
Добавление данных	Ввод новых данных в соответствующее поле и нажатие кнопки «Сохранить»	Успешное сохранение данных.	Успешное сохранение данных.	Отсутствуют
Изменение данных.	Ввод новых данных в соответствующее поле и нажатие кнопки «Сохранить»	Успешное сохранение измененных данных.	Успешное сохранение измененных данных.	Отсутствуют
Сохранение информации	Отправка запроса к БД	Успешное сохранение	Успешное сохранение	Отсутствуют
Использование кнопок главного меню	Нажатие на кнопку	Переход на соответствующую страницу	Переход на соответствующую страницу	Отсутствуют
Поиск	Ввод данных в поле	Вывод данных согласно поисковому запросу	Вывод данных согласно поисковому запросу	Отсутствуют
Экспорт	Нажатие кнопки «Экспорт» на странице	Данные экспортированы в формате .xlsx	Данные экспортированы в формате .xlsx	Отсутствуют
Импорт	Нажатие кнопки «Импорт» на странице	Данные импортированы в список	Данные импортированы в список	Отсутствуют
Отмена	Нажатие на кнопку «Отмена», на странице Добавления/	Возвращает на начальную страницу	Возвращает на начальную страницу	Отсутствуют

	Редактирование			
Заполнение договоров по шаблону	Заполнение документа данными	Данные сохранены в формате .docx	Данные сохранены в формате .docx	Отсутствуют
Сортировка и фильтрация	Раскрытие списка данными	Изменения списка согласно выбранным данным	Изменения списка согласно выбранным данным	Отсутствуют

Таблица 10

Тестирование в определенной среде

Система	Разрешение экрана	Ошибки/несоответствия
Windows 7	1366x768	Нет
Windows 8	1280x1024	Нет
Windows 10	3840x2160	Нет
Windows 10	1920x1080	Нет

8 (800) 100-67-87

Таблица 11

Стресс-тестирование

Функция	Действие	Ожидаемый результат	Полученный результат	Фатальные ошибки
1 Авторизация пользователя	Ввод некорректных данных для входа в поле Логин и Пароль	Сообщение о неверном вводе данных	Сообщение о неверном вводе данных	Отсутствуют
2 Добавление данных	Ввод некорректных данных	Сообщение о неверном вводе данных	Сообщение о неверном вводе данных	Отсутствуют
3 Поля поиска	Ввод некорректных данных	Сообщение об отсутствии данных	Сообщение об отсутствии данных	Отсутствуют

3.6 Тестирование программного продукта

Для корректной работы программы в рабочих условиях необходимо произвести отладку модулей программного продукта, чтобы убедиться, что в процессе работы недоработки программного продукта не нарушили работу всей системы [39]. При отладке всех модулей программного продукта была выявлена проблема, связанная с перебоями в сети. Для того, чтобы избежать подобной проблемы в дальнейшем был прописан обработчик отлова ошибок и исключений.

```
"public App ()
{
    DispatcherUnhandledException +=
App_DispatcherUnhandledException;
}
private void App_DispatcherUnhandledException(object
sender,
System.Windows.Threading.DispatcherUnhandledExceptionEventArgs
e)
{
    MessageBox.Show("При работе с программой
произошла ошибка! Обратитесь к администратору!");
    e.Handled = true;
}
```

8 (800) 100-67-87

При отсутствии стабильного подключения аварийное завершение работы программы не происходит, а выдается сообщение об ошибке, и приложение продолжает дальше функционировать (рисунок 33).

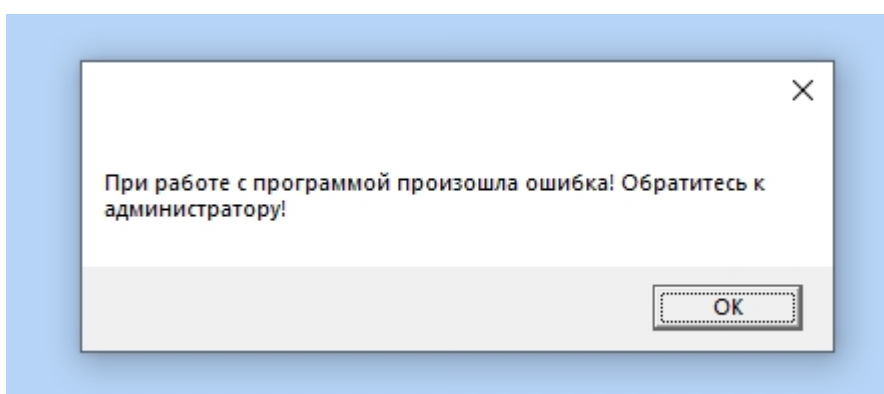


Рисунок 33 – Ошибка при отсутствии интернета

ГЛАВА 4 ДОКУМЕНТАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММНОГО ПРОДУКТА

4.1 Документированный программный код

Документирование кода является неотъемлемой частью при создании документационного сопровождения, но часто ей уделяется недостаточно внимания.

Документирование кода – это вставка в код определённых комментариев, которые позволяют в дальнейшем упростить работу с кодом, как автору, так и другим программистам. Чтобы было понятно, как пользователю, так и программистам, что есть что, то комментировать код нужно правильно: т.е. не повторять код, а описать что делает то, или иное действие, что означает эта переменная и как она используется в дальнейшем, с чем связывается. Если код имеет ссылку на библиотеку или класс, то желательно это тоже указывать. При документировании кода необходимо руководствоваться ГОСТ 19 402-78 ЕСПД «Описание программы» [8].

Полный листинг документированного кода программного продукта, представлен в приложении Г.

Пример документированного кода программного продукта приведен ниже.

```
    /// <summary>
    /// Редактируемый/добавляемый товар
    /// </summary>
    Contract contract;
    /// <summary>
    /// Конструктор страницы добавления/редактирования
    договора
    /// </summary>
    /// <param name="contract">Договор</param>
    public ContractEditPage(Contract contract)
    {
        //Записываем договор в переменную
        this.contract = contract;
```

```

        InitializeComponent();
        //Получаем список поставщиков из бд
        CbSuppliers.ItemsSource =
EfModel.Init().Suppliers.ToList();
        //Записываем договор в DataContext
        DataContext = this.contract;
    }
    /// <summary>
    /// Сохранение товара
    /// </summary>
    private void BtSaveContractClick(object sender,
RoutedEventArgs e)
    {
        //Отлавливаем исключения
        try
        {
            contract.Status =
(ContractStatus)CbStatus.SelectedIndex;
            //Если добавляем
            if (contract.ID_Contract == 0)
                //Добавляем договор в соответствующий
СПИСОК
                EfModel.Init().Contracts.Add(contract);
            EfModel.Init().SaveChanges();
            if (NavigationService.CanGoBack)
                NavigationService.GoBack();
            else MessageBox.Show("Данные сохранены");
            //Если случилось исключение проверки Entity
            catch (DbEntityValidationException ex)
            {
                MessageBox.Show(String.Join(", ",
ex.EntityValidationErrors.Last().ValidationErrors.Select(ve =>
ve.ErrorMessage)));
            }
        }
    }
    /// <summary>
    /// Метод загрузки картинки
    /// </summary>
    private void ImageChangeClick(object sender,
MouseButtonEventArgs e)
    {
        //Формируем диалог открытия картинки
        OpenFileDialog openFileDialog = new OpenFileDialog {
Filter = "Jpeg files|.jpg|All files|*.*" };
        //Открываем его
        if (openFileDialog.ShowDialog() == true)
        {
            //Если файл выбран - грузим его в
contract.DocumentScan
            contract.DocumentScan =
File.ReadAllBytes(openFileDialog.FileName);

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

}
}

4.2 Руководство пользователя программного продукта

Введение

Область применения

Требования настоящего документа применяются при:

- предварительных комплексных испытаниях;
- опытной эксплуатации;
- приёмочных испытаниях;
- промышленной эксплуатации.

Краткое описание возможностей

Программа позволяет автоматизировать процесс управления документооборотом в предприятии ООО «НефтеГазоТехнологии».

Уровень подготовки пользователя

Пользователь ПК должен иметь опыт работы с ОС MS Windows 10, а также обладать следующими знаниями:

- знать соответствующую предметную область;
- знать и иметь навыки работы с аналитическими приложениями.

Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

- Данное руководство пользователя.

Назначение и условия применения.

Автоматизация осуществляемых на предприятии процессов позволит хранить информацию в одной базе, информация в которую вводится с помощью удобного интерфейса.

Автоматизация позволит учесть все особенности учёта склада в

компании, то есть разработать только то, что нужно для данной компании.

Подготовка к работе

Состав и содержание дистрибутивного носителя данных

Для работы с ПО необходимо следующее программное обеспечение:

- операционная система Windows;
- набор библиотек .NET Framework для запуска программы, написанной на данной программной платформе;
- Google Chrome;
- драйвер MySQL Connector ODBC для подключения удаленной базы данных к программе.

Порядок загрузки данных и программ

Перед началом работы с приложением «Электронный документооборот» на рабочем месте пользователя необходимо выполнить следующие действия:

Выборить запускать файл с программой и дважды кликнуть мышкой на значке.

Порядок проверки работоспособности

Для проверки доступности приложением «Электронный документооборот» с рабочего места пользователя необходимо выполнить следующие действия:

- Открыть приложение, для этого необходимо дважды кликнуть мышкой на значке программы;
- Убедиться, что в окне открылось приложение «Электронный документооборот»;
- В случае если приложение не запускается, то следует обратиться в службу поддержки.

Описание операций

Выполняемые функции и задачи

Приложение «Электронный документооборот» выполняет функции и задачи, приведённые в таблице ниже (таблица 12).

Таблица 12

Функции и задачи программного продукта

Функции	Задачи	Описание
Запуск программы, прохождение аутентификации	Аутентификация пользователя	При запуске программы требуется ввод логина и пароля для входа в главное меню
Поле для поиска	Быстрый поиск по базе данных	При введении искомого запроса результат выводится в окне со списком данных
Работа с главным меню	Реализация второстепенных функций программы	Появляется меню программы
Добавление, редактирование данных	Добавление, редактирование данных в программе, обновляя удаленную базу данных	Добавление, изменение, удаление данных из базы данных из приложения
Экспорт и импорт данных	Экспортирует и импортирует выбранные данные из базы	При нажатии на кнопку открывается системное окно для выбора импортируемого/экспортируемого файла
Заполнение документов по шаблону	Быстрое заполнение документов	Открывается страница заполнения шаблона, в которую вписываются данные, после сохраняется

Описание операций технологического процесса обработки данных, необходимых для выполнения задач [2].

Ниже приведено описание пользовательских операций для выполнения каждой из задач.

Задача: «Аутентификация пользователя»

Операция 1: запуск программы

Условия, при соблюдении которых возможно выполнение операции:

1. Компьютер пользователя загружен;
2. Приложение установлено.

Подготовительные действия:

Запустить программу.

Основные действия в требуемой последовательности:

На иконке «Авторизация» рабочего стола произвести двойной щелчок левой кнопкой мышки.

Заключительные действия:

Не требуются.

Ресурсы, расходуемые на операцию:

12 секунд.

Основные действия в требуемой последовательности:

1. В появившемся окне ввести логин, пароль.

Заключительные действия:

После ввода данных выбрать пункт «Вход».

Ресурсы, расходуемые на операцию:

35 секунд.

Операция 3: ввод данных

Условия, при соблюдении которых возможно выполнение операции:

Приложение установлено.

Подключение к базе данных настроено.

Подготовительные действия:

Не требуются.

Основные действия в требуемой последовательности:

2. В появившемся окне ввести логин и пароль для входа в программу.

Заключительные действия:

После ввода данных выбрать пункт «Войти».

Ресурсы, расходуемые на операцию:

15 секунд.

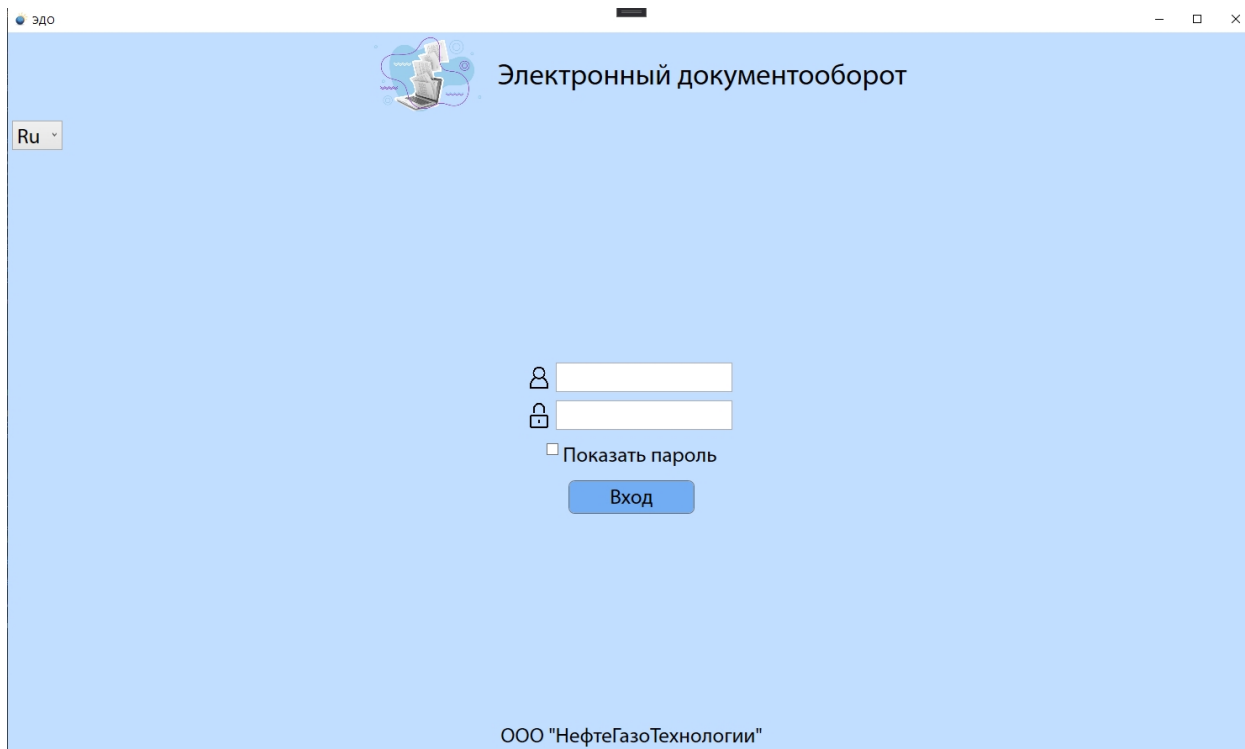


Рисунок 34 – Задача «Авторизация»

- toptem.ru**
- 8 (800) 100-67-87**
- info@toptem.ru**
- Задача: «Поиск»**
- Условия, при соблюдении которых возможно выполнение операции:*
Приложение запущено.
- Подготовительные операции:*
Не требуются.
- Основные действия в требуемой последовательности:*
1. Ввести критерий для быстрого поиска по базе данных в строку.
- Заключительные действия:*
Очистить строку поиска.
- Ресурсы, расходуемые на операцию:*
10-15 секунд.

		Все статусы
	Договор купли-продажи Керамтрейд № 22 Исполнитель: Маслов Пётр Степанович	Заключен 2021-02-02 Истекает 2022-06-02
	Договор поставки материалов Regamet № 98 Исполнитель: Егорова Виктория Ивановна	Заключен 2021-03-03 Истекает 2022-10-04
	Договор поставки КраскиГерметики № 456 Исполнитель: Иванов Илья Романович	Заключен 2021-02-07 Истекает 2022-05-30
	Договор строительного найма Deluxe	Заключен 2022-05-19

Добавить 8 из 8

Рисунок 35 – Задача поиск

toptem.ru

Задача: «Добавить договор»

Условия, при соблюдении которых возможно выполнение операции:
 Приложение запущено.

Подготовительные действия:
 Не требуются.

Основные действия в требуемой последовательности:

1. На главной форме Системы выбрать «Договор»;
2. Заполнить поля в появившемся окне.

Заключительные действия:

Нажать кнопку добавить.

Ресурсы, расходуемые на операцию:

Зависит от количества вводимых данных.



Рисунок 36 – Задача «Добавить договор»

toptem.ru
8 (800) 100-67-87
info@toptem.ru

Задача: «Редактировать товар»
Условия, при соблюдении которых возможно выполнение операции:
Приложение запущено.
Подготовительные действия:
Не требуются.

Основные действия в требуемой последовательности:

1. На главной форме Системы выбрать «Договор»;
2. Заполнить поля в появившемся окне.

Заключительные действия:

Нажать кнопку изменить.

Ресурсы, расходуемые на операцию:

Зависит от количества вводимых данных.

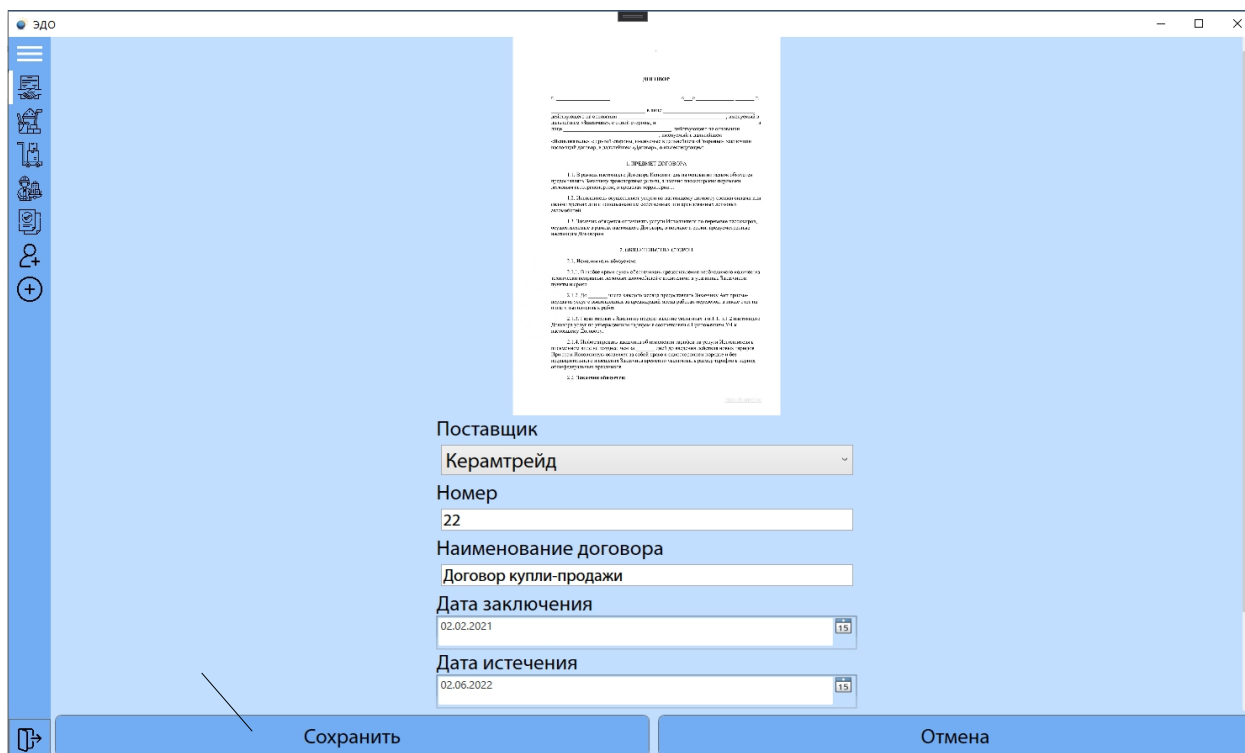


Рисунок 37 – Задача «Редактировать договор»

toptem.ru

Задача: «Удалить договор из базы данных»

Условия, при соблюдении которых возможно выполнение операции:

Приложение запущено.

Подготовительные действия:

Не требуются.

Основные действия в требуемой последовательности:

1. Выбрать договор из базы данных.

Заключительные действия:

Нажать на кнопку удалить.

Ресурсы, расходуемые на операцию:

Зависит от количества вводимых данных.

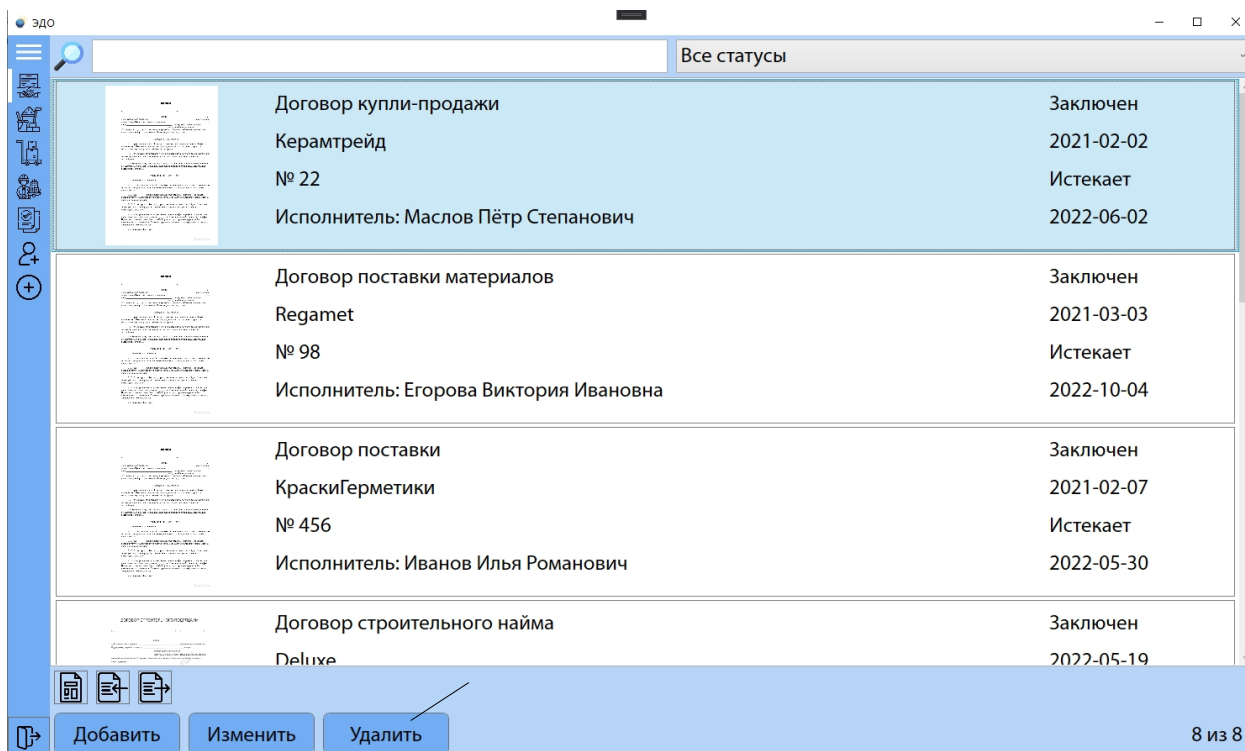


Рисунок 38 – Задача «Удалить договор из базы данных»

Аварийные ситуации

В случае возникновения ошибки в Электронном документообороте пользователю выводится сообщение об ошибке подобного плана:

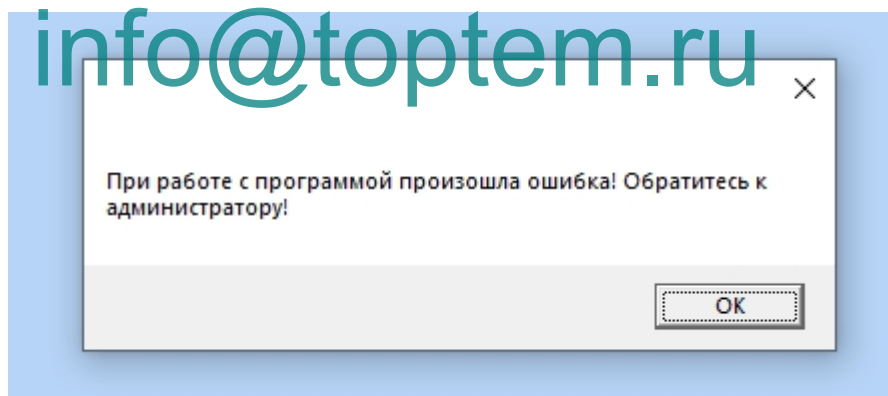


Рисунок 39 - Ошибка

4.3 Руководство администратора (включая инструкцию по установке)

Наименование программы

Программа Электронный документооборот. Программа написана на языке программирования C# в среде разработки Visual Studio 2019.

Назначение программы

Данная программа предназначена для автоматизации документооборота предприятия.

Условия выполнения программы

Программа была разработана на компьютере с ОС Windows 10. Процессор core i7 9700k, 64-разрядная операционная система, клавиатура и мышь. 128 ГБ оперативной памяти. При таких условиях программа будет функционировать. Обязательно требуется установленный в системе .Net Framework 4.8 и доступ к серверу MySql.

Программа Электронный документооборот занимает 242 МБ (254 201 574 байт) дискового пространства.

Выполнение программы

Для работы программного продукта необходима база данных MySql, расположенная на сервере локально, или удаленно. Для настройки подключения к базе данных — в соответствующей странице необходимо задать следующие параметры:

- Адрес сервера MySql
- Порт сервера
- Имя пользователя
- Пароль пользователя
- Название БД



ЭДО

Сервер
cwf31.ru

Порт
3306

Наименование БД
ISPr21-33_SoldaevaAD_EDO

Имя пользователя
ISPr21-33_SoldaevaAD

Пароль
ISPr21-33_SoldaevaAD

Сохранить Отмена

Рисунок 40 – Настройка соединения с базой

toptem.ru

Начало работы

Для начала работы с новой базой данных достаточно просто запустить приложение и задать соответствующие поля все данные, необходимые для подключения к БД, после чего приложение автоматически создаст необходимые таблицы и поля в базе данных. Затем при необходимости можно импортировать дампы старой БД, либо создать данные вручную [10].

ГЛАВА 5 ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ПРОГРАММНОГО ПРОДУКТА

5.1 Разработка мер защиты информации от несанкционированного доступа

На данном этапе разрабатываются меры защиты информации от несанкционированного доступа и мероприятия по обеспечению техники безопасности и пожарной безопасности.

Разработка мер защиты информации от несанкционированного доступа

Защита программного обеспечения преследует цели:

- ограничение несанкционированного доступа к программам или их преднамеренное разрушение и хищение;
- исключение несанкционированного копирования (тиражирования) программ.

Программный продукт и базы данных должны быть защищены по нескольким направлениям от воздействия:

1. человека — хищение машинных носителей и документации программного обеспечения; нарушение работоспособности программного продукта и др.;
2. аппаратуры — подключение к компьютеру аппаратных средств для считывания программ и данных или их физического разрушения;
3. специализированных программ — приведение программного продукта или базы данных в неработоспособное состояние (например, вирусное заражение), несанкционированное копирование программ и базы данных и т.д.

Самый простой и доступный способ защиты программных продуктов и

базы данных — ограничение доступа. Контроль доступа к программному продукту и базе данных строится путем:

- парольной защиты программ при их запуске;
- использования ключевой дискеты для запуска программ;
- ограничения программ или данных, функций обработки, доступных пользователям, и др.

В программе защита от несанкционированного доступа осуществляется посредством авторизации, которая запускается первой (рисунок 41).

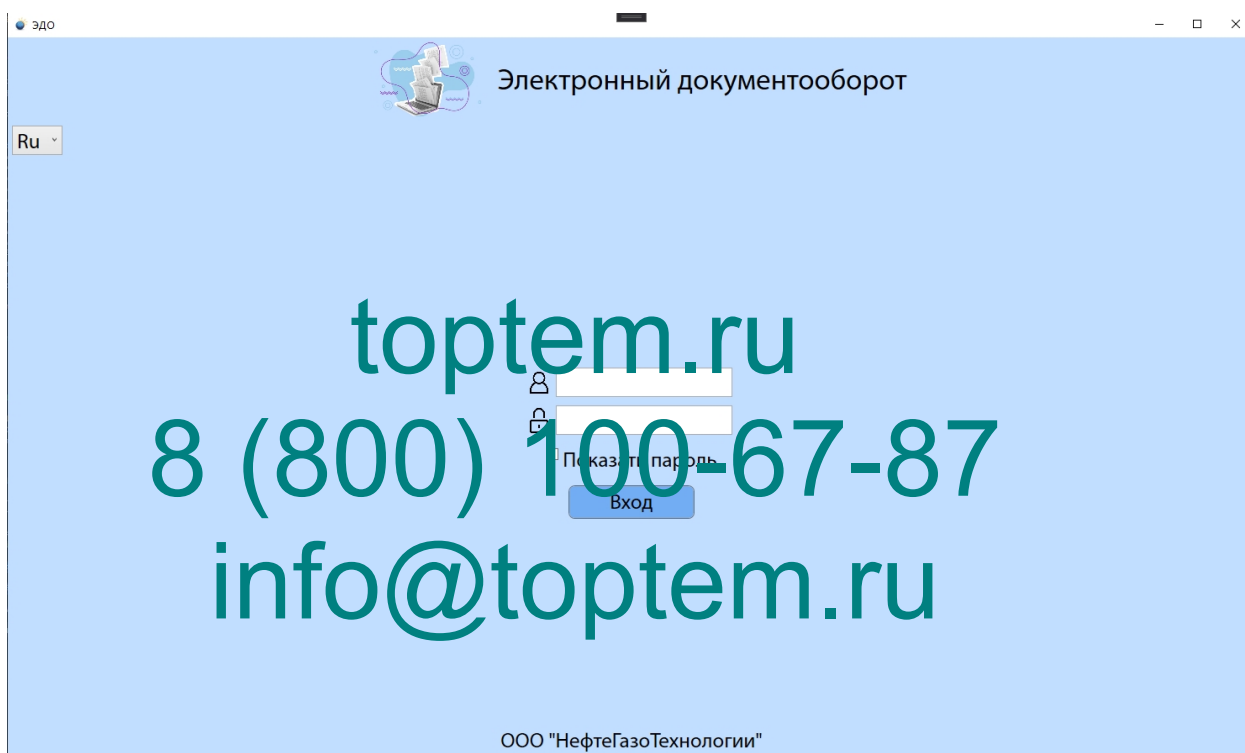


Рисунок 41 - Страница авторизации

5.2 Мероприятия по обеспечению техники безопасности и пожарной безопасности

Общие требования безопасности

Настоящая инструкция распространяется на персонал, эксплуатирующий средства вычислительной техники и периферийное

оборудование. Инструкция содержит общие указания по безопасному применению электрооборудования в учреждении. Требования настоящей инструкции являются обязательными, отступления от нее не допускаются. К самостоятельной эксплуатации электроаппаратуры допускается только специально обученный персонал не моложе 18 лет, пригодный по состоянию здоровья и квалификации к выполнению указанных работ.

Требования безопасности перед началом работы

Перед началом работы следует убедиться в исправности электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, наличии заземления компьютера, его работоспособности.

Требования безопасности во время работы

Для снижения или предотвращения влияния опасных и вредных факторов необходимо соблюдать санитарные правила и нормы. Гигиенические требования к видео дисплейным терминалам, персональным электронно-вычислительным машинам и организации работы (Утверждено Постановлением Госкомсанэпиднадзора России от 14 июля 1996 г. N 14 СанПиН 2.2.2.541-96)

Во избежание повреждения изоляции проводов и возникновения коротких замыканий не разрешается: вешать что-либо на провода, закрашивать и белить шнуры и провода, закладывать провода и шнуры за газовые и водопроводные трубы, за батареи отопительной системы, выдергивать штепсельную вилку из розетки за шнур, усилие должно быть приложено к корпусу вилки.

Для исключения поражения электрическим током запрещается: часто включать и выключать компьютер без необходимости, прикасаться к экрану и к тыльной стороне блоков компьютера, работать на средствах вычислительной техники и периферийном оборудовании мокрыми руками, работать на средствах вычислительной техники и периферийном

оборудовании, имеющих нарушения целостности корпуса, нарушения изоляции проводов, неисправную индикацию включения питания, с признаками электрического напряжения на корпусе, класть на средства вычислительной техники и периферийное оборудование посторонние предметы.

Запрещается под напряжением очищать от пыли и загрязнения электрооборудование.

Запрещается проверять работоспособность электрооборудования в непригодных для эксплуатации помещениях с токопроводящими полами, сырых, не позволяющих заземлить доступные металлические части.

Недопустимо под напряжением проводить ремонт средств вычислительной техники и периферийного оборудования. Ремонт электроаппаратуры производится только специалистами-техниками с соблюдением необходимых технических требований.

Во избежание поражения электрическим током, при пользовании электроприборами нельзя касаться одновременно каких-либо трубопроводов, батарей отопления, металлических конструкций, соединенных с землей.

При пользовании электроэнергией в сырых помещениях соблюдать особую осторожность.

Требования безопасности в аварийных ситуациях

При обнаружении неисправности немедленно обесточить электрооборудование, оповестить администрацию. Продолжение работы возможно только после устранения неисправности.

При обнаружении оборванного провода необходимо немедленно сообщить об этом администрации, принять меры по исключению контакта с ним людей. Прикосновение к проводу опасно для жизни.

Во всех случаях поражения человека электрическим током немедленно вызывают врача. До прибытия врача нужно, не теряя времени, приступить к оказанию первой помощи пострадавшему.

На рабочем месте запрещается иметь огнеопасные вещества.

При возникновении пожароопасной ситуации или пожара персонал должен немедленно принять необходимые меры для его ликвидации, одновременно оповестить о пожаре администрацию.

Помещения с электрооборудованием должны быть оснащены огнетушителями типа ОУ-2 или ОУБ-3.

Требования безопасности по окончании работы

После окончания работы необходимо обесточить все средства вычислительной техники и периферийное оборудование. В случае непрерывного производственного процесса необходимо оставить включенными только необходимое оборудование.

toptem.ru
8 (800) 100-67-87
info@toptem.ru

ЗАКЛЮЧЕНИЕ

В ходе выполнения данного дипломного проекта разработано программное обеспечение, предназначенное для сбора, обработки и представления информации в соответствии с составленным техническим заданием. Программа внедрена в опытную эксплуатацию для производительности труда благодаря упрощению процессов согласования и передачи договоров, актов и другой строительной документации. Данный процесс модернизирует порядок дел в компании, повысит эффективность и защитит от ошибок. Был произведен выбор соответствующих программ. Была разработана удаленная база данных для хранения информации.

База данных была связана с программным продуктом и все функции программного продукта оптимизированы. Программный продукт отвечает всем требованиям, которые были поставлены по техническому заданию. При выполнении работы над дипломным проектом гипотеза была подтверждена, выполнены все поставленные задачи, а именно:

- проведен анализ объекта, процесса, предметной области как объекта информатизации;
- проведен анализ информационного обеспечения объекта информатизации;
- построены организационные и функциональные схемы объектов информатизации;
- выполнено описание категорий пользователей и потоков данных объекта информатизации;
- проведен анализ функциональных требований программного продукта;
- проведен анализ входных и выходных данных;
- построена структура данных;
- разработаны тестовые сценарии, программы и методики испытаний;

- разработано техническое задание для программного продукта;
- разработаны и оптимизированы схемы данных для программного продукта;
- выбрано программное обеспечение для реализации (разработки) программного продукта;
- выбрано программное обеспечение для разработки и эксплуатации удаленных базы данных;
- разработан интерфейс на русском и английском языках для программного продукта;
- разработана удаленная база данных и необходимые запросы;
- реализованы функциональные требования;
- выполнено описание интеграции программных модулей;
- выполнена отладка программного продукта;
- проведено тестирование программного продукта;
- выполнено документирование программного кода;
- разработано руководство пользователя для программного продукта;
- разработаны меры защиты информации от несанкционированного доступа;
- выполнено описание мероприятий по обеспечению техники безопасности и пожарной безопасности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Нормативные материалы

1. РД 50-34.698-90 АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ДОКУМЕНТОВ (в редакции 2019 года) (дата обращения: 31.05.2022).
2. РД 50-34.698-90. Подразделом 3.4 Руководство пользователя автоматизированной системы (в редакции 2019 года) (дата обращения: 31.05.2022).
3. ГОСТ 19.101-77 Виды программ и программных документов. – М.: Стандартформ, (в редакции 2019 года) (дата обращения: 31.05.2022). – 2 с.
4. ГОСТ 19.102-77 ЕСПД. Стадии разработки. – М.: Стандартформ, (в редакции 2019 года) (дата обращения: 31.05.2022).
5. ГОСТ 19.105-78 Общие требования к программным документам. – М.: Стандартформ (в редакции 2019 года) (дата обращения: 31.05.2022).
6. ГОСТ 19.201-78 ЕСПД. Техническое задание. Требования к содержанию и оформлению. – М.: Стандартформ, (в редакции 2019 года) (дата обращения: 31.05.2022).
7. ГОСТ 19.301-79 ЕСПД. Порядок и методика испытаний. – М.: Стандартформ, (в редакции 2019 года) (дата обращения: 31.05.2022).
8. ГОСТ 19.402-78 ЕСПД. Описание программы. – М.: Стандартформ, (в редакции 2019 года) (дата обращения: 31.05.2022).
9. ГОСТ 19.502-78 ЕСПД. Описание применения. Требования к содержанию и оформлению. – М.: Стандартформ, (в редакции 2019 года) (дата обращения: 31.05.2022).
10. ГОСТ 19.503-79 Руководство системного программиста. – М.:

- Стандартформ, (в редакции 2019 года) (дата обращения: 31.05.2022).
11. ГОСТ 19.504-79 Руководство программиста. – М.: Стандартформ, (в редакции 2019 года) (дата обращения: 31.05.2022).
12. ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. – М.: Стандартформ, (в редакции 2019 года) (дата обращения: 31.05.2022).

Научные, технические и учебно-методические издания

13. Албахари Б., Албахари Д. С# 7.0. Справочник. Полное описание языка. – М.: Издательский дом «Вильямс», 2018. – 1024 с.
14. Афонин, В. В. Моделирование систем / В.В. Афонин, С.А. Федосин. – М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2021. – 231 с.
15. Бёрд, Ричард Жемчужины проектирования алгоритмов. Функциональный подход / Ричард Бёрд. – М.: ДМК Пресс, 2019. – 330 с.
16. Васильев А. Программирование на С# для начинающих. Особенности языка. – М.: Издат. Дом «Бомбора», 2018. – 528 с.
17. Васильев А. С# 6.0. Программирование на С# для начинающих. Основные сведения. – М.: Издательский дом «Эксмо», 2018. – 592 с.
18. Гагарина, Л. Г. Разработка и эксплуатация автоматизированных информационных систем. Учебное пособие / Л.Г. Гагарина. – М.: Форум, Инфра-М, 2018. – 384 с.
19. Гвоздева, В. А. Введение в специальность программиста / В.А. Гвоздева. – М.: Форум, Инфра-М, 2018. – 208 с.
20. Гончаров, В. А. Методы оптимизации. Учебное пособие / В.А. Гончаров. – М.: Юрайт, 2018. – 191 с.

21. Зыков, С. В. Основы современного программирования / С.В. Зыков. – М.: Горячая линия – Телеком, 2021. – 400 с.
22. Исаев, Г. Н. Проектирование информационных систем. Учебное пособие / Г.Н. Исаев. – М.: Омега-Л, 2018. – 248 с.
23. Макаровских, Т. А. Документирование программного обеспечения. В помощь техническому писателю. Учебное пособие / Т.А. Макаровских. – М.: Ленанд, 2018. – 266 с.
24. Макконнелл С. Совершенный код. – М.: Издательский дом «Русская редакция», 2017. – 896 с.
25. Мартин Р. Чистый код. Создание, анализ и рефакторинг. Библиотека программиста. – П.: Издат. Дом «Питер», 2017. – 464 с.
26. Рихтер Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#. 4-е изд., 2016, с. 70-105.
27. Соколова, В.Е. Вычислительная техника и информационные технологии. Разработка мобильных приложений. Учебное пособие для прикладного бакалавриата / В.В. Соколова. – М.: Юрайт, 2020. – 175 с.
28. Стиллмен Э., Грин Д. HeadFirst. Изучаем C#. 3-е изд. 2017. – 384 с.
29. Фленов И. Библия C#. – П.: Изд. Дом «ВНУ», 2021. – 544 с.

Ресурсы сети Интернет

30. Википедия – свободная энциклопедия: Описание Microsoft Visual Studio [Электронный ресурс] // URL: https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата обращения: 31.05.2022).
31. Википедия – свободная энциклопедия: Описание Workbench [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/MySQL_Workbench (дата обращения: 31.05.2022).

32. Википедия – свободная энциклопедия [Электронный ресурс]:
Описание MySQL – Режим доступа: <https://ru.wikipedia.org/wiki/MySQL>
(дата обращения: 31.05.2022).
33. Википедия – свободная энциклопедия [Электронный ресурс]:
Описание Oracle 12c – Режим доступа:
https://ru.wikipedia.org/wiki/Oracle_Database (дата обращения:
31.05.2022).
34. Википедия – свободная энциклопедия [Электронный ресурс]:
Описание Microsoft SQL сервер – Режим доступа:
https://ru.wikipedia.org/wiki/Microsoft_SQL_Server (дата обращения:
31.05.2022).
35. Википедия – свободная энциклопедия [Электронный ресурс]:
Описание Microsoft Visual Studio – Режим доступа:
https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio (дата обращения:
31.05.2022).
36. Википедия – свободная энциклопедия [Электронный ресурс]:
Описание Java – Режим доступа: <https://ru.wikipedia.org/wiki/Java> (дата
обращения: 31.05.2022).
37. Основы программной инженерии [Электронный ресурс]:
http://swebok.sorlik.ru/4_software_testing.html (дата обращения:
31.05.2022).
38. Ресурс для IT-специалистов [Электронный ресурс]: MySQL:
особенности и сферы применения – Режим доступа:
<https://www.bytemag.ru/articles/detail.php?ID=6547> (дата обращения:
31.05.2022).
39. Ресурс для IT-специалистов [Электронный ресурс]: Методы
тестирования ПО – Режим доступа: [http://juice-
health.ru/program/software-testing/495-software-testing-methods](http://juice-health.ru/program/software-testing/495-software-testing-methods) (дата
обращения: 31.05.2022).

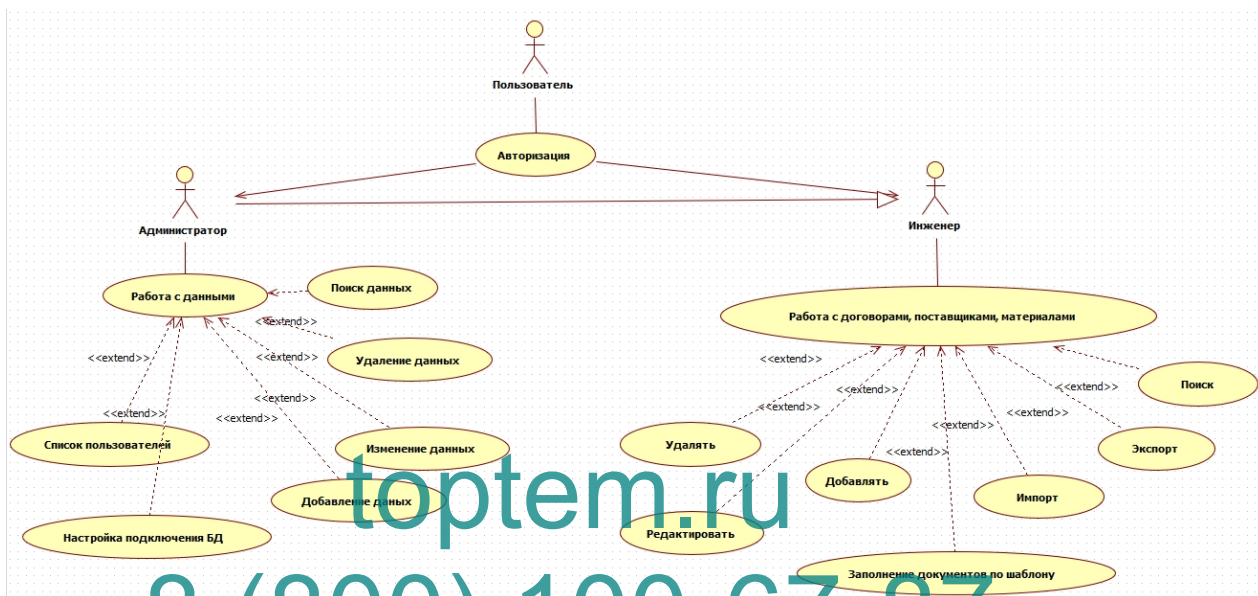
40. Ресурс для IT-специалистов [Электронный ресурс]: Оператор SQL SELECT – Режим доступа: [https://ru.wikipedia.org/wiki/Select_\(SQL\)](https://ru.wikipedia.org/wiki/Select_(SQL)) (дата обращения: 31.05.2022).
41. Ресурс для IT-специалистов [Электронный ресурс]: Оператор SQL UPDATE – Режим доступа: <http://2sql.ru/novosti/sql-update/> (дата обращения: 31.05.2022).
42. Ресурс для IT-специалистов [Электронный ресурс]: Работа с классами с# – Режим доступа: <https://metanit.com/sharp/tutorial/3.1.php> (дата обращения: 31.05.2022).
43. Ресурс для IT-специалистов [Электронный ресурс]: Интервьюирование: особенности интервью – Режим доступа: <https://students-library.com/library/read/25320-intervuirovanie-osobennosti-i-vidy-intervu> (дата обращения 31.05.2022).

toptem.ru
8 (800) 100-67-87
info@toptem.ru

ПРИЛОЖЕНИЕ А

UML-диаграммы

UML-диаграммы



8 (800) 100-67-87

Рисунок А.1 – UML-диаграмма вариантов использования

info@toptem.ru

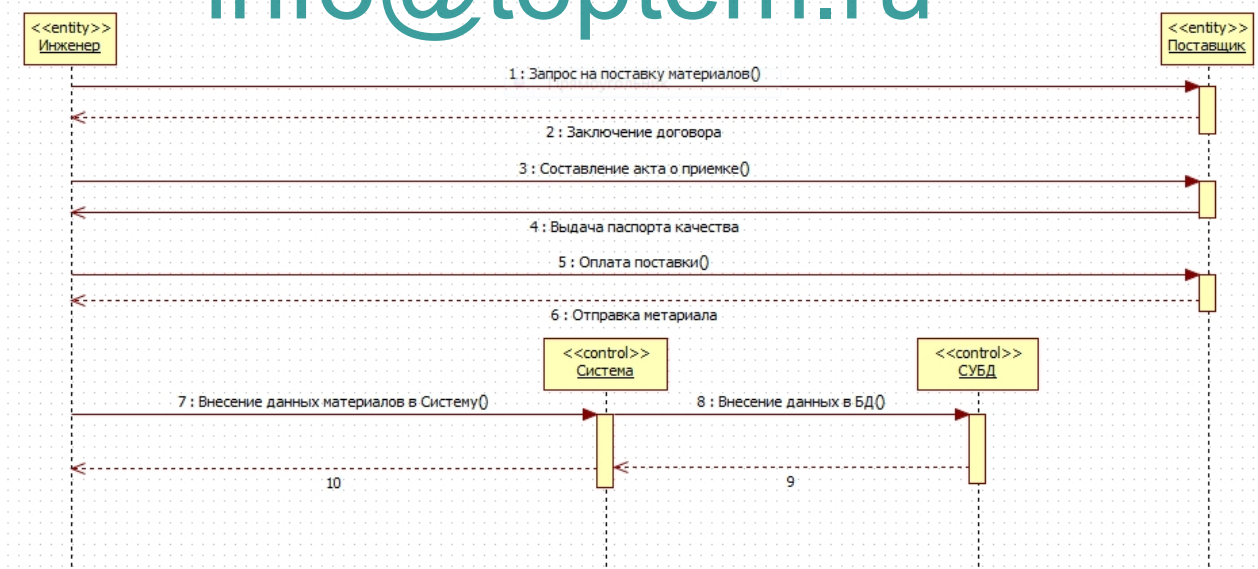


Рисунок А.2 – UML-диаграмма последовательностей

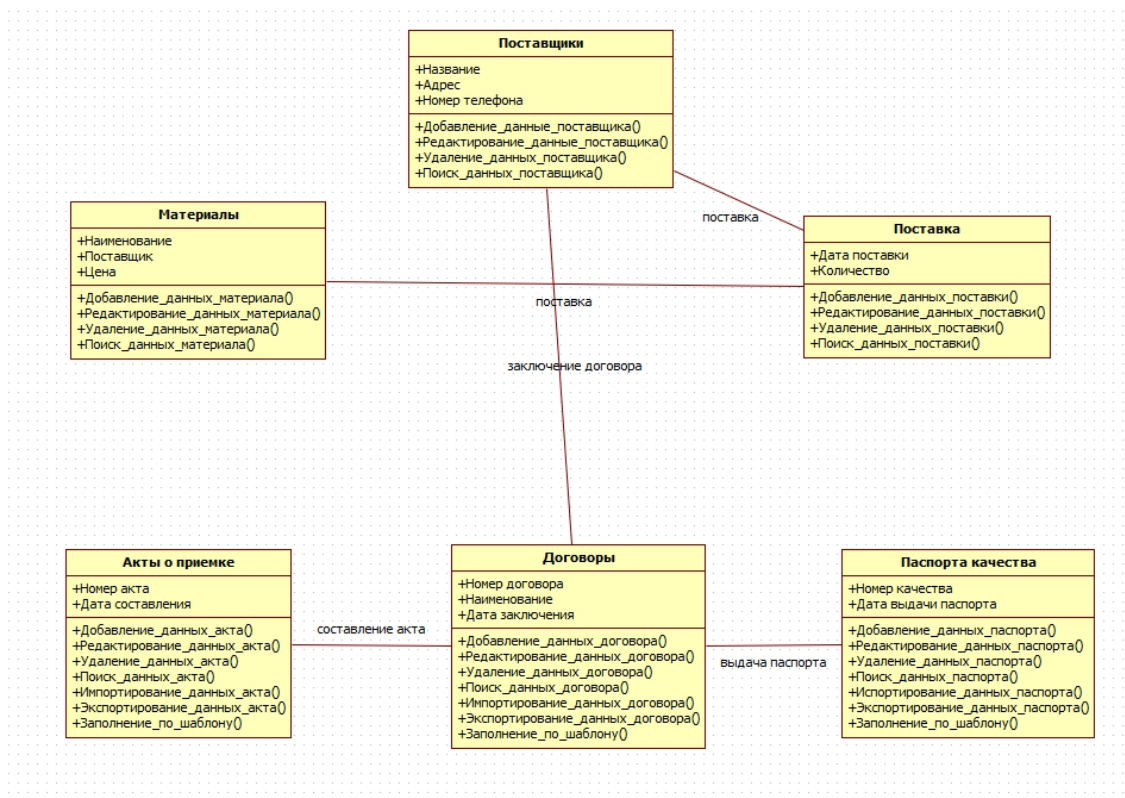


Рисунок А.3 – UML-диаграмма классов

toptem.ru
 8 (800) 100-67-87
 info@toptem.ru

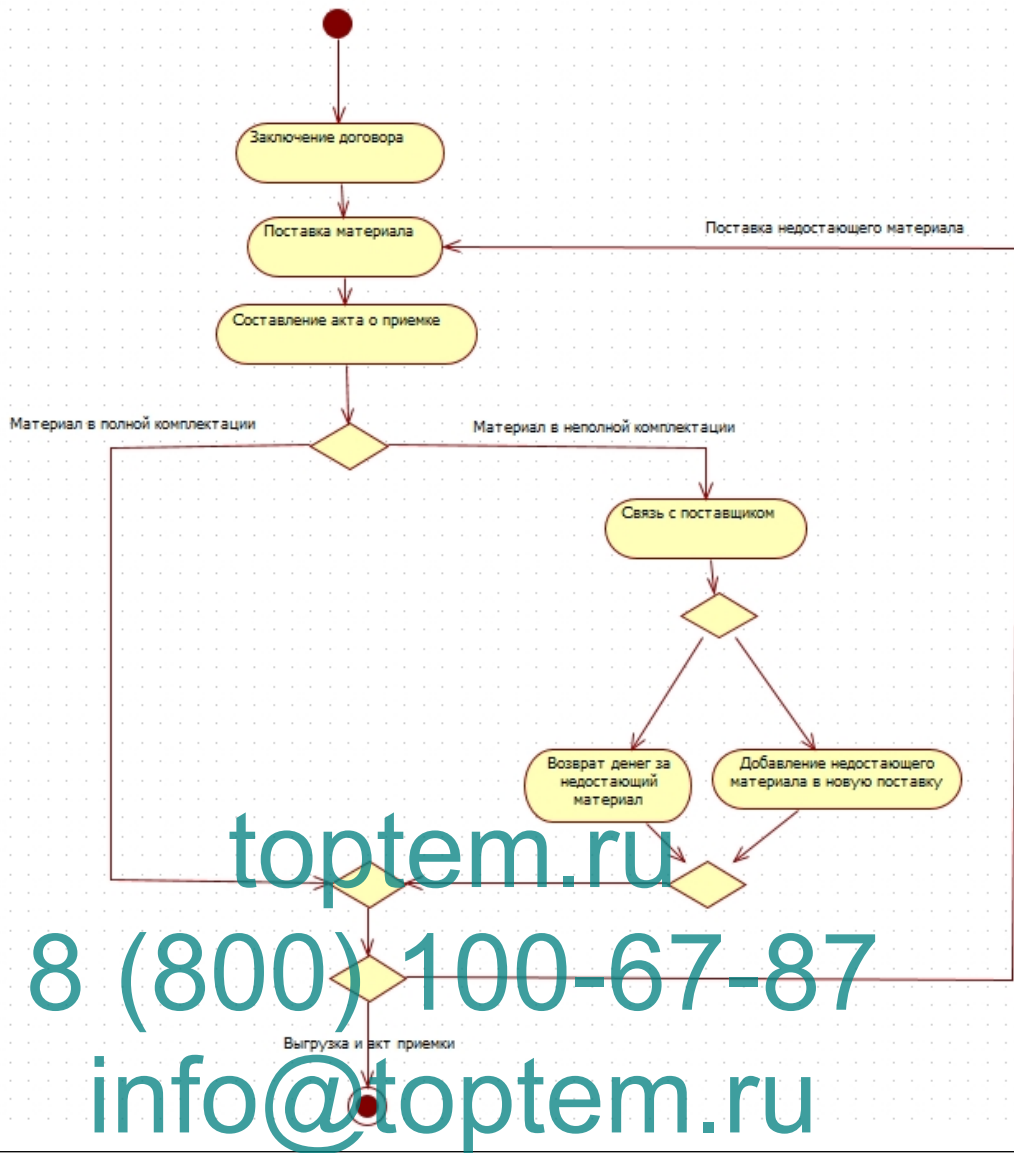


Рисунок А.4 – UML-диаграмма активности

ПРИЛОЖЕНИЕ Б

Руководство пользователя (на английском языке)

Руководство пользователя (на английском языке)

Introduction

Application area

The requirements of this document apply when:

- preliminary complex tests;
- trial operation;
- acceptance tests;
- industrial operation.

Brief Description of Features

The program allows you to automate the process of document management in the enterprise NefteGazoTekhnologii LLC

User experience level

The user of the software must have experience with MS Windows 10, and also have the following knowledge:

- know the relevant subject area;
- know and have skills in working with analytical applications.

List of operational documentation that the user needs to read

- This user manual.
- Purpose and conditions of use.
- Automation of the processes carried out at the enterprise will allow storing information in one database, information into which is entered using a convenient interface.
- Automation will take into account all the features of accounting for a warehouse in a company, that is, develop only what is needed for a given

company.

Preparation for work

Composition and content of the distribution data carrier

The following software is required to operate the software:

- Windows operating system;
- a set of .NET Framework libraries for running a program written on a given software platform;
- Google Chrome;
- MySQL Connector ODBC driver for connecting a remote database to the program.

How to download data and programs

Before starting work with the application "Automated Information System of Warehouse Accounting" at the user's workplace, you must perform the following steps:

Select the executable file with the program and double-click on the icon.

Health Check Procedure

To check the availability of the application "Electronic Document Management" from the user's workplace, you must perform the following steps:

- Open the application, for this you need to double-click on the program icon;
- Make sure that the "Electronic Document Management" application has opened in the window;
- If the application does not start, you should contact the support service.

Description of operations

Functions and tasks performed

The Electronic Document Management application performs the functions and tasks listed in the table below (Table 1).

Table B.1

Functions and tasks of the software product

Functions	Tasks	Description
Launching the program,	User authentication	When you start the program, you

passing authentication		need to enter a login and password to enter the main menu
Search field	Quick database search	When you enter a search query, the result is displayed in a window with a list of data
Working with the main menu	Implementation of secondary program functions	The program menu appears
Adding, editing data	Adding, editing data to the program, updating a remote database	Adding, modifying, deleting data from the database from the application
Export and import of data	Exports and imports selected data from the database	When you click on the button, a system window opens for selecting an imported/exported file
Filling out documents according to the template	Fast filling of documents	The template filling page opens, into which the data is entered, after which it is saved

Description of the data processing workflow operations required to complete the tasks

Below is a description of the user operations for performing each of the tasks.

Task: "User Authentication"

Operation 1: launching the program

Conditions under which the operation is possible:

1. The user's computer is loaded;
2. Application installed.

Preparatory steps:

Run the program.

The main actions in the required sequence:

Double-click the left mouse button on the "Authorization" icon on the desktop.

Final steps:

Not required.

Resources spent on the operation:

12 seconds.

The main actions in the required sequence:

3. In the window that appears, enter your login and password..

Final steps:

After entering the data, select "Login".

Resources spent on the operation:

35 seconds.

Operation 3: data entry

Conditions under which the operation is possible:

The application is running.

The database connection is set up.

Preparatory steps:

Not required.

The main actions in the required sequence:

4. In the window that appears, enter your username and password to enter the program.

Final steps:

After entering the data, select "Login".

Resources spent on the operation:

15 seconds.

toptem.ru

8 (800) 100-67-87

info@toptem.ru

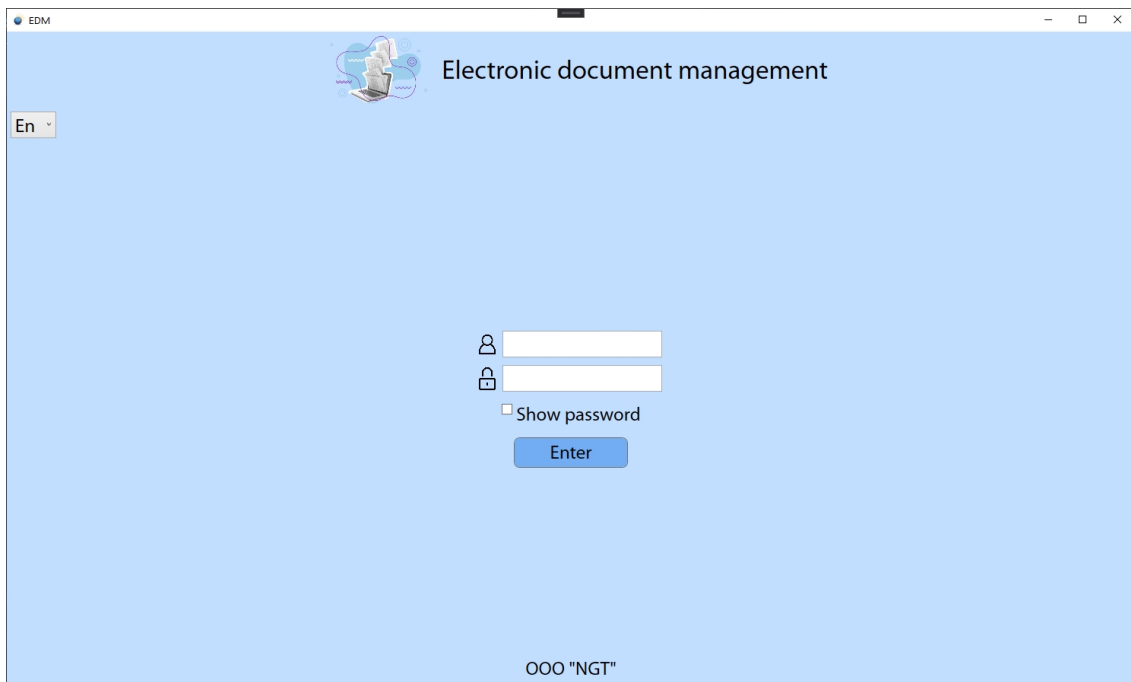


Figure Б.1 - Task "Authorization"

Task: "Search"

Conditions under which the operation is possible:

The application is running.

Preparatory steps:

Not required.

The main actions in the required sequence:

1. Enter a criterion for a quick database search in a string.

Final steps:

Clear the search bar.

Resources spent on the operation:

10-15 seconds.

		All status
<p>***</p> <p>Договор купли-продажи</p> <p>Керамтрейд</p> <p>№ 22</p> <p>Исполнитель: Маслов Пётр Степанович</p>	<p>Заключен</p> <p>2021-02-02</p> <p>Истекает</p> <p>2022-06-02</p>	
<p>***</p> <p>Договор поставки материалов</p> <p>Regamet</p> <p>№ 98</p> <p>Исполнитель: Егорова Виктория Ивановна</p>	<p>Заклучен</p> <p>2021-03-03</p> <p>Истекает</p> <p>2022-10-04</p>	
<p>***</p> <p>Договор поставки</p> <p>КраскиГерметики</p> <p>№ 456</p> <p>Исполнитель: Иванов Илья Романович</p>	<p>Заклучен</p> <p>2021-02-07</p> <p>Истекает</p> <p>2022-05-30</p>	
<p>Договор строительного найма</p> <p>Deluxe</p>	<p>Заклучен</p> <p>2022-05-19</p>	

Add 8 из 8

Figure Б.2 - Search task

Task: "Add contract"

Conditions under which the operation is possible:

The application is running.

Preparatory steps:

Not required.

The main actions in the required sequence:

1. On the main form of the System, select "Agreement";
2. Fill in the fields in the window that appears.

Final steps:

Click the add button.

Resources spent on the operation:

Depends on the amount of data entered.

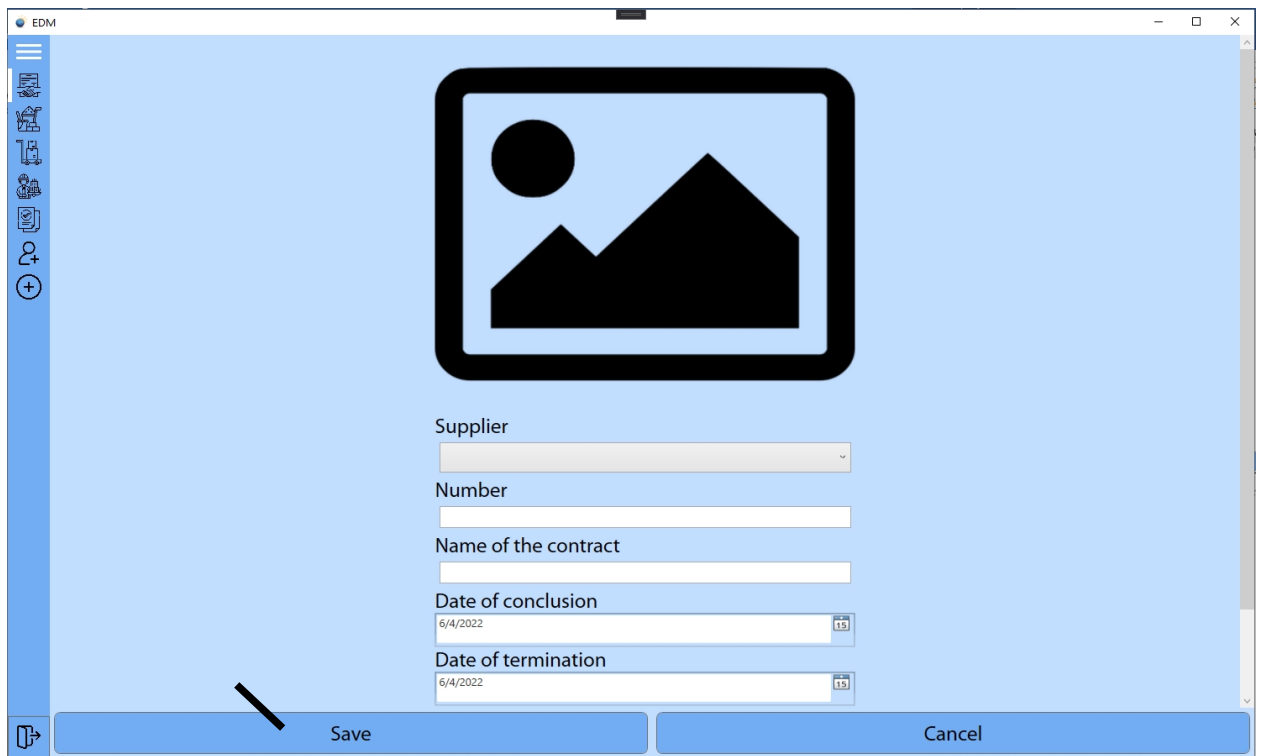


Figure B.3 - Task "Add contract"

Task: "Edit product"

Conditions under which the operation is possible:

The application is running.

Preparatory steps:

Not required.

The main actions in the required sequence:

1. On the main form of the System, select "Agreement";
2. Fill in the fields in the window that appears.

Final steps:

Click the change button.

Resources spent on the operation:

Depends on the amount of data entered.

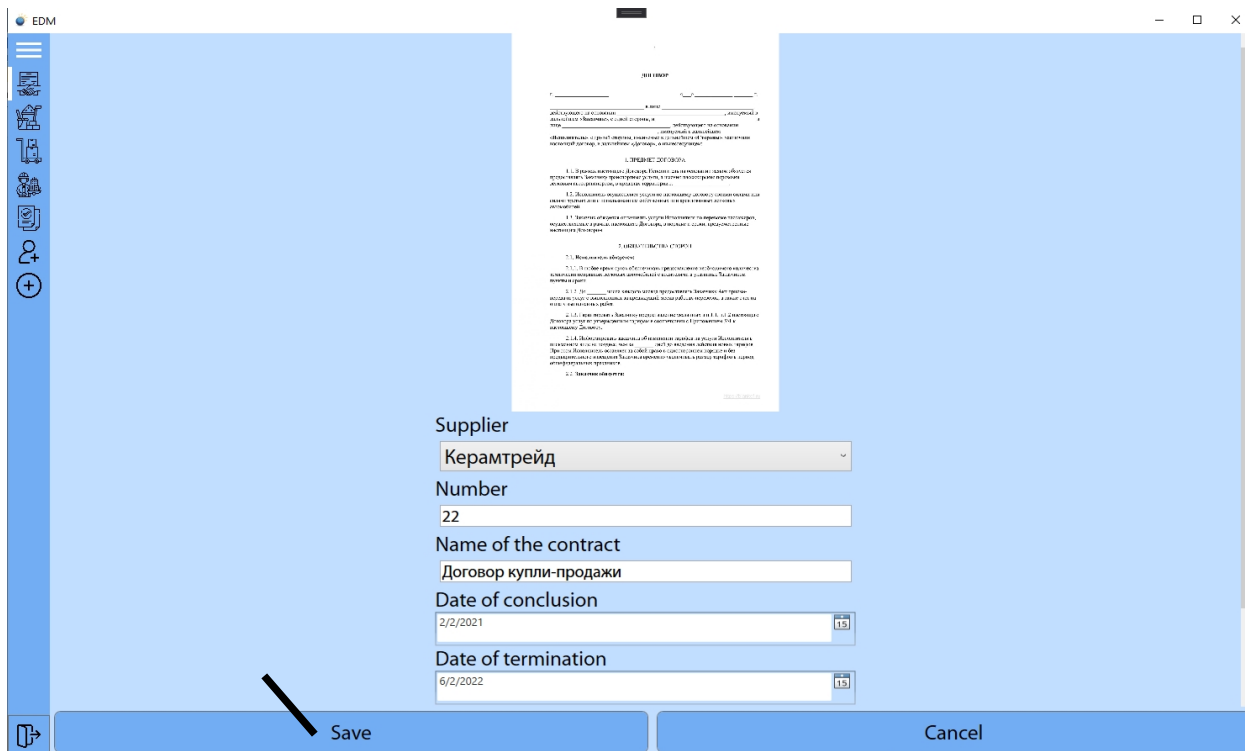


Figure Б.4 - The task "Edit the contract"

Task: "Delete the contract from the database"

Conditions under which the operation is possible:

The application is running.

Preparatory steps:

Not required.

The main actions in the required sequence:

1. Select a contract from the database...

Final steps:

Click on the delete button...

Resources spent on the operation:

Depends on the amount of data entered.

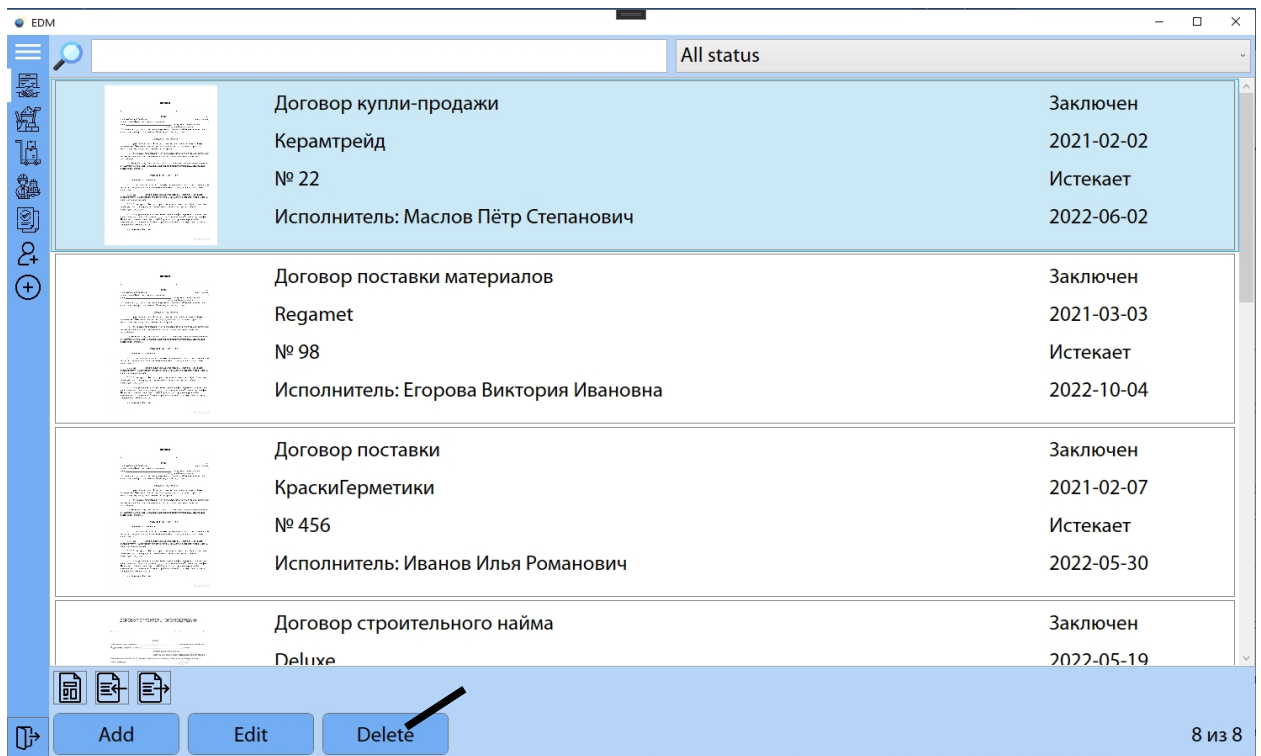


Figure Б.5 - The task "Delete the contract from the database"

toptem.ru
Emergency situations

In the event of an error in the Electronic Document Management, the user is presented with an error message of a similar plan:

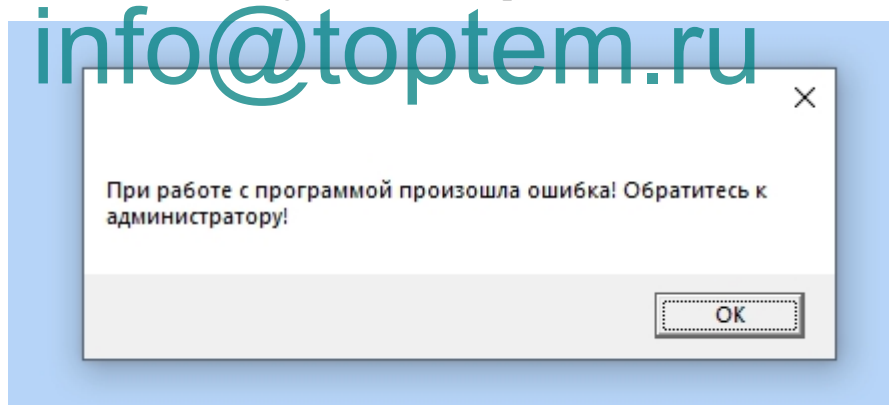


Figure Б.6 - Error

ПРИЛОЖЕНИЕ В

Руководство администратора (на английском языке)

Руководство администратора (на английском языке)

Program name

Electronic document management program. The program is written in the C# programming language in the VisualStudio 2019 development environment.

Purpose of the program

This program is designed to automate the workflow of an enterprise.

Program execution conditions

The program was developed on a Windows 10 computer. Intel Core i5-9400F processor, 64-bit operating system, keyboard and mouse. 256 GB of RAM. Under such conditions, the program will function. The .Net Framework 4.8 installed in the system and access to the MySQL server are required.

The Electronic Document Management program takes up 242 MB (254,201,574 bytes) of disk space.

Program execution

The software product requires a MySQL database located on the server locally or remotely. To set up a connection to the database, you need to set the following parameters in the corresponding page:

- MySQL server address,
- Server Port,
- Username,
- User password,
- DB name.

The image shows a screenshot of a software window titled "EDM". The window has a light blue background and contains a configuration form for a database connection. The form consists of several labeled text input fields stacked vertically. At the bottom of the window, there are two buttons: "Save" on the left and "Cancel" on the right. The fields and their values are as follows:

Field Label	Value
Server	cff31.ru
Port	3306
Database name	ISPr21-33_SoldaevaAD_EDO
Username	ISPr21-33_SoldaevaAD
Password	ISPr21-33_SoldaevaAD

Figure B.1 - Configuring the connection to the base

Beginning of work

To start working with a new database, simply run the application and set the appropriate fields with all the data necessary to connect to the database, after which the application will automatically create the necessary tables and fields in the database. Then, if necessary, you can import a dump of the old database, or create data manually.

ПРИЛОЖЕНИЕ Г

Документированный код программного продукта

Документированный код программного продукта

//Класс модели БД

```
namespace EDODiplom.Database
{
    public partial class EfModel : DbContext
    {
        private static EfModel Instance;
        public static EfModel Init()
        {
            if (Instance == null)
                Instance = new EfModel();
            return Instance;
        }
        public EfModel()
            : base($"server={Properties.Settings.Default.Server};" +
                $"user={Properties.Settings.Default.Username};" +
                $"password={Properties.Settings.Default.Password};" +
                $"database={Properties.Settings.Default.Name}")
        {
        }
        public virtual DbSet<BuildObject> BuildObjects {
            get; set; }
        public virtual DbSet<Client> Clients { get; set; }
        public virtual DbSet<Contract> Contracts { get; set; }
    }
    public virtual DbSet<Material> Materials { get; set; }
}
    public virtual DbSet<Materials_has_Suppliers>
Materials_has_Suppliers { get; set; }
    public virtual DbSet<Materials_has_Supply>
Materials_has_Supply { get; set; }
    public virtual DbSet<MaterialType> MaterialTypes {
get; set; }
    public virtual DbSet<ObjectDocument> ObjectDocuments
{ get; set; }
    public virtual DbSet<Supplier> Suppliers { get; set; }
}
    public virtual DbSet<Supply> Supplies { get; set; }
    public virtual DbSet<User> Users { get; set; }
```

```

    }
}


//Класс BuildObject

[Table("BuildObject")]
public partial class BuildObject :
INotifyPropertyChanged
{
[System.Diagnostics.CodeAnalysis.SuppressMessage("Micro
soft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
public BuildObject()
{
ObjectDocuments = new HashSet<ObjectDocument>();
Supplies = new HashSet<Supply>();
}
public enum StatusBuildObject : sbyte
{
completed, //завершенные
unfinished, //незавершенные
suspended, //приостановленные
}
[Key]
public int ID_BuildObject { get; set; }
[Required]
[StringLength(45)]
public string BuildObjectName { get; set; }
[Required]
private byte[] _BuildObjectPhoto;
public byte[] BuildObjectPhoto
{
get { return _BuildObjectPhoto; }
set
{
_BuildObjectPhoto = value;
PropChange("BuildObjectPhoto");
}
}
public StatusBuildObject BuildStatus { get; set; }

public int Clients_id_Client { get; set; }

public virtual Client Client { get; set; }
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Us
age", "CA2227:CollectionPropertiesShouldBeReadOnly")]
public virtual ICollection<ObjectDocument>
ObjectDocuments { get; set; }

```



8 (800) 100-67-87

info@toptem.ru

```

[System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
public virtual ICollection<Supply> Supplies { get;
set; }

public event PropertyChangedEventHandler
PropertyChanged;
public void PropChange([CallerMemberName] string
PropertyName = "")
{
    if (PropertyChanged != null)
        PropertyChanged.Invoke(this, new
PropertyChangedEventArgs(PropertyName));
}
}

```

//Класс Client

```

[Table("Client")]
public partial class Client
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
    public Client()
    {
        BuildObjects = new HashSet<BuildObject>();
    }

    [Key]
    public int ID_Client { get; set; }

    [Required]
    [StringLength(100)]
    public string FIO { get; set; }

    [Required]
    [StringLength(45)]
    public string INN { get; set; }

    public long Phone { get; set; }

    [Required]
    [StringLength(45)]
    public string Adress { get; set; }

    [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<BuildObject> BuildObjects
{ get; set; }
}

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

```

    }

//Класс Contract

[Table("Contract")]
public partial class Contract : INotifyPropertyChanged
{
    public enum ContractStatus:sbyte
    {
        terminated, //рассторгнутые
        processed, // обрабатываются
        concluded //заклученные
    };

    [Key]
    public int ID_Contract { get; set; }

    [Required]
    [StringLength(45)]
    public string Number { get; set; }

    [Required]
    [StringLength(45)]
    public string Name { get; set; }

    public DateTime Date { get; set; } = DateTime.Now;
    public DateTime DateTermination { get; set; } =
DateTime.Now;

    public int Suppliers_ID_Supplier { get; set; }

    [Required]
    private byte[] _DocumentScan;
    public byte[] DocumentScan {

        get { return _DocumentScan; }
        set
        {
            _DocumentScan = value;
            PropChange("DocumentScan");
        }
    }

    public ContractStatus Status { get; set; }
    public virtual Supplier Supplier { get; set; }

    public event PropertyChangedEventHandler
PropertyChanged;
    public void PropChange([CallerMemberName] string
PropertyName = "")
    {
        if (PropertyChanged != null)

```

```

        PropertyChanged.Invoke(this, new
PropertyChangedEventArgs(PropertyName));
    }
}

//Класс Material

[Table("Material")]
public partial class Material:INotifyPropertyChanged
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
    public Material()
    {
        Materials_has_Suppliers = new
HashSet<Materials_has_Suppliers>();
        Materials_has_Supply = new
HashSet<Materials_has_Supply>();
        MaterialTypes = new HashSet<MaterialType>();
    }

    [Key]
    public int Id { get; set; }

    [Required]
    [StringLength(45)]
    public string Name { get; set; }

    [Required]
    private byte[] _ImageMaterial;
    public byte[] ImageMaterial {
        get { return _ImageMaterial; }
        set
        {
            _ImageMaterial = value;
            PropChange("ImageMaterial");
        }
    }

    [StringLength(45)]
    public string Description { get; set; }
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Micros
oft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual HashSet<Materials_has_Suppliers>
Materials_has_Suppliers { get; set; }

    [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Materials_has_Supply>
Materials_has_Supply { get; set; }
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsof

```

toptem.ru
8 (800) 100-67-87
info@toptem.ru


```

t.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<MaterialType>
MaterialTypes { get; set; }

    public event PropertyChangedEventHandler
PropertyChanged;
    public void PropChange([CallerMemberName] string
PropertyName = "")
    {
        if (PropertyChanged != null)
            PropertyChanged.Invoke(this, new
PropertyChangedEventArgs(PropertyName));
    }
}

```

//Класс Materials_has_Suppliers

```

[Table("Materials_has_Suppliers")]
public partial class Materials_has_Suppliers
{
    [Key]
    [Column(Order = 0)]
    [DatabaseGenerated(DatabaseGeneratedOption.None)]
    public int Materials_ID_Material { get; set; }

    [Key]
    [Column(Order = 1)]
    [DatabaseGenerated(DatabaseGeneratedOption.None)]
    public int Suppliers_ID_Supplier { get; set; }

    public decimal Material_Price { get; set; }

    public virtual Material Material { get; set; }

    public virtual Supplier Supplier { get; set; }
}

```

//Класс Materials_has_Supply

```

[Table("Materials_has_Supply")]
public partial class Materials_has_Supply
{
    [Key]
    [Column(Order = 0)]
    [DatabaseGenerated(DatabaseGeneratedOption.None)]
    public int Materials_ID_Material { get; set; }

    [Key]
    [Column(Order = 1)]
    [DatabaseGenerated(DatabaseGeneratedOption.None)]
    public int Supply_ID_Supplier { get; set; }
}

```

```

        public int Material_Count { get; set; }

        public decimal Material_Price { get; set; }

        public virtual Material Material { get; set; }

        public virtual Supply Supply { get; set; }
    }

```

//Класс MaterialType

```

[Table("MaterialType")]
public partial class MaterialType
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
    public MaterialType()
    {
        Materials = new HashSet<Material>();
    }

    [Key]
    public int ID_MaterialType { get; set; }

    [Required]
    [StringLength(45)]
    public string MaterialTypeName { get; set; }

    [StringLength(2071741823)]
    public string MaterialTypeDescription { get; set; }

    [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Material> Materials {
get; set; }
}

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

//Класс ObjectDocument

```

[Table("ObjectDocument")]
public partial class ObjectDocument :
INotifyPropertyChanged
{
    public enum ObjectDocumentStatus : sbyte
    {
        processed, // обрабатываются
    }
}

```

```

        raw //необработанные
    }
    [Key]
    public int ID_ObjectDocument { get; set; }

    [Required]
    [StringLength(45)]
    public string ObjectName { get; set; }

    [Required]
    private byte[] _ObjectScan;
    public byte[] ObjectScan
    {
        get { return _ObjectScan; }
        set
        {
            _ObjectScan = value;
            PropChange();
        }
    }
    [Required]
    public string DocumentNumber { get; set; }

    [Required]
    public ObjectDocumentStatus DocumentStatus { get;
set; }
    public int BuildObject_ID_BuildObject { get; set; }

    public virtual BuildObject BuildObject { get; set; }
    public event PropertyChangedEventHandler
PropertyChanged;
    public void PropChange([CallerMemberName] string
PropertyName = "")
    {
        if (PropertyChanged != null)
            PropertyChanged.Invoke(this, new
PropertyChangedEventArgs(PropertyName));
    }
}

//Класс Supplier

[Table("Supplier")]
public partial class Supplier : INotifyPropertyChanged
{
    [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
    public Supplier()
    {

```

```

        Contracts = new HashSet<Contract>();
        Materials_has_Suppliers = new
HashSet<Materials_has_Suppliers>();
        Supplies = new HashSet<Supply>();
    }

    [Key]
    public int ID_Supplier { get; set; }

    [Required]
    [StringLength(45)]
    public string Title { get; set; }

    [Required]
    [StringLength(45)]
    public string Adress { get; set; }

    public double Phone { get; set; }

    [Required]
    private byte[] _Photo;
    public byte[] Photo
    {
        get { return _Photo; }
        set
        {
            _Photo = value;
            PropChange();
        }
    }

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Contract> Contracts {
get; set; }
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Materials_has_Suppliers>
Materials_has_Suppliers { get; set; }
[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
    public virtual ICollection<Supply> Supplies { get;
set; }

    public event PropertyChangedEventHandler
PropertyChanged;
    public void PropChange([CallerMemberName] string
PropertyName = "")
    {
        if (PropertyChanged != null)
            PropertyChanged.Invoke(this, new
PropertyChangedEventArgs(PropertyName));
    }

```

toptem.ru
8 (800) 100-67-87
info@toptem.ru

```

    }

    //Класс Supply
    [Table("Supply")]
    public partial class Supply
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
        public Supply()
        {
            Materials_has_Supply = new
HashSet<Materials_has_Supply>();
        }

        [Key]
        public int ID_Supply { get; set; }

        public int Suppliers_ID_Supplier { get; set; }

        public int BuildObject_ID_BuildObject { get; set; }

        public DateTime Date { get; set; }

        public virtual BuildObject BuildObject { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<Materials_has_Supply>
Materials_has_Supply { get; set; }

        public virtual Supplier Supplier { get; set; }
    }

```

//Класс User

```

    [Table("User")]
    public partial class User : INotifyPropertyChanged
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Mi
crosoft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
        public User()
        {
        }

        [Key]
        public int idUser { get; set; }

        [Required]
        [StringLength(45)]
        public string Password { get; set; }
    }

```

```

        [Required]
        [StringLength(45)]
        public string Login { get; set; }

        [Required]
        [StringLength(60)]
        public string FIO { get; set; }

        public int Role { get; set; }
        [Required]
        private byte[] _UserPhoto;
        public byte[] UserPhoto
        {
            get { return _UserPhoto; }
            set
            {
                _UserPhoto = value;
                PropChange();
            }
        }
        public event PropertyChangedEventHandler
PropertyChang
PropertyChang
        public void PropChange([CallerMemberName] string
PropertyName = "")
        {
            if (PropertyChanged != null)
                PropertyChanged.Invoke(this, new
PropertyChang
PropertyChang
                }
        }
    }
}
//Класс страницы «Авторизация»

namespace EDODiplom.Pages
{
    /// <summary>
    /// Interaction logic for AutorizationPage.xaml
    /// </summary>
    public partial class AutorizationPage : Page
    {
        public string password = "";
        public AutorizationPage()
        {
            InitializeComponent();
            TbLogin.DataContext = this;
            TbPass.DataContext = this;
            if (Properties.Settings.Default.Localisation ==
"RU")
                CbLocale.SelectedIndex = 0;
            else
                CbLocale.SelectedIndex = 1;
        }
    }
}

```

```

private void Login() {
    //Тернарное выражение
    string password = IsPassShow.IsChecked.Value ?
    TbPass.Text : PbPass.Password;
    if (AuthClass.Auth(TbLogin.Text, password))
    {
        NavigationService.Navigate(new MenuPage());
    }
    else
        MessageBox.Show(Localisation.MessageLocalisa
tion.MessageAuth);
}

private void BtEntryClick(object sender,
RoutedEventArgs e)
{
    Login();
}

private void CBPassCheck(object sender,
RoutedEventArgs e)
{
    TbPass.Text = PbPass.Password;
}

private void CBPassUncheck(object sender,
RoutedEventArgs e)
{
    PbPass.Password = TbPass.Text;
}

private void KeyBtDown(object sender, KeyEventArgs
e)
{
    if (e.Key == Key.Enter)
        if (!TbLogin.IsFocused)
            Login();
}

private void BtLoginEnter(object sender,
KeyEventArgs e)
{
    if (e.Key == Key.Enter)
        if (IsPassShow.IsChecked.Value)
            TbPass.Focus();
        else
            PbPass.Focus();
}

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

```

        private void BtSetting(object sender,
RoutedEventArgs e)
        {
            NavigationService.Navigate(new SettingPage());
        }

        private void UpdateLocale()
        {
            ResourceDictionary resourceDictionary;
            if (Properties.Settings.Default.Localisation ==
"RU")
            {
                resourceDictionary = new ResourceDictionary
{ Source = new Uri("Localisation/Localisation-Ru.xaml",
UriKind.Relative) };
            }
            else
            {
                resourceDictionary = new ResourceDictionary
{ Source = new Uri("Localisation/Localisation-En.xaml",
UriKind.Relative) };
            }
            Thread.CurrentThread.CurrentCulture = new
CultureInfo(Properties.Settings.Default.Localisation);
            Thread.CurrentThread.CurrentUICulture = new
CultureInfo(Properties.Settings.Default.Localisation);
            Application.Current.Resources.MergedDictionaries
.Clear();
            Application.Current.Resources.MergedDictionaries
.Add(resourceDictionary);
        }

        private void CbLocaleChange(object sender,
SelectionChangedEventArgs e)
        {
            ComboBoxItem localItem = CbLocale.SelectedItem
as ComboBoxItem;
            Properties.Settings.Default.Localisation =
localItem.Tag as string;
            Properties.Settings.Default.Save();
            UpdateLocale();
        }

        private void BtConnection(object sender,
RoutedEventArgs e)
        {
            NavigationService.Navigate(new Connection2BD());
        }
    }
}

```

//Класс страницы «Строительный объект»


```

namespace EDODiplom.Pages
{
    /// <summary>
    /// Interaction logic for BuildObjectListPage.xaml
    /// </summary>
    public partial class BuildObjectListPage : Page
    {
        public BuildObjectListPage()
        {
            InitializeComponent();
            UpdateData();
            CbFilterBuild.SelectedIndex = 0;
        }

        private void UpdateData()
        {
            IEnumerable<BuildObject> buildObjects =
EfModel.Init().BuildObjects.Include(b=>b.ObjectDocuments).Include
(b=>b.Client).
                Where(o =>
o.BuildObjectName.Contains(TbSearch.Text));
            if (CbFilterBuild.SelectedIndex > 0)
                buildObjects = buildObjects.Where(b =>
b.BuildStatus == (StatusBuildObject) (CbFilterBuild.SelectedIndex
- 1));
            LvBuildObjects.ItemsSource =
buildObjects.ToList();
            if (LvBuildObjects.Items.Count == 0)
            {
                TbNotFound.Visibility = Visibility.Visible;
            }
            else
            {
                TbNotFound.Visibility = Visibility.Hidden;
            }
            TbResultCount.Text = LvBuildObjects.Items.Count
+ " из " + EfModel.Init().BuildObjects.Count();
        }

        private void BtBuildObjectAddClick(object sender,
RoutedEventArgs e)
        {
            NavigationService.Navigate(new
BuildObjectEditPage(new BuildObject()));
        }

        private void BtBuildObjectEditClick(object sender,
RoutedEventArgs e)

```

toptem.ru
8 (800) 100-67-87
info@toptem.ru

```

        {
            if (LvBuildObjects.SelectedItems.Count > 0)
            {
                BuildObject buildObject =
LvBuildObjects.SelectedItem as BuildObject;
                NavigationService.Navigate(new
BuildObjectEditPage(buildObject));
            }
        }

        private void BtBuildObjectDelClick(object sender,
RoutedEventArgs e)
        {
            foreach (BuildObject buildObject in
LvBuildObjects.SelectedItems)
            {
                if (MessageBox.Show("Вы точно хотите удалить
" + buildObject.BuildObjectName + "?", "Удалить объект",
MessageBoxButton.YesNo) == MessageBoxResult.Yes)
                {
                    foreach (Supply sup in
buildObject.Supplies.ToList())
                    {
                        foreach (Materials_has_Supply
materials_has_Supply in sup.Materials_has_Supply.ToList())
                        {
                            EfModel.Init().Materials_has_Sup
ply.Remove(materials_has_Supply);
                        }
                    }
                    EfModel.Init().Supplies.Remove(sup);
                }
                EfModel.Init().SaveChanges();
                EfModel.Init().BuildObjects.Remove(build
Object);
                EfModel.Init().SaveChanges();
            }
        }
        UpdateData();
    }

    private void TbSearchChange(object sender,
TextChangedEventArgs e)
    {
        UpdateData();
    }

    private void BtExportClick(object sender,

```

```

RoutedEventArgs e)
    {
        SaveFileDialog saveFile = new SaveFileDialog {
Filter = "Excel Files|*.xlsx" };
        if (saveFile.ShowDialog() == true)
        {
            ExportClass.BuildObjectExport(saveFile.FileName);
        }
        else
            MessageBox.Show(Localisation.MessageLocalisation.MessageExport);
    }

    private void BtImportBuildObject(object sender,
RoutedEventArgs e)
    {
        OpenFileDialog openFileDialog = new
OpenFileDialog { Filter = "Excel Files|*.xlsx" };
        if (openFileDialog.ShowDialog() == true)
        {
            ImportClass.ImportBuildObject(openFileDialog
.FileName);
        }
        else
            MessageBox.Show(Localisation.MessageLocalisation.MessageImport);
        UpdateData();
    }

    private void PageVisChange(object sender,
DependencyPropertyChangedEventArgs e)
    {
        UpdateData();
    }

    private void FilterBuildChange(object sender,
SelectionChangedEventArgs e)
    {
        UpdateData();
    }
}

//Класс «Договоры»

namespace EDODiplom.Pages
{
    /// <summary>
    /// Interaction logic for ContractListPage.xaml
    /// </summary>
    public partial class ContractListPage : Page

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

```

{
    public ContractListPage()
    {
        InitializeComponent();
        UpdateData();

        CbFilter.SelectedIndex = 0;
    }
    private void UpdateData()
    {
        IEnumerable<Contract> contracts =
EfModel.Init().Contracts
        .Include(c => c.Supplier).Where(c =>
c.Name.Contains(TbSearch.Text));

        if (CbFilter.SelectedIndex > 0)
            contracts = contracts.Where(c => c.Status ==
(ContractStatus) (CbFilter.SelectedIndex - 1));

        LvContracts.ItemsSource = contracts.ToList();

        if (LvContracts.Items.Count == 0)
        {
            TbNotFound.Visibility = Visibility.Visible;
        }
        else
            TbNotFound.Visibility = Visibility.Hidden;
        TbResultCount.Text = LvContracts.Items.Count + "
из " + EfModel.Init().Contracts.Count();
    }

    private void TbSearchChange(object sender,
TextChangedEventArgs e)
    {
        UpdateData();
    }

    private void BtContractAddClick(object sender,
RoutedEventArgs e)
    {
        NavigationService.Navigate(new
ContractEditPage(new Contract()));
    }

    private void BtContractEditClick(object sender,
RoutedEventArgs e)
    {
        if (LvContracts.SelectedItems.Count > 0)
        {
            Contract contract = LvContracts.SelectedItem

```

toptem.ru
8 (800) 100-67-87
info@toptem.ru

```

as Contract;
        NavigationService.Navigate(new
ContractEditPage(contract));
    }

    }

    private void BtContractDelClick(object sender,
RoutedEventArgs e)
    {
        foreach (Contract contract in
LvContracts.SelectedItems)
        {
            if (LvContracts.SelectedItems.Count > 0)
            {
                if (MessageBox.Show("Вы точно хотите
удалить " + contract.Name + "?", "Удалить договор",
MessageBoxButton.YesNo) == MessageBoxResult.Yes)
                {
                    EfModel.Init().SaveChanges();
                    EfModel.Init().Contracts.Remove(cont
ract);
                    EfModel.Init().SaveChanges();
                }
            }
        }
        UpdateData();
    }

    private void PageVisChange(object sender,
DependencyPropertyChangedEventArgs e)
    {
        UpdateData();
    }

    private void BtTemplateClick(object sender,
RoutedEventArgs e)
    {
        NavigationService.Navigate(new
TemplateDocxPage());
    }

    private void BtExportContract(object sender,
RoutedEventArgs e)
    {
        SaveFileDialog saveFile = new SaveFileDialog {
Filter = "Excel Files|*.xlsx" };
        if (saveFile.ShowDialog() == true)
        {
            ExportClass.ContractExport(saveFile.FileName
);
        }
    }
}

```

toptem.ru
8 (800) 100-67-87
info@toptem.ru

```

    }
    else
        MessageBox.Show("Файл не выбран!");
}

private void BtImportContract(object sender,
RoutedEventArgs e)
{
    OpenFileDialog openFileDialog = new
OpenFileDialog { Filter = "Excel Files|*.xlsx" };
    if (openFileDialog.ShowDialog() == true)
    {
        ImportClass.ImportContract(openFileDialog.Fi
leName);

        UpdateData();
    }
    else
        MessageBox.Show("Файл не выбран!");
}

private void FilterChange(object sender,
SelectionChangedEventArgs e)
{
    UpdateData();
}
}

//Класс страницы «Материалы»
namespace EDODiplom.Pages
{
    /// <summary>
    /// Interaction logic for MaterialListPage.xaml
    /// </summary>
    public partial class MaterialListPage : Page
    {
        public MaterialListPage()
        {
            InitializeComponent();
            UpdateData();
            List<MaterialType> materialTypes =
EfModel.Init().MaterialTypes.ToList();
            materialTypes.Insert(0, new MaterialType {
MaterialTypeName = "Все типы" });
            CbFilter.ItemsSource = materialTypes;
            CbFilter.SelectedIndex = 0;

            CbSort.Items.Add("Наименование");
            CbSort.Items.Add("▲ Цена");
            CbSort.Items.Add("▼ Цена");

```

```

        CbSort.SelectedIndex = 0;
    }
    //Метод обновления данных
    private void UpdateData()
    {
        IEnumerable<Material> materials =
EfModel.Init().Materials
        .Include(m => m.MaterialTypes).Include(m =>
m.Materials_has_Suppliers)
        .Where(m => m.Name.Contains(TbSearch.Text));

        if (CbFilter.SelectedIndex > 0)
            materials = materials.Where(m =>
m.MaterialTypes.Select(mt => mt.ID_MaterialType).Contains(
(CbFilter.SelectedItem as
MaterialType).ID_MaterialType));

        switch (CbSort.SelectedIndex)
        {
            case 0:
                materials = materials.OrderBy(m =>
m.Name);
            case 1:
                materials =
materials.OrderByDescending(m => m.Materials_has_Suppliers.Count
< 1 ? 0 : m.Materials_has_Suppliers.Min(ms =>
ms.Material_Price));
            case 2:
                materials = materials.OrderBy(m =>
m.Materials_has_Suppliers.Count < 1 ? 0 :
m.Materials_has_Suppliers.Max(ms => ms.Material_Price));
            break;
        }

        LvMaterials.ItemsSource = materials.ToList();

        if (LvMaterials.Items.Count == 0)
        {
            TbNotFound.Visibility = Visibility.Visible;
        }
        else
        {
            TbNotFound.Visibility = Visibility.Hidden;
        }
        TbResultCount.Text = LvMaterials.Items.Count + "
из " + EfModel.Init().Materials.Count();
    }

    private void TbSearchChange(object sender,

```

toptem.ru

8 (800) 100-67-87

info@toptem.ru

```

TextChangedEventArgs e)
    {
        UpdateData();
    }

    private void SortChange(object sender,
SelectionChangedEventArgs e)
    {
        UpdateData();
    }

    private void FilterChange(object sender,
SelectionChangedEventArgs e)
    {
        UpdateData();
    }

    private void BtMaterialAddClick(object sender,
RoutedEventArgs e)
    {
        NavigationService.Navigate(new
MaterialEditPage(new Material()));
    }

    private void BtMaterialDelClick(object sender,
RoutedEventArgs e)
    {
        foreach (Material material in
LvMaterials.SelectedItems)
        {
            if (MessageBox.Show("Вы точно хотите удалить
" + material.Name + "?", "Удалить материал",
MessageBoxButton.YesNo) == MessageBoxResult.Yes)
            {
                EfModel.Init().SaveChanges();
                EfModel.Init().Materials.Remove(material
);
                EfModel.Init().SaveChanges();
            }
        }
        UpdateData();
    }

    private void BtMaterialEditClick(object sender,
RoutedEventArgs e)
    {
        if (LvMaterials.SelectedItems.Count > 0)
        {
            Material material = LvMaterials.SelectedItem
as Material;

```

toptem.ru

8 (800) 100-67-87

info@toptem.ru


```

        NavigationService.Navigate(new
MaterialEditPage(material));
    }

    }

    private void BtExportMaterial(object sender,
RoutedEventArgs e)
    {
        SaveFileDialog saveFile = new SaveFileDialog {
Filter = "Excel Files|*.xlsx" };
        if (saveFile.ShowDialog() == true)
        {
            ExportClass.MaterialExport(saveFile.FileName
);
        }
        else
            MessageBox.Show("Файл не выбран!");
    }

    private void BtImportMaterial(object sender,
RoutedEventArgs e)
    {
        OpenFileDialog openFileDialog = new
OpenFileDialog { Filter = "Excel Files|*.xlsx" };
        if (openFileDialog.ShowDialog() == true)
            ImportClass.ImportMaterial(openFileDialog.Fi
leName);
        UpdateData();
    }
    else
        MessageBox.Show("Файл не выбран!");
    }
}
}
}

```

//Класс страницы «Меню»

```

namespace EDODiplom.Pages
{
    /// <summary>
    /// Interaction logic for MenuWindow.xaml
    /// </summary>
    public partial class MenuPage : Page
    {
        public MenuPage()
        {
            InitializeComponent();
            if(AuthClass.User.Role == 2)

```

```

        {
            MIBuilds.Visibility = Visibility.Visible;
            MIDocs.Visibility = Visibility.Visible;
            MIUser.Visibility = Visibility.Visible;
            MIConnection.Visibility =
Visibility.Visible;
        }
    }
    private void ItemMaterials(object sender,
RoutedEventArgs e)
    {
        FRnav.Navigate(new MaterialListPage());
    }

    private void ItemContracts(object sender,
RoutedEventArgs e)
    {
        FRnav.Navigate(new ContractListPage());
    }

    private void ItemSuppliers(object sender,
RoutedEventArgs e)
    {
        FRnav.Navigate(new SupplierListPage());
    }

    private void ItemBuildObject(object sender,
RoutedEventArgs e)
    {
        FRnav.Navigate(new BuildObjectListPage());
    }

    private void ItemDocumentObject(object sender,
RoutedEventArgs e)
    {
        FRnav.Navigate(new ObjectDocumentPage());
    }

    private void BtExit(object sender, RoutedEventArgs
e)
    {
        NavigationService.Navigate(new
AuthorizationPage());
    }

    private void ItemUser(object sender, RoutedEventArgs
e)
    {
        FRnav.NavigationService.Navigate(new
SettingPage());
    }

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

```

        private void ItemConnection(object sender,
RoutedEventArgs e)
        {
            NavigationService.Navigate(new Connection2BD());
        }
    }
}

```

//Класс страницы «Строительная документация»

```

namespace EDODiplom.Pages
{
    /// <summary>
    /// Interaction logic for ObjectDocumentPage.xaml
    /// </summary>
    public partial class ObjectDocumentPage : Page
    {
        public ObjectDocumentPage ()
        {
            InitializeComponent();
            UpdateData();
            CbFilterDocument.Items.Add("Все статусы");
            CbFilterDocument.Items.Add("Обработан");
            CbFilterDocument.Items.Add("Необработан");
            CbFilterDocument.SelectedIndex = 0;
        }
        private void UpdateData()
        {
            IEnumerable<ObjectDocument> objectDocuments =
EfModel.Init().ObjectDocuments.Include(o => o.BuildObject).
                Where(o =>
o.ObjectName.Contains(TbSearch.Text));

            if (CbFilterDocument.SelectedIndex > 0)
                objectDocuments = objectDocuments.Where(o =>
o.DocumentStatus ==
(ObjectDocumentStatus) (CbFilterDocument.SelectedIndex - 1));
            LvObjectDocument.ItemsSource =
objectDocuments.ToList();

            if (LvObjectDocument.Items.Count == 0)
            {
                TbNotFound.Visibility = Visibility.Visible;
            }
            else
            {
                TbNotFound.Visibility = Visibility.Hidden;
            }
            TbResultCount.Text =

```

```

LvObjectDocument.Items.Count + " из " +
EfModel.Init().ObjectDocuments.Count();
    }

    private void TbSearchChange(object sender,
TextChangedEventArgs e)
    {
        UpdateData();
    }

    private void BtObjectDocumentAddClick(object sender,
RoutedEventArgs e)
    {
        NavigationService.Navigate(new
ObjectDocumentEditPage(new ObjectDocument()));
    }

    private void BtObjectDocumentEditClick(object
sender, RoutedEventArgs e)
    {
        if (LvObjectDocument.SelectedItems.Count > 0)
        {
            ObjectDocument objectDocument =
LvObjectDocument.SelectedItem as ObjectDocument;
            NavigationService.Navigate(new
ObjectDocumentEditPage(objectDocument));
        }
    }

    private void BtObjectDocumentDelClick(object sender,
RoutedEventArgs e)
    {
        foreach (ObjectDocument objectDocument in
LvObjectDocument.SelectedItems)
        {
            if (LvObjectDocument.SelectedItems.Count >
0)
            {
                if (MessageBox.Show("Вы точно хотите
удалить " + objectDocument.ObjectName + "?", "Удалить документ",
MessageBoxButton.YesNo) == MessageBoxResult.Yes)
                {
                    EfModel.Init().SaveChanges();
                    EfModel.Init().ObjectDocuments.Remove
(objectDocument);
                    EfModel.Init().SaveChanges();
                }
            }
        }
        UpdateData();
    }
}

```

toptem.ru
8 (800) 100-67-87

info@toptem.ru

```

        private void BtExportObjectDocument(object sender,
RoutedEventArgs e)
        {
            SaveFileDialog saveFile = new SaveFileDialog {
Filter = "Excel Files|*.xlsx" };
            if (saveFile.ShowDialog() == true)
            {
                ExportClass.ObjectDocumentExport(saveFile.Fi
leName);
            }
            else
                MessageBox.Show("Файл не выбран!");
        }

        private void BtImportObjectDocument(object sender,
RoutedEventArgs e)
        {
            OpenFileDialog openFileDialog = new
OpenFileDialog { Filter = "Excel Files|*.xlsx" };
            if (openFileDialog.ShowDialog() == true)
            {
                ImportClass.ImportObjectDocument(openFileDia
log.FileName);
                UpdateData();
            }
            else
                MessageBox.Show("Файл не выбран!");
        }

        private void FilterChangeDocument(object sender,
SelectionChangedEventArgs e)
        {
            UpdateData();
        }

        private void BtTemplateDocumentClick(object sender,
RoutedEventArgs e)
        {
        }
    }
}

```

//Класс страницы «Поставщики»

```

namespace EDODiplom.Pages
{
    /// <summary>
    /// Interaction logic for SupplierListPage.xaml
    /// </summary>
    public partial class SupplierListPage : Page

```

```

    {
        public SupplierListPage()
        {
            InitializeComponent();
            UpdateData();
        }

        private void UpdateData()
        {
            IEnumerable<Supplier> suppliers =
EfModel.Init().Suppliers.
                Where(s =>
s.Title.Contains(TbSearchSupplier.Text));
            LvSuppliers.ItemsSource = suppliers.ToList();

            if (LvSuppliers.Items.Count == 0)
            {
                TbNotFound.Visibility = Visibility.Visible;
            }
            else
            {
                TbNotFound.Visibility = Visibility.Hidden;
            }
            TbResultCount.Text = LvSuppliers.Items.Count + "
из " + EfModel.Init().Suppliers.Count();
        }

        private void TbSearchSupplierChanged(object sender,
TextChangedEventArgs e)
        {
            UpdateData();
        }

        private void BtSupplierAddClick(object sender,
RoutedEventArgs e)
        {
            NavigationService.Navigate(new
SupplierEditPage(new Supplier()));
        }

        private void BtSupplierEditClick(object sender,
RoutedEventArgs e)
        {
            if (LvSuppliers.SelectedItems.Count > 0)
            {
                Supplier supplier = LvSuppliers.SelectedItem
as Supplier;
                NavigationService.Navigate(new
SupplierEditPage(supplier));
            }
        }
    }

```

toptem.ru
8 (800) 100-67-87
info@toptem.ru

```

        private void BtSupplierDelClick(object sender,
RoutedEventArgs e)
        {
            foreach (Supplier supplier in
LvSuppliers.SelectedItems)
            {
                if (MessageBox.Show("Вы точно хотите
удалить" + supplier.Title + "?", "Удалить поставщика",
MessageBoxButton.YesNo) == MessageBoxResult.Yes)
                {
                    EfModel.Init().SaveChanges();
                    EfModel.Init().Suppliers.Remove(supplier
);
                    EfModel.Init().SaveChanges();
                }
            }
            UpdateData();
        }

        private void BtExportSupplier(object sender,
RoutedEventArgs e)
        {
            SaveFileDialog saveFile = new SaveFileDialog {
Filter = "Excel Files|*.xlsx" };
            if (saveFile.ShowDialog() == true)
            {
                ExportClass.SupplierExport(saveFile.FileName
);
            }
            else
            {
                MessageBox.Show("Файл не выбран!");
            }
        }

        private void BtImportSupplier(object sender,
RoutedEventArgs e)
        {
            OpenFileDialog openFileDialog = new
OpenFileDialog { Filter = "Excel Files|*.xlsx" };
            if (openFileDialog.ShowDialog() == true)
            {
                ImportClass.ImportSupplier(openFileDialog.Fi
leName);
                UpdateData();
            }
            else
            {
                MessageBox.Show("Файл не выбран!");
            }
        }

        private void PageVisChange(object sender,
DependencyPropertyChangedEventArgs e)

```

toptem.ru
8 (800) 100-67-87
info@toptem.ru

```

        {
            UpdateData();
        }
    }
}

//Класс страницы «Заполнения шаблона»

namespace EDODiplom.Pages
{
    /// <summary>
    /// Interaction logic for TemplateDocxPage.xaml
    /// </summary>
    public partial class TemplateDocxPage : Page
    {
        TemplateItem templateItem = new TemplateItem();
        public TemplateDocxPage()
        {
            InitializeComponent();
            DataContext = templateItem;
        }

        private void BtFill(object sender, RoutedEventArgs
e)
        {
            SaveFileDialog saveFile = new SaveFileDialog {
Filter = "Docx Files|.docx"},
            if (saveFile.ShowDialog() == true)
            {
                TemplateDocx.TemplateParse(saveFile.FileName
, templateItem);
            }
            else
                MessageBox.Show("Не все поля заполнены!");
        }

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

//Класс авторизации

namespace EDODiplom.Classes
{
    static class AuthClass

```



```

    {
        public static User User { get; private set; }
        public static bool Auth(string login, string
password)
    {
        User = EfModel.Init().Users.FirstOrDefault(u =>
u.Login == login && u.Password == password);
        return User != null;
    }
    }
}

```

//Класс экспорта

```

namespace EDODiplom.Classes
{
    static class ExportClass
    {
        public static void BuildObjectExport(string
filename)
        {
            XLWorkbook workbook = new XLWorkbook();
            IXLWorksheet worksheet =
workbook.AddWorksheet("BuildObject");
            worksheet.Column(1).Width = 50;
            worksheet.Column(2).Width = 50;
            worksheet.Column(3).Width = 50;
            worksheet.Column(4).Width = 50;
            worksheet.Column(5).Width = 50;
            IXLRow row0 = worksheet.Row(1);
            row0.Cell(1).Value = "ИД Объекта";
            row0.Cell(2).Value = "Наименование";
            row0.Cell(3).Value = "Фотография";
            row0.Cell(4).Value = "ФИО Клиента";
            row0.Cell(5).Value = "ИД Клиента";

            List<BuildObject> buildObjects =
EfModel.Init().BuildObjects.ToList();
            for (int i = 0; i < buildObjects.Count; i++)
            {
                BuildObject buildObject = buildObjects[i];

                IXLRow row = worksheet.Row(i + 2);
                row.Cell(1).Value =
buildObject.ID_BuildObject;
                row.Cell(2).Value =
buildObject.BuildObjectName;

                MemoryStream stream = new
MemoryStream(buildObject.BuildObjectPhoto);

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

```

        worksheet.AddPicture(stream).MoveTo(row.Cell
(3)).WithSize(150,150);

        row.Cell(4).Value = buildObject.Client.FIO;
        row.Cell(5).Value =
buildObject.Clients_id_Client;
        row.Cell(6).Value = String.Join(",",
buildObject.ObjectDocuments.Select(o => o.ObjectName));

        row.Height = 100;
    }
    workbook.SaveAs(filename);
}

public static void ContractExport(string filename)
{
    XLWorkbook workbook = new XLWorkbook();
    IXLWorksheet worksheet =
workbook.AddWorksheet("Contract");
    worksheet.Column(1).Width = 50;
    worksheet.Column(2).Width = 50;
    worksheet.Column(3).Width = 50;
    worksheet.Column(4).Width = 50;
    worksheet.Column(5).Width = 50;

    IXLRow row0 = worksheet.Row(1);
    row0.Cell(1).Value = "ИД Договора";
    row0.Cell(2).Value = "Номер";
    row0.Cell(3).Value = "Дата";
    row0.Cell(4).Value = "ИД Поставщика";
    row0.Cell(5).Value = "Скан Договора";

    List<Contract> contract =
EfModel.Init().Contracts.ToList();
    for (int i = 0; i < contract.Count; i++)
    {
        Contract contract = contract[i];

        IXLRow row = worksheet.Row(i + 2);
        row.Cell(1).Value = contract.ID_Contract;
        row.Cell(2).Value = contract.Number;
        row.Cell(3).Value = contract.Name;
        row.Cell(4).Value = contract.Date;

        MemoryStream stream = new
MemoryStream(contract.DocumentScan);
        worksheet.AddPicture(stream).MoveTo(row.Cell
(6)).WithSize(150, 150);

        row.Cell(5).Value =
contract.Suppliers_ID_Supplier;

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

```

        row.Height = 100;
    }
    workbook.SaveAs(filename);
}

public static void MaterialExport(string filename)
{
    XLWorkbook workbook = new XLWorkbook();
    IXLWorksheet worksheet =
workbook.AddWorksheet("Material");
    worksheet.Column(1).Width = 50;
    worksheet.Column(2).Width = 50;
    worksheet.Column(3).Width = 50;
    worksheet.Column(4).Width = 50;
    worksheet.Column(5).Width = 50;

    IXLRow row0 = worksheet.Row(1);
    row0.Cell(1).Value = "ИД Материала";
    row0.Cell(2).Value = "Наименование";
    row0.Cell(3).Value = "Фотография";
    row0.Cell(4).Value = "Описание";

    List<Material> materials =
EfModel.Init().Materials.ToList();
    for (int i = 0; i < materials.Count; i++)
        Material material = materials[i];
        IXLRow row = worksheet.Row(i + 2);
        row.Cell(1).Value = material.ID_Material;
        row.Cell(2).Value = material.Name;

        MemoryStream stream = new
MemoryStream(material.ImageMaterial);
        worksheet.AddPicture(stream).MoveTo(row.Cell
(3)).WithSize(150, 150);

        row.Cell(4).Value = material.Description;
        row.Cell(5).Value = String.Join(", ",
material.Materials_has_Suppliers.Select(ms => ms.Supplier.Title
+ " - " + ms.Material_Price));

        row.Height = 100;
    }

    workbook.SaveAs(filename);
}

public static void ObjectDocumentExport(string
filename)
{

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

```

        XLWorkbook workbook = new XLWorkbook();
        IXLWorksheet worksheet =
workbook.AddWorksheet("ObjectDocument");
        worksheet.Column(1).Width = 50;
        worksheet.Column(2).Width = 50;
        worksheet.Column(3).Width = 50;
        worksheet.Column(4).Width = 50;
        worksheet.Column(5).Width = 50;

        IXLRow row0 = worksheet.Row(1);
        row0.Cell(1).Value = "ИД Документа";
        row0.Cell(2).Value = "Наименование";
        row0.Cell(3).Value = "Скан документа";
        row0.Cell(4).Value = "ИД Строительного объекта";

        List<ObjectDocument> objectDocuments =
EfModel.Init().ObjectDocuments.ToList();
        for (int i = 0; i < objectDocuments.Count; i++)
        {
            ObjectDocument objectDocument =
objectDocuments[i];

            IXLRow row = worksheet.Row(i + 2);
            row.Cell(1).Value =
objectDocument.ID_ObjectDocument;
            row.Cell(2).Value =
objectDocument.ObjectName;
            MemoryStream stream = new
MemoryStream(objectDocument.ObjectScan);
            worksheet.AddPicture(stream).MoveTo(row.Cell
(3)).WithSize(150, 150);

            row.Cell(4).Value =
objectDocument.BuildObject_ID_BuildObject;

            row.Height = 100;
        }
        workbook.SaveAs(filename);
    }

    public static void SupplierExport(string filename)
    {
        XLWorkbook workbook = new XLWorkbook();
        IXLWorksheet worksheet =
workbook.AddWorksheet("Supplier");
        worksheet.Column(1).Width = 50;
        worksheet.Column(2).Width = 50;
        worksheet.Column(3).Width = 50;
        worksheet.Column(4).Width = 50;
        worksheet.Column(5).Width = 50;

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

```

        IXLRow row0 = worksheet.Row(1);
        row0.Cell(1).Value = "ИД Поставщика";
        row0.Cell(2).Value = "Наименование";
        row0.Cell(3).Value = "Адрес";
        row0.Cell(4).Value = "Телефон";
        row0.Cell(5).Value = "Фотография";

        List<Supplier> suppliers=
EfModel.Init().Suppliers.ToList();
        for (int i = 0; i < suppliers.Count; i++)
        {
            Supplier supplier = suppliers[i];

            IXLRow row = worksheet.Row(i + 2);
            row.Cell(1).Value = supplier.ID_Supplier;
            row.Cell(2).Value = supplier.Title;
            row.Cell(3).Value = supplier.Adress;
            row.Cell(4).Value = supplier.Phone;

            MemoryStream stream = new
MemoryStream(supplier.Photo);
            worksheet.AddPicture(stream).MoveTo(row.Cell
(5)).WithSize(150, 150);
            row.Height = 100;
        }
        workbook.SaveAs(filename)
    }
}
//Класс импорта

namespace EDODiplom.Classes
{
    class ImportClass
    {
        public static void ImportBuildObject(string
filename)
        {
            XLWorkbook xLWorkbook = new
XLWorkbook(filename);
            IXLWorksheet worksheet =
xLWorkbook.Worksheets.ElementAt(0);

            Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture> PicturesByCellAddress
                = new Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture>();
            foreach (ClosedXML.Excel.Drawings.IXLPicture pic
in worksheet.Pictures)
            {

```

toptem.ru

8 (800) 100-67-87

info@toptem.ru

```

        PicturesByCellAddress.Add(
            pic.TopLeftCell.Address.RowNumber + ":"
+ pic.TopLeftCell.Address.ColumnNumber,
            pic);
    }

    for (int i = 2; i <
worksheet.RangeUsed().Rows().Count() + 1; i++)
    {
        IXLRow row = worksheet.Row(i);
        BuildObject build = new BuildObject
        {
            ID_BuildObject = 0,
            BuildObjectName =
Convert.ToString(row.Cell(2).Value),

            BuildObjectPhoto =
PicturesByCellAddress[i + ":" + 3].ImageStream.ToArray(),
            Clients_id_Client =
Convert.ToInt32(row.Cell(5).Value),
        };

        IEnumerable<ObjectDocument> objectDocuments
= row.Cell(6).Value.ToString().Split(',')
.Select(o =>
            EfModel.Init().ObjectDocuments.
FirstOrDefault(o => o.ObjectName.Trim() == o.Trim()));
        foreach (var objectdocument in
objectDocuments)
        {
            build.ObjectDocuments.Add(objectdocument
);
        }

        EfModel.Init().BuildObjects.Add(build);
    }

    EfModel.Init().SaveChanges();
}

public static void ImportContract(string filename)
{
    XLWorkbook xlWorkbook = new
XLWorkbook(filename);
    IXLWorksheet worksheet =
xlWorkbook.Worksheets.ElementAt(0);

    Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture> PicturesByCellAddress
= new Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture>();

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru

```

        foreach (ClosedXML.Excel.Drawings.IXLPicture pic
in worksheet.Pictures)
        {
            PicturesByCellAddress.Add(
                pic.TopLeftCell.Address.RowNumber + ":"
+ pic.TopLeftCell.Address.ColumnNumber,
                pic);
        }

        for (int i = 2; i <
worksheet.RangeUsed().Rows().Count(); i++)
        {
            IXLRow row = worksheet.Row(i);

            Contract contract = new Contract
            {
                ID_Contract = 0,
                Number =
Convert.ToString(row.Cell(2).Value),
                Name =
Convert.ToString(row.Cell(3).Value),
                Date = row.Cell(4).GetDateTime(),
                Suppliers_ID Supplier =
Convert.ToInt32(row.Cell(5).Value),
                DocumentScan = PicturesByCellAddress[i +
":" + 6].ImageStream.ToArray(),
            };
            EfModel.Init().Contracts.Add(contract);
        }
        EfModel.Init().SaveChanges();
    }

    public static void ImportMaterial(string filename)
    {
        XLWorkbook xLWorkbook = new
XLWorkbook(filename);
        IXLWorksheet worksheet =
xLWorkbook.Worksheets.ElementAt(0);

        Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture> PicturesByCellAddress
            = new Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture>();
        foreach (ClosedXML.Excel.Drawings.IXLPicture pic
in worksheet.Pictures)
        {
            try
            {
                PicturesByCellAddress.Add(
                    pic.TopLeftCell.Address.RowNumber +
":" + pic.TopLeftCell.Address.ColumnNumber,
                    pic);
            }
            catch { }
        }
    }
}

```

toptem.ru
8 (800) 100-67-87
info@toptem.ru

```

        }
        catch
        {
        }
    }

    for (int i = 2; i <
worksheet.RangeUsed().Rows().Count() + 1; i++)
    {
        IXLRow row = worksheet.Row(i);

        Material material = new Material
        {
            ID_Material = 0,
            Name =
Convert.ToString(row.Cell(2).Value),
            ImageMaterial = PicturesByCellAddress[i
+ ":" + 3].ImageStream.ToArray(),
            Description =
Convert.ToString(row.Cell(4).Value),
        };

        List<Supplier> suppliers =
EfModel.Init().Suppliers.ToList();
        IEnumerable<Materials_has_Suppliers>
materials_Has_Suppliers =
row.Cell(0).Value.ToString().Split(',')
.Select(ms =>
        new Materials_has_Suppliers
        {
            Suppliers_ID_Supplier =
suppliers.FirstOrDefault(s => s.Title.Trim() == ms.Split('-',
'-')[0].Trim()).ID_Supplier,
            Material_Price =
Convert.ToDecimal(ms.Split('-', '-') [1].Trim())
        }
    );

        foreach (var materialHasSupplierItem in
materials_Has_Suppliers)
        {
            material.Materials_has_Suppliers.Add(mate
rialHasSupplierItem);
        }
        EfModel.Init().Materials.Add(material);
    }
    EfModel.Init().SaveChanges();
}

public static void ImportObjectDocument(string
filename)

```

toptem.ru
 8 (800) 100-67-87
info@toptem.ru


```

        {
            XLWorkbook xlWorkbook = new
XLWorkbook(filename);
            IXLWorksheet worksheet =
xlWorkbook.Worksheets.ElementAt(0);

            Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture> PicturesByCellAddress
                = new Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture>();
            foreach (ClosedXML.Excel.Drawings.IXLPicture pic
in worksheet.Pictures)
            {
                PicturesByCellAddress.Add(
                    pic.TopLeftCell.Address.RowNumber + ":"
+ pic.TopLeftCell.Address.ColumnNumber,
                    pic);
            }

            for (int i = 2; i <
worksheet.RangeUsed().Rows().Count(); i++)
            {
                IXLRow row = worksheet.Row(i);
                ObjectDocument objectDocument = new
ObjectDocument
                {
                    ID_ObjectDocument = 0,
                    ObjectName =
Convert.ToString(row.Cell(2).Value),
                    ObjectStart = PicturesByCellAddress[i +
":" + 3].ImageStream.ToArray(),
                    BuildObject_ID_BuildObject =
Convert.ToInt32(row.Cell(4).Value),
                };
                EfModel.Init().ObjectDocuments.Add(objectDoc
ument);
            }
            EfModel.Init().SaveChanges();
        }

        public static void ImportSupplier(string filename)
        {
            XLWorkbook xlWorkbook = new
XLWorkbook(filename);
            IXLWorksheet worksheet =
xlWorkbook.Worksheets.ElementAt(0);

            Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture> PicturesByCellAddress
                = new Dictionary<string,
ClosedXML.Excel.Drawings.IXLPicture>();

```

toptem.ru

8 (800) 100-67-87

info@toptem.ru

```

        foreach (ClosedXML.Excel.Drawings.IXLPicture pic
in worksheet.Pictures)
        {
            PicturesByCellAddress.Add(
                pic.TopLeftCell.Address.RowNumber + ":"
+ pic.TopLeftCell.Address.ColumnNumber,
                pic);
        }

        for (int i = 2; i <
worksheet.RangeUsed().Rows().Count(); i++)
        {
            IXLRow row = worksheet.Row(i);

            Supplier supplier = new Supplier
            {
                ID_Supplier = 0,
                Title =
Convert.ToString(row.Cell(2).Value),
                Adress =
Convert.ToString(row.Cell(3).Value),
                Phone =
Convert.ToDouble(row.Cell(4).Value),
                Photo = PicturesByCellAddress[i + ":" +
5].ImageStream.ToArray(),
            };
            EfModel.Init().Suppliers.Add(supplier);
        }
        EfModel.Init().SaveChanges();
    }
}

```

toptem.ru
8 (800) 100-67-87
info@toptem.ru

ПРИЛОЖЕНИЕ Д

ФЛЕШКА С МАТЕРИАЛАМИ

Флешка с материалами

toptem.ru
8 (800) 100-67-87
info@toptem.ru