

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

**ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ CRM ПРИЛОЖЕНИЯ КОМПАНИИ
ООО «ТЕКРО»**

Выпускная квалификационная работа
обучающегося по направлению подготовки
02.03.02. Фундаментальная информатика и информационные технологии
очной формы обучения, группы 07001301
Лифинцева Максима Александровича

Научный руководитель
к.т.н., доцент
Бурданова Е. В.

БЕЛГОРОД 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. ТЕХНИКО-ЭКОНОМИЧЕСКИЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	5
1.1 Характеристика деятельности предприятия ООО «Текро».....	5
1.2 Обоснование необходимости и цели использования программного продукта	8
1.3 Постановка цели и задач	10
1.4 Обзор современных CRM-разработок	12
ГЛАВА 2. ПРОЕКТИРОВАНИЕ CRM – ПРИЛОЖЕНИЯ ДЛЯ КОМПАНИИ ООО «ТЕКРО»И ВЫБОР СРЕДСТВ РЕАЛИЗАЦИИ	20
2.1 Проектирование логической модели базы данных.....	20
2.2 Проектирование физической модели базы данных.....	23
2.3 Выбор средств реализации.....	27
ГЛАВА 3. РЕАЛИЗАЦИЯ CRM ПРИЛОЖЕНИЯ ДЛЯ КОМПАНИИ ООО «ТЕКРО»	36
3.1 Создание интерфейса программного продукта.....	36
3.2 Программная разработка модулей CRM-приложения	39
3.3 Тестирование контрольного примера	42
ЗАКЛЮЧЕНИЕ	50
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	51
ПРИЛОЖЕНИЕ 1	56
ПРИЛОЖЕНИЕ 2	99
ПРИЛОЖЕНИЕ 3	102

ВВЕДЕНИЕ

В современном мире рынок программных продуктов активно развивается. Новые законодательные акты вносят существенные коррективы в развитие отделов продаж. Одной из основных задач данных отделов является оптимизация бизнес-процессов, а также учет клиентской базы, проведенных услуг и внутренних процессов предприятия. Для решения данных задач рациональнее всего будет внедрение и стандартизация в компанию системы взаимодействия с клиентами (CRM) [13].

CRM – система – это специализированный программный продукт, предназначенный для планирования задач и контроля их выполнения, автоматизации процесса взаимодействия с клиентами компании, а именно для увеличения уровня продаж, оптимизации маркетинговой деятельности и учета клиентов, хранения всей документации по проектам и автоматизации их создания [24].

Такие системы предназначены для решения следующего ряда задач:

- получение единой базы данных клиентов и контрагентов компании;
- проведение статистических и аналитических исследований работы отдела продаж;
- повышение эффективности деятельности отдела продаж в целом.

В настоящее время насчитывается большое количество компаний. Их основной деятельностью является работа с клиентами, которая не везде автоматизирована. В результате чего CRM-приложения являются неотъемлемым инструментом в деятельности любой без исключения организации. Именно этим и обуславливается актуальность данной выпускной квалификационной работы.

Предметом выпускной квалификационной работы являются способы проектирования и разработки CRM-системы для отдела продаж компании ООО

«Текро». Объектом - деятельность сотрудников отдела продаж компании ООО «Текро».

Целью выпускной квалификационной работы является проектирование и разработка CRM-приложения для отдела продаж компании ООО «Текро».

Для решения поставленной цели необходимо решение следующих задач:

- изучение деятельности предприятия ООО «Текро» и его структуры;
- провести обзор современных информационных систем, выявить их недостатки;
- провести инфологическое и даталогическое проектирование базы данных CRM-системы для отдела продаж компании ООО «Текро»;
- определить требования к разрабатываемому программному продукту;
- провести тестирование программного продукта.

Выпускная квалификационная работа состоит из введения, 4 глав, заключения, списка использованной литературы и приложения.

В первой главе рассматривается технико-экономический анализ предметной области, организационная структура компании ООО «Текро», обоснование необходимости автоматизации, постановка цели и задач, а также выявляются недостатки в существующих CRM-системах.

Во второй главе обосновывается выбор средств реализации для написания выпускной квалификационной работы.

В третьей главе производится проектирование CRM-приложения для отдела продаж компании ООО «Текро», а также производится инфологическое и даталогическое проектирование предметной области.

В четвертой главе реализуется CRM-приложение для отдела продаж компании ООО «Текро», создается интерфейс программного продукта, описываются математические расчеты, и производится тестирование системы.

Выпускная квалификационная работа написана на 50 листах, включает в себя 3 таблицы, 24 рисунка, 50 источников литературы и 4 приложения.

ГЛАВА 1. ТЕХНИКО-ЭКОНОМИЧЕСКИЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Характеристика деятельности предприятия ООО «Текро»

ООО «Текро» - компания, которая занимается производством кормов, разведением животных (сельскохозяйственных и домашних), а также предоставлением консультаций по товарам и услугам в данной сфере. Данная компания является представителем в Российской Федерации компании ООО «Текро», которая является чешским производителем различного рода премиксов, концентратов и кормов для животных и птиц сельского хозяйства, а также для домашних животных.

Главной миссией организации ООО «Текро» является создание премиксов, лечебных кормов для животных, витаминно-минеральных консистенций, белковых добавок для животных и птиц сельского хозяйства [30].

Целью фирмы считается не только производство премиксов и концентратов для животных и птиц, а также сотрудничество с потребителями, направленное на повышение прибыли предприятия. ООО «Текро» в большей степени нацелено на обеспечение товаром своих потребителей. Основным из направлений работы предприятия считается формирование индивидуальных кормовых программ для потребителей и предоставление им полной консультации по вопросам кормления и содержания животных и птиц. Постоянным потребителям продукции ООО «Текро» обеспечивается техническое сопровождение продукции, проводится первичное тестирование сырья и примесей в Чехии и Российской Федерации.

Компания занимается данной деятельностью уже на протяжении двадцати лет. За этот период, благодаря качеству собственных товаров и

высоким стандартам своей деятельности, успела получить самое значимое для любой фирмы – доверие потребителей. Вопросам свойств и безопасности изготавливаемых примесей компания уделяет особое внимание. Поставщиками ингредиентов для изготовления собственной продукции ООО «Текро» считаются ведущие крупные производители, которые гарантируют качество продаваемого товара.

Преимуществами ООО «Текро» перед другими аналогичными компаниями являются:

- оперативность производства и доставки;
- оптимальная цена;
- качество производимого товара;
- ориентированность на потребности клиента;
- индивидуальный подход к каждому клиенту.

ООО «Текро», расположенный на территории Российской Федерации, предоставляет ряд услуг [30]:

- доставка кормов и примесей в любые регионы государства – организация производит транспортировку товара в кратчайшие сроки при помощи автотранспортных компаний, которые в свою очередь осуществляют доставку груза до места назначения в прописанные сроки и согласно правилам перевозок;

- программа сопровождения потребителей компании – организация производит отбор кормового сырья в сельском хозяйстве, рассчитывает рацион питания для каждого животного, проводит консультации по кормлению. Есть ряд представителей компании в различных регионах Российской Федерации, которые работают напрямую с хозяйствами, консультируют их, анализируют деятельность и дают рекомендации по кормлению. При необходимости возможна консультация основных специалистов Российской Федерации и Чехии;

- проведение анализов в лабораторной базе – в структуре компании имеется подразделение, в котором разрабатываются новейшие технологии

производства и примесей, а также улучшаются уже созданные технологии. Одной из задач данного подразделения является проведение опытов на 2 современных свиноводческих комплексах, одной птицеферме и молочной ферме, которые расположены в Чешской республике.

В структуру фирмы входят премиксные и комбикормовые фабрики, оснащенные передовыми технологиями для соблюдения стандартов входящих и исходящих товаров, свинокомплексов, птицефабрик, инкубаторов, а также ферм для животных в таких республиках как: Чехия, Словакия, а также в Украине и Российской Федерации.

В настоящее время организация состоит из следующих компаний:

- в Чешской республике: 12 свинокомплексов, 7 птицефабрик, 7 ферм крупного рогатого скота, 2 сельских хозяйства по разведению индеек, 4 яичные птицефабрики, а также мясокомбинат, убойных цех и сельские угодья для выкармливания животных и птиц;

- в Словацкой республике: 3 свинокомплекса и хозяйство по разведению кур-несушек (в том числе и бройлеров);

- в Украине: 2 инкубатора;

- в Российской Федерации: 10 свинокомплексов, 5 птицефабрик, 3 фермы крупного рогатого скота, 1 сельское хозяйство по разведению индеек, 5 яичных птицефабрик, а также 2 мясокомбината, убойный цех и сельское угодье для выкармливания сельскохозяйственных и домашних животных и птиц.

В деятельности предприятия ООО «Текро» используется линейная организационная структура. Она представляет собой «шахтный» способ построения процессов по различным подразделениям компании (к примеру, отдел продаж, отдел закупок, администрация и бухгалтерия).

В результате данной организационной структуры предприятия могут оцениваться показатели работы, как сотрудников, так и отдела в целом, а также вводиться система штрафов и поощрений.

Все отделы, являющиеся «второстепенными» в данной структуре, работают по поручениям вышестоящих руководителей предприятия.

Существующие подразделения компании представлены на рис. 1.1.



Рис. 1.1. Существующие подразделения компании ООО «Текро»

В ООО «Текро» существует ряд подразделений, функции которых описаны в таблице 1.1.

Таблица 1.1

Функции подразделений

Название подразделения	Функции подразделения
Отдел логистики	- выбор машины для перевозки; - контроль бартера; - контроль движения.
Отдел продаж	- поиск новых клиентов; - продажа премиксов и прочих кормов; - оформление документации на продажу.
Отдел закупок	- поиск новых поставщиков; - закупка сырья у поставщиков; - оформление документации на закупку.
Отдел бухгалтерии	- контроль финансового положения; - осуществление бухгалтерских выплат; - составление сметы доходов (расходов).
Отдел по кормлению	- создание рецептов; - адаптация рецептов под нужды покупателей; - контроль наличия сырья.

1.2 Обоснование необходимости и цели использования программного продукта

ООО «Текро» занимается производством кормов, разведением животных (сельскохозяйственных и домашних), а также предоставлением консультаций

по товарам и услугам в данной области. Данная компания является представителем в Российской Федерации ООО «Текро», которая является чешским производителем различного рода примесей, концентратов и кормов для животных и птиц сельского хозяйства, а также для домашних животных [30].

Для оптимизации деятельности предприятия и разделения обязанностей между сотрудниками ООО «Текро» в компании создан отдел, занимающийся продажами примесей, концентратов и кормов для животных и птиц сельского и домашнего хозяйства. Отдел по продажам решает ряд следующих задач:

- оформление и заключение различного рода документации с потребителями товаров;
- проведение консультаций клиентам по товарам предприятия;
- прием и ведение учета заявок на заказ товара;
- рассмотрение вопросов ценовой политики и схем скидок организации.

В настоящее время в ООО «Текро» для обеспечения деятельности отдела продаж используются информационные технологии, такие как Microsoft Word, Excel. Офисный пакет служит лишь для создания и распечатывания документации. Дальнейшее решение задач предприятия традиционными средствами возможно, но все больше становится неэффективным.

Офисный пакет, а именно программный продукт Microsoft Excel, не позволяет в полной мере выполнять функции системы управления базами данных, что в свою очередь увеличивает время на поиск необходимой информации работнику предприятия. Данный факт говорит об уменьшении производительности сотрудников и прибыли предприятия.

Со стороны сотрудника предприятия информационные технологии дают ряд неудобств в поиске нужных сведений, в фильтрации и создании необходимой отчетной документации. Все заказы, поступившие от клиентов, вводятся в табличный редактор Microsoft Excel. После этого данные о заявке передаются соответствующим сотрудникам. В последствие выполнения заказа в Excel вводится отметка об исполнении заказа. Так как данный процесс не

автоматизирован, и все данные вводятся вручную, это занимает значительное время.

При увеличении количества записей, увеличивается размер файла, а соответственно и объем занимаемой памяти в системе. Главным недостатком является то, что вся информация хранится в компьютере и соответственно не дублируется, что может привести к полной потере данных при возникновении проблем с аппаратным обеспечением [9].

В настоящее время в ООО «Текро» используются ряд компьютеров, имеющие следующие характеристики:

- тип процессора персонального компьютера – Intel, AMD;
- объем оперативной памяти от 2 гб до 8 гб;
- емкость жесткого диска от 300 гб до 2 тб;
- операционная система семейства Windows;
- офисный пакет Microsoft Office 2007 – 2016.

Следовательно, при внедрении созданного программного продукта возможны следующие преимущества его использования:

- быстрый поиск и фильтрация необходимых сведений сотрудником компании;
- обеспечение защищенности и хранения информации;
- автоматическое оформление договоров с покупателями и поставщиками;
- снижение рутинной работы сотрудников предприятия.

1.3 Постановка цели и задач

Во время написания выпускной квалификационной работы разрабатывался программный продукт для отдела продаж предприятия ООО «Текро».

Целью данной автоматизации является проектирование и разработка CRM приложения для отдела продаж компании ООО «Текро».

При внедрении CRM-приложения автоматизируется учет проведенных продаж компании, процесс составления договоров купли-продажи, выписки командировочных листов сотрудников, а также документооборот предприятия в целом.

Информационная система имеет простой и понятный интерфейс, в котором быстро разберется даже не самый опытный пользователь.

Данное программное обеспечение предполагает обращение к локальной базе данных, что означает, что один сотрудник, не сможет просматривать данные другого. Таким образом, благодаря данной функции реализована защищенность и конфиденциальность базы данных.

Для решения поставленной цели необходимо решение следующих задач:

- изучение деятельности предприятия ООО «Текро» и его структуры;
- провести обзор современных информационных систем, выявить их недостатки;
- провести инфологическое и даталогическое проектирование базы данных CRM-системы для отдела продаж компании ООО «Текро»;
- определить требования к разрабатываемому программному продукту;
- провести тестирование программного продукта.

Кроме этого, программный продукт «ТекроCRM» должен иметь следующий ряд свойств:

- работа с клиентской базой данных;
- автоматизация процессов совершения сделок;
- хранение и распределение документооборота;
- создание отчетов деятельности;
- экспорт информации из базы данных в информационные технологии Excel;
- статистика и анализ деятельности сотрудников.

1.4 Обзор современных CRM-разработок

В настоящее время существует большое количество разработок для автоматизации деятельности предприятия. К таким относят:

- MRP – системы планирования материальных ресурсов предприятия – это информационная система, предназначенная для планирования потребностей в материалах;

- MRP II – система планирования ресурсов организации – это информационная система, предназначенная для планирования внутренних ресурсов предприятия;

- ERP – система планирования ресурсов организации – это информационная система, предназначенная для сокращения времени выпуска товара, а также для управления товарно-материальными запасами предприятия;

- CSRP – система управления организацией – это информационная система, предназначенная для взаимодействия с клиентами, а также для автоматизации документооборота предприятия;

- SCM – система управления цепочками поставок организации – это информационная система, предназначенная для автоматизации процесса начинающегося закупкой сырья у поставщика и заканчивающегося продажей готового продукта клиенту;

- CRM – система управления взаимоотношениями с клиентами организации – это информационная система, предназначенная для автоматизации стратегических действий с клиентами, в результате чего предприятие стремится к максимизации прибыли [8, 11].

В таблице 1.2 представлены основные интегрируемые информационные системы управления предприятием (иначе ИСУП).

Таблица 1.2

Основные ИСУП

	Информационная система
ИСУП для управления деятельностью крупного предприятия	ERP
ИСУП для управления деятельностью среднего предприятия	CSRP
	ERP
	CRM
ИСУП для управления деятельностью малого и среднего предприятия	ERP
	MRP
	MRPII
	CRM

Изучив современные информационные системы для управления предприятием, в качестве основы для написания программного продукта были выбраны CRM-системы.

CRM - разработки используются для автоматизации процесса взаимодействия с клиентами и увеличения уровня продаж предприятия, оптимизации маркетинговой деятельности и совершенствования обслуживания клиентов путем сохранения сведений о клиентах и историй взаимоотношения с ними, установления и совершенствования бизнес-процессов компании в целом [12, 14].

CRM - системы включают в себя функционал, который представлен на рис. 1.2.



Рис.1.2. Функционал CRM - системы

В настоящее время существует ряд CRM-разработок:

1. «CRM MANGO OFFICE» - программный продукт, при помощи которого осуществлять деятельность можно из любой географической точки, лишь при наличии учетных сведений и выхода во всемирную сеть Интернет. При помощи статистических данных, выполняемых автоматизированной информационной системой, ведется учет совершенных сделок и обращений клиентов предприятия. Кроме этого, есть возможность отслеживания работы сотрудников компании [47].

В «CRM MANGO OFFICE» есть ряд преимуществ:

- хранение истории взаимоотношений с клиентами компании – при помощи телефонной и почтой интеграции проводится полная история отношений с момента первого звонка либо сообщения с клиентом, которые автоматически регистрируются в данной системе. Все данные о сделках с клиентом хранятся в специализированных дата - центрах;

- продажи по установленным сценариям предприятия – есть возможность рассмотрения продажи по отдельным этапам. В основном этапы подразделяются на: звонки от клиентов либо email, раскрытие потребностей,

оформление и отправка платных предложений, заключение контрактов, проверка оплаты счета и другие;

- принципы и анализ продаж – возможен полный просмотр входящих и исходящих звонков каждого сотрудника организации. В данном модуле будут доступны такие возможности как: общая сумма звонков и письменных обращений в компанию, среднее время обработки каждого обращения, учет товаров, проверка соблюдения стандартов при общении с клиентами;

- «холодный» обзвон потребителей – при помощи лишь одной кнопки возможен обзвон клиентов по заблаговременным настроенным сценариям беседы и подготовленным базам. Фиксация ответов клиентов автоматически происходит в CRM-систему, также возможен прозрачный контроль результатов при учете каждого без исключения звонка;

- мотивация сотрудников и всей команды в целом – позволяет следить за различного рода показателями любого сотрудника компании либо отдела: в данном случае уделяется большое внимание на взаимоотношения с клиентами, количеству принятых заказов к доведенным до продажи, общей сумме товарооборота и среднему чеку. Любая из данных характеристик может быть использована для получения положительных результатов деятельности компании [47].

2. «Битрикс 24» – это программный продукт, созданный для автоматизации деятельности компании. Он упрощает такие процессы как: управление проектами компании, контроль поручений администрации, документооборот, работу с клиентами, подтверждение документов, а также расчетных счетов, командировок и отпусков. Стоит отметить, что «Битрикс 24» упрощает рутинную деятельность сотрудников с клиентами компании, что в свою очередь приводит к увеличению прибыли организации [6].

В «Битрикс 24» есть ряд преимуществ:

- общение с постоянными клиентами – при помощи данной функции клиенту не нужно тратить время на ожидание соединения с определенным

сотрудником, постоянных клиентов система соединяет автоматически с нужными менеджерами компании;

- борьба с повторяющимися данными – при помощи данной функции происходит проверка данных на повторение. В случае дубликата есть 4 варианта их работы: пропустить, удалить, дополнить или же создать новый;

- планирование дел – при помощи данной функции возможно календарное планирование дел. Возможны оповещения следующего характера: напоминание о невыполненной задаче, важной встрече, звонке, отправке отчетной документации. Кроме этого, возможна фильтрация по важности запланированных дел;

- отправка писем постоянным клиентам – при помощи данной функции предусмотрена рассылка клиентам шаблонных заготовок, заранее созданных в системе «Битрикс 24»;

- управление сделками компании – при помощи данной функции возможно отражение этапа сделки компании. Возможны следующие этапы: обработка, ожидание, переговоры, коммерческое предложение или заключение [6].

3. «1С: CRM СТАНДАРТ» - программный продукт, направленный на решение функциональных задач, часто встречающихся при организации работы с клиентами в малых предприятиях [46].

В «1С: CRM СТАНДАРТ» есть ряд преимуществ [46]:

- возможность изменения и добавления нового перечня возможностей программного продукта;

- возможность работы через облачную платформу (через web-клиента);

- интеграция с офисным пакетом Microsoft (Word, Excel);

- встроенный почтовый клиент.

4. «Terrasoft CRM» - программный продукт, который охватывает главные области управления отношений с клиентами и организации внутреннего документооборота компании. Особенностью данного продукта является клиент - серверная технология доступа к данным, которая охватывает 2 схемы

лицензирования – именные и конкурентные лицензии, приобретаемые без срока использования [48].

В «Terrasoft CRM» есть ряд преимуществ:

- управление полной информацией о клиентах – данная функция позволяет хранить полную историю взаимоотношений с клиентами, имеет удобный доступ к сведениям о клиенте, а также предусмотрена возможность создания новых полей и закладок;

- управление бизнес-процессами – данная функция позволяет автоматизировать рутинные операции предприятия, предусмотрена возможность ветвления действий по бизнес-процессу, производится контроль выполнения функций сотрудников компании;

- управление продажами – данная функция предоставляет полный спектр функций, необходимых для управления сделками, проектами, а также учитывает контроль сроков оплаты;

- статистика и аналитика – данная функция позволяет проводить статистические и аналитические данные на основе продаж компании;

- управление временем сотрудника компании - при помощи данной функции возможно календарное планирование дел, создание оповещений о предстоящем перечне дел [48].

5. «ТекроCRM» - это CRM - приложение, предназначенное для упрощения работы с клиентами и внутренними функциями компании ООО «Текро». Программный продукт в большей степени направлен на уменьшение времени работы сотрудника с клиентами и на оптимизацию документооборота предприятия.

В «ТекроCRM» есть ряд преимуществ:

- управление информацией о клиентах, а также возможность разделения клиентов на «лиды» и «контакты»;

- управление бизнес - процессами предприятия – автоматизация документооборота компании ООО «Текро»;

- иерархия документации – расположение файлов в директориях клиентов и пользователей, что в свою очередь позволяет быстро найти документы без помощи автоматизированной информационной системы;

- сбор статистических данных и создание отчетов;

- используются локальные безсерверные базы данных – позволяет не перегружать сервер ООО «Текро».

Для сравнения описанных систем необходимым является проведение их анализа с точки зрения работоспособности и функционала программных продуктов. Анализ данных продуктов производился по следующим критериям:

- разделение клиентов на «лиды» и «контакты» - возможен быстрый поиск того или иного клиента;

- наличие email - рассылки – наличие рассылки сообщений клиентам компании;

- работа со сделкой – автоматизированный документооборот предприятия;

- интуитивно понятный интерфейс и простота использования – система не требует дополнительного обучения;

- «заточенность» под определенную компанию – наличие в автоматизированной системе «лишних» функций, которые не требуются определенной фирме.

Кроме этого, CRM-приложения оценивались по 3-бальной шкале, затем данные баллы суммировались. Баллы были разбиты следующим образом:

- 0 баллов – функция или возможность отсутствует в системе;

- 1 балл – имеется функция или возможность в системе, но использование практически невозможно;

- 2 балла – имеется функция или возможность в системе, но раскрыта не в полном объеме;

- 3 балла – имеется функция или возможность в системе, раскрыта в полном объеме.

Анализ программных продуктов по описанным выше критериям представлен в таблице 1.3.

Таблица 1.3

Анализ программных продуктов

Критерий \ Система	CRM Mango	Битрикс 24	1С: CRM	Terrasoft CRM	ТекроCRM
Разделение на «лиды» и «контакты»	0	3	0	0	2
Наличие email-рассылки	0	2	2	2	1
Работа со сделкой	3	2	2	1	2
Интуитивно понятный интерфейс и простота использования	3	2	2	2	3
«Заточенность» под определенную компанию	0	0	0	0	3
Итого:	6	9	6	5	11

Таким образом, можно сделать вывод, что наиболее подходящей системой для компании ООО «Текро» является «ТекроCRM». В данной системе реализованы все функции, необходимые для деятельности отдела продаж предприятия. Отличительной особенностью «ТекроCRM» является «заточенность» под определенный функционал предприятия.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ CRM ПРИЛОЖЕНИЯ ДЛЯ КОМПАНИИ ООО «ТЕКРО» И ВЫБОР СРЕДСТВ РЕАЛИЗАЦИИ

2.1 Проектирование логической модели базы данных

Проектирование логической модели базы данных представляет собой процесс конструирования единой модели организации на базе моделей пользователей системы, являющейся независимой от отличительных характеристик реально применяемой системы управления базами данных и иных физических условий [27].

Логическое проектирование состоит из ряда этапов:

- преобразование локальной концептуальной модели базы данных в локальную логическую модель (удаление взаимосвязей, рекурсивных взаимосвязей);
- определение комплекса взаимоотношений исходя из структуры локальной логической модели базы данных;
- проверка модели при помощи правил нормализации базы данных;
- проверка модели в отношении транзакций пользователей базы данных;
- создание диаграммы «сущность – связь» базы данных;
- определение требований поддержки целостности информации базы данных (к ним относятся ограничения и целостность сущностей базы данных);
- обсуждение и утверждение созданных локальных логических моделей базы данных с конечными пользователями системы.

Логическое проектирование базы данных описывает понятия изучаемой предметной области, а также не только связи между основными определениями, но и возможные ограничения на информацию, налагаемую предметной областью.

Логическое проектирование базы данных представляет собой начальный прототип будущей системы. Стоит отметить, что логическое проектирование осуществляется в терминах информационных единиц, но без определенной привязки к определенной системе управления базами данных. Кроме того, логическая модель базы данных не всегда может быть выражена средствами реляционной модели данных [28].

В настоящее время одним из главных продуктов разработки логической модели базы данных являются различного рода варианты ER-диаграмм (иначе диаграммы «сущность – связь»). В данного рода диаграммах одна и та же модель может быть реализована как реляционной, так и в иерархической, сетевой и постреляционной системе управления базами данных.

Основными понятиями диаграмм «сущность – связь» являются:

- сущность – это настоящий или вымышленный объект, имеющий определенное значение для изучаемой предметной области, данные о которой подлежат хранению. Другими словами, сущность является таблицей реляционной модели создаваемой базы данных;

- взаимосвязи – это средство, при помощи которого представляются отношения между объектами базы данных. Взаимосвязи между сущностями бывают: бинарные, тернарные, а также n-арные;

- атрибут – это объект сущности, который имеет ряд характеристик [27].

Связи по признаку множественности могут быть трех типов [28]:

- 1) «один-к-одному» (1:1);
- 2) «один-ко-многим» (1: M);
- 3) «многие-ко-многим» (M: M).

В данной выпускной квалификационной работе между всеми сущностями базы данных были реализованы связи «один – ко – многим».

На рис. 2.1. представлена логическая схема данных CRM-приложения «Текро».

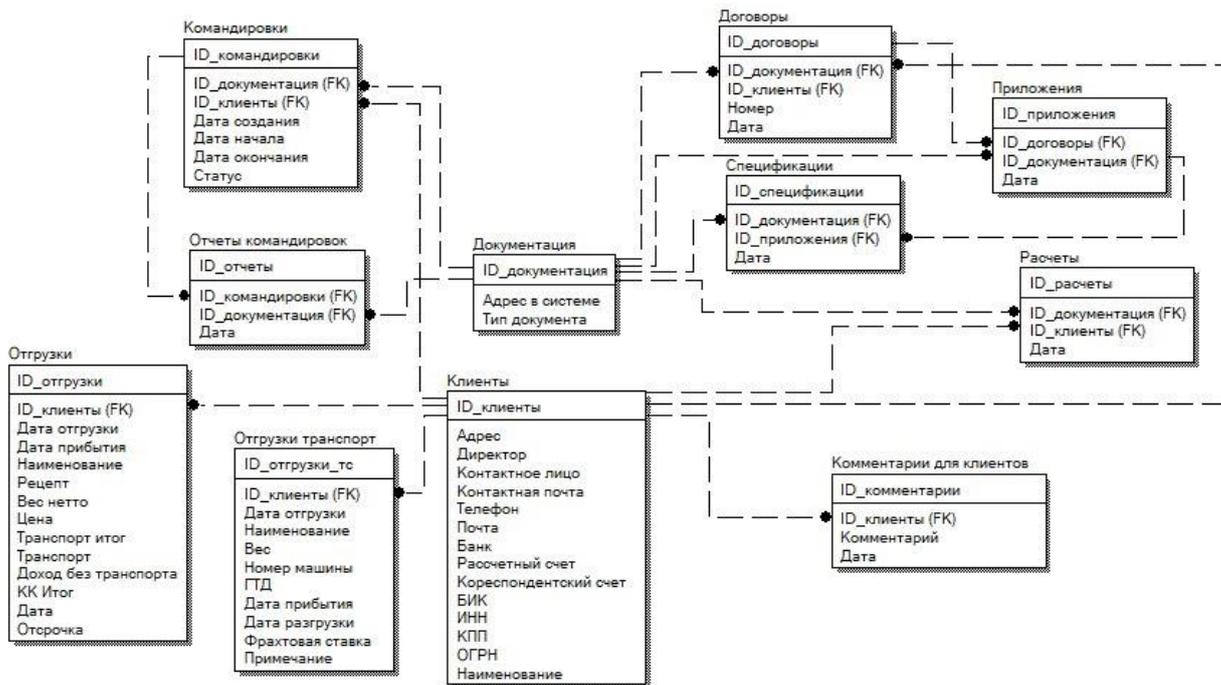


Рис. 2.1. Логическая схема данных

Сущностями данной БД являются:

- командировки – включает в себя информацию о датах и статусе командировки, а также о расположении документа на диске;
- отчеты командировок – включает в себя информацию о дате создания отчета и расположении его на диске;
- отгрузки – включает в себя информацию об отгрузках текущего пользователя;
- отгрузки транспорт – включает в себя информацию о транспорте и датах отгрузки товара;
- клиенты – содержит в себе информацию о клиентах компании, а также данные для создания необходимой документации;
- комментарии для клиентов – содержит в себе информацию о заметках для клиентов компании;
- документация – содержит в себе информацию о месте документа на диске;

- договоры – содержит в себе информацию о номере договора и дате его создания, а также о его расположении на диске;
- спецификации – содержит в себе информацию о дате создания и расположении на диске;
- приложения – содержит в себе информацию о дате создания и расположении на диске;
- расчеты – содержит в себе информацию о расчетах для клиента и дате создания.

2.2 Проектирование физической модели базы данных

Физическое проектирование базы данных - процесс подготовки описания реализации базы данных на вторичных запоминающих устройствах; на этом этапе рассматриваются основные отношения, организация файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, а также все связанные с этим ограничения целостности и средства защиты [38].

Физическое проектирование является третьим и последним этапом создания проекта базы данных, при выполнении которого проектировщик принимает решения о способах реализации разрабатываемой базы данных. Во время предыдущего этапа проектирования была определена логическая структура базы данных (которая описывает отношения и ограничения в рассматриваемой прикладной области).

Как правило, основной целью физического проектирования базы данных является описание способа физической реализации логического проекта базы данных.

В случае реляционной модели данных под этим подразумевается следующее:

- создание набора реляционных таблиц и ограничений для них на основе информации, представленной в глобальной логической модели данных;

- определение конкретных структур хранения данных и методов доступа к ним, обеспечивающих оптимальную производительность СУБД;

- разработка средств защиты создаваемой системы.

Этапы концептуального и логического проектирования больших систем следует отделять от этапов физического проектирования.

Проектирование базы данных — это процесс, который имеет свое начало, но не имеет конца и состоит из бесконечного ряда уточнений. Его следует рассматривать, прежде всего, как процесс познания [38].

Особо важную роль в общем процессе успешного создания системы играет концептуальное и логическое проектирование базы данных. Если на этих этапах не удастся получить полное представление о деятельности предприятия, то задача определения всех необходимых пользовательских представлений или обеспечения защиты базы данных становится чрезмерно сложной или даже неосуществимой. К тому же может оказаться затруднительным определение способов физической реализации или достижения приемлемой производительности системы. С другой стороны, способность адаптироваться к изменениям является одним из признаков удачного проекта базы данных.

Физическое проектирование баз данных состоит из следующих этапов:

- перенос глобальной логической модели данных в среду целевой СУБД;
- проектирование основных отношений;
- разработка способов получения производных данных
- реализация ограничений предметной области;
- проектирование физического представления базы данных;
- анализ транзакций;
- выбор файловой структуры;
- определение индексов;
- определение требований к дисковой памяти;
- проектирование пользовательских представлений;
- разработка механизмов защиты;

- обоснование необходимости введения контролируемой избыточности;
- текущий контроль и настройка операционной системы [39].

Физическое проектирование баз данных включает шесть основных этапов. Логическое проектирование охватывает три первых этапа разработки баз данных, а физическое проектирование — этапы 4-9.

Реализация ограничений предметной области физического проектирования включает разработку основных отношений и реализацию ограничений предметной области с использованием доступных функциональных средств целевой СУБД. На этом этапе должно быть также принято решение по выбору способов получения производных данных, которые включены в модель данных.

Проектирование физического представления базы данных включает выбор файловой организации и индексов для основных отношений. Как правило, СУБД для персональных компьютеров имеют фиксированную структуру внешней памяти, а другие СУБД предоставляют несколько альтернативных вариантов файловой организации для хранения данных. С точки зрения пользователя организация внутренней структуры хранения отношений должна быть совершенно прозрачной — пользователь должен иметь возможность получать доступ к любому отношению и к отдельным его строкам без учета способа хранения данных. Это означает, что СУБД должна обеспечивать полную независимость физического хранения данных от их логической организации. Только в этом случае внесение изменений в физическую организацию базы данных не окажет никакого влияния на работу пользователей [39].

Соответствие между логической моделью данных и физической моделью данных определяется внутренней схемой базы данных. Разработчик должен предоставить подробные физические проекты базы данных с учетом применяемой СУБД и операционной системы. В проекте реализации базы данных в СУБД разработчик должен определить структуры файлов, которые будут использоваться для представления каждого отношения. В проекте

реализации базы данных в операционной системе разработчик должен указать расположение отдельных файлов и обеспечить необходимую их защиту.

На этапе анализа транзакций необходимо принять решение о том, как должно быть реализовано каждое пользовательское представление. А на этапе выбора файловой структуры осуществляется проектирование средств защиты, необходимых для предотвращения несанкционированного доступа к данным, включая управление доступом к основным отношениям.

На этапе определения индексов анализируется также необходимость снижения уровня требований нормализации данных в логической модели, что может способствовать повышению общей производительности системы. Однако эти действия следует предпринимать только в случае реальной необходимости, поскольку введение в базу данных избыточности неизбежно вызовет появление проблем с поддержанием целостности данных. На этапе 9 описан способ организации текущего контроля операционной системы, позволяющий своевременно обнаруживать и устранять все проблемы производительности, которые могут быть решены на уровне проекта, а также учитывать новые или изменившиеся требования.

На рис. 2.2. показана физическая схема данных CRM-приложения «Текро».

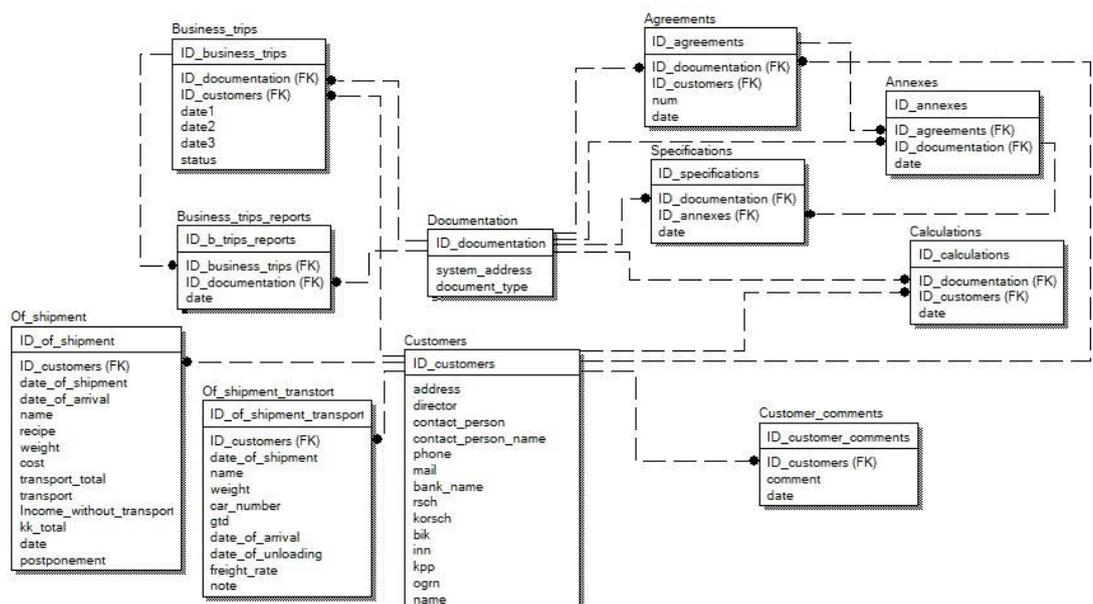


Рис. 2.2. Физическая схема данных

После физического проектирования базы данных приложения «ТекроCRM» можно переходить к выбору средств реализации CRM–системы.

2.3 Выбор средств реализации

Обоснование выбора СУБД SQLITE. SQLite – это без серверная реляционная база данных, запросы к которой осуществляются при помощи языка SQL. Недостатком ее является то, что она поддерживает не все особенности языка SQL, а также по своему функционалу во многом уступает более развитым системам управления базами данных, однако SQLite подходит для хранения и извлечения данных предприятия [41].

Современные системы управления базами данных, к примеру, MySQL (в том числе MSSQL, Oracle, PostgreSQL) включают в себя отдельные сервера, которые поддерживают всю работу базы данных и прослушивающих определенный порт, на предмет обращения посетителей. Движок SQLite и интерфейс к ней реализованы в одной библиотеке, собственно, благодаря чему увеличивается скорость исполнения запросов. Данный сервер нередко считают встроенным.

Стоит отметить, что SQLite считается бестиповой базой данных. Она включает в себя несколько типов данных, таких как [41]:

- «integer» - целочисленный тип данных – используется лишь для первичного ключа таблицы;
- «text» - текстовый тип данных – используется для всех атрибутов таблицы, ограничений по символам в данном типе нет;
- «real» - числовой тип данных с плавающей точкой – используется для записи вещественных чисел;
- «blob» - бинарный тип данных – используется для хранения файлов информационной системы.

Вся информация базы данных хранится в файлах. Число баз данных, а также количество таблиц в них, ограничено исключительно объемом свободной

памяти, который имеется на сайте. Максимально допустимый объем одной базы данных может составлять 2 Тб. Исходя из того, что вся информация хранится в файлах, проблем с перемещением базы данных с одного места в другое нет, необходимо лишь скопировать нужные файлы [5].

У системы управления базами данных SQLite есть ряд преимуществ:

- популярность – система управления базами данных применяется в браузерах, музыкальных устройствах и на различных смартфонах;
- простота и удобство в использовании;
- надежность – при выпуске каждой версии системы управления базами данных SQLite проводится ряд тестов на проверку функционала;
- отсутствует необходимость в настройке сервера системы управления базами данных;
- вероятность свободного распространения с собственным продуктом;
- полностью независимая лицензия;
- поддержка кроссплатформенности системы управления базами данных (устанавливается на Linux, Windows, MacOS);
- высокая скорость обработки запросов, чем в таких СУБД как MySQL, PostgreSQL;
- поддержка версий языка запросов к базам данных SQLite-92;
- возможность создания транзакций, триггеров, а также представлений;
- безопасность – созданная база данных хранится в одном файле, причем права доступа можно контролировать стандартными средствами операционной системы;
- возможность использования разные высокоуровневые языки программирования, такие как: C, C++, C#, PHP, Java, Python;
- экономичная архитектура, которая представляет собой цепочку «клиент» - «файлы» системы управления базами данных;
- имеется поддержка гибкого механизма для передачи ресурсов информационной базы в процедурные интерфейсы;

- мощный объектно-ориентированный интерфейс – СУБД может быть использована для быстрого и эффективного извлечения сведений из базы данных [41].

Кроме преимуществ система управления базами данных SQLite имеет ряд недостатков:

- не подходит для работы с крупными проектами;
- различная совместимость версий – к примеру, 2 версия системы управления базами данных SQLite полноценно поддерживается любой версией PHP 5.x, а 3 имеет ряд ограничений при использовании версий PHP;
- функционал системы плохо работает, если транзакция больше 50 Мбайт;
- имеется ограничение столбцов по дефолту – не более 2000 столбцов.

Как и в любом другом программном обеспечении в SQLite имеется ряд преимуществ и недостатков. Функционал СУБД SQLite в большей степени подходит для малых предприятий, чем для деятельности крупных. Данное ограничение говорит о том, что структура SQLite основана лишь на одном файле, который не предусматривает распределение информации на многократно создаваемые файлы, кроме этого используется блокировка на запись [41].

Обоснование выбора операционной системы Windows 10. Windows 10 – это операционная система для персональных компьютеров, созданная фирмой Microsoft. Операционная система призвана стать единой для различного рода устройств [49]:

- персональных компьютеров;
- планшетов;
- смартфонов;
- консоли Xbox One.

Поддержка обновлений операционной системы Windows 10 происходит на протяжении всего цикла обновлений. На протяжении первого года после выхода пользователи, использующие лицензии операционных систем Windows

7, Windows 8, а также Windows 8.1, имели возможность бесплатно обновиться на Windows 10.

В Windows 10 имеются 2 нововведения, такие как: голосовой помощник Кортана и создание и переключение несколько рабочих столов.

У операционной системы Windows 10 есть ряд преимуществ [26]:

- универсальность – имеется возможность запуска приложений на смартфонах, планшетах, персональных компьютерах;

- улучшение кнопки пуск – меню «Пуск» разделено на 2 части: одна часть представляет собой стандартный тип Windows 7, а вторая часть больше относится к Windows 8;

- улучшение поиска приложений – для поиска в Windows 10 создана отдельная кнопка, кроме этого оптимизирован поиск приложений;

- возможность работы с виртуальными рабочими столами – создана специальная кнопка для вызова рабочих столов операционной системы, имеется возможность увидеть все рабочие столы одновременно, а также управлять ими;

- прикрепление рабочих окон – развитие функции Snap, с помощью которой можно прикреплять окна на рабочий экран, увеличение размещение окон на рабочем столе до 4;

- усовершенствование менеджера файлов – предполагает использование того стиля, который подходит больше всего;

- усовершенствование командной строки – имеется возможность вставки, выделения и копирования текста;

- просмотр запущенных задач – созданное приложение позволяет закреплять запущенные задачи непосредственно на панели, запуск происходит при нажатии единственной кнопки;

- усовершенствование проводника – созданы новые кнопки «Добавить в избранное» и «Поделиться»;

- возможность использования «отпечатка пальцев» на смартфонах или планшетах.

Кроме преимуществ операционная система Windows 10 имеет ряд недостатков [49]:

- при небольшой производительности скорость загрузки операционной системы Windows 10 намного ниже, чем у Windows 8;
- не большое количество приложений, адаптированных к операционной системе Windows 10;
- помощник Кортана работает лишь на английском языке;
- операционная система обновляется автоматически, не предупредив пользователя.

Стоит отметить, что операционная система Windows 10 включила в себя лучшие возможности Windows 7 и Windows 8. Особенностью операционной системы является то, что в ней имеется возможность создания индивидуальных настроек и изменение визуализации экрана.

Обоснование выбора программного продукта Visual Studio 2017. Microsoft Visual Studio – это программный продукт, разработанный компанией Microsoft, и включающий в себя интегрированную среду разработки программного обеспечения [23].

Данный программный продукт позволяет создавать не только консольные приложения, а также приложения с графическим интерфейсом, в том числе с поддержкой технологий Windows Form, а также создавать web-сайты и web-приложения [42].

Программный продукт Visual Studio дает мощные и эффективные средства написания, исправления, компиляции и отладки Windows-приложений, использующих для своего написания платформу .NET. Visual Studio поддерживает такие высокоуровневые языки программирования как [22]:

- C#;
- VB.NET;
- C++;
- J#.

Платформа .NET считается открытой средой. Это означает, что собственно компиляторы могут поставляться и сторонники разработчиками [19].

Под .NET Framework понимается платформа для написания программ, выпущенная фирмой Microsoft. Основой данной платформы выступает среда исполнения Common Language Runtime (иначе CLR) [40].

Данная платформа считается технологией компании Microsoft, что позволяет говорить о том, что она рассчитана в большей мере на операционные системы семейства Windows. Однако существуют независимые проекты (к примеру, Mono и Portable.NET), которые решают данную проблему, с их помощью можно запустить платформу на других операционных системах [44].

Windows – приложение в ходе создания называется проектом. Он соединяет в себе все необходимое:

- папки;
- файлы;
- ссылки;
- прочие ресурсы.

Программный продукт Visual Studio позволяет создавать Windows-приложения разных типов [22]:

- windows-приложение использует составляющие компоненты интерфейса Windows, в том числе формы, кнопки и флажки;
- стандартное консольное приложение, которое результат работы приложения выводит в окно командного процессора;
- библиотека классов соединяет в себе все классы, созданные для использования в других Windows-приложениях;
- составляющие web-сервиса, способы и методы которого могут вызываться через всемирную сеть Интернет.

В программном обеспечении Visual Studio есть ряд преимуществ [23]:

- быстрое создание интеллектуальных Windows-приложений – усовершенствована проверка зависимостей, навигация по коду, а также предусмотрены подсказки в исправлении кода;

- быстрый поиск и исправление ошибок – улучшен процесс отладки и тестирования приложения, улучшен помощник по исправлению и выявлению новых ошибок;

- интеграция с облачной системой – встроенные средства программного продукта гарантируют полную интеграцию с рядом таких приложений как: Azure, .NET Core, контейнерами Docker;

- эффективная совместная работа – возможно управление командными проектами, которые размещены у поставщиков (Visual Studio Team Services, Team Foundation Server либо Git Hub), при помощи функции «Открыть папку» возможно открытие практически любого файла;

- реализация качественных мобильных приложений – при помощи усовершенствования отладки Xamarin и других средств существенно ускоряется и упрощается процесс сборки, подключения и настройки интегрированных мобильных приложений для платформ Android, iOS и Windows. Кроме этого возможно создание мобильных приложений при помощи Apache Cordova либо создание кроссплатформенных библиотек C++;

- улучшение языка программирования – развитие поддержки новых функций языков (C#, C++, Visual Basic, Java Script) на протяжении всего процесса разработки;

- оптимизация производительности – улучшены основные применяемые функции, в результате чего уменьшено время загрузки, размер используемой памяти и увеличена скорость отклика во всем цикле разработки приложения;

- доступ к лицензиям программного обеспечения.

Под Visual Studio понимается интегрированная среда разработки (иначе IDE), разработанная фирмой Microsoft, а также является главным средством разработки Windows-приложений для платформ .NET и Windows. В Visual

Studio приложения создаются на языках C, C++, C#, F#, VB.NET, а также на дополнительных языках Python и Ruby.

Обоснование выбора языка программирования C# WPF. C# (иначе CSharp) – это высокоуровневый язык программирования, разработанный фирмой Microsoft для платформы Microsoft.NET. В нем собраны все основные преимущества различных языков программирования. Скорость исполнения приближается к языку Assembler [4, 36].

Объектно-ориентированный язык программирования C# включает в себя более 300 000 библиотек различного функционала, которые работают с максимальным быстродействием.

Компилятор C Sharp включен в стандартную разработку платформы .NET, исходя из этого программы на нем можно создавать и компилировать, не имея инструментальных средств вроде Visual Studio.

Внешне язык программирования C Sharp довольно сильно напоминает такие языки как C++ и Java, однако по организации среды выполнения идентичен лишь с Java, а компилятором языка C++ имеет очень мало общего. Главной характерной чертой языка программирования C Sharp является возможность запуска программ без мгновенной компиляции в машинный код, в отличие от C++. Все действия, производимые над языком программирования, выполняются под управлением виртуальной машины Common Language Runtime (CLR) [9].

Особенностями объектно-ориентированного языка программирования C Sharp являются [16, 33]:

- полный и хорошо определенный набор главных типов данных;
- интегрированная поддержка автоматической генерации XML-документов, соответственно освобождение занимаемой памяти;
- вероятность отметки классов и способов атрибутами, определяемыми пользователями;
- полный доступ к библиотеке классов .NET, простой доступ к набору базовых функций Windows API;

- при необходимости осуществляется прямой доступ к памяти персонального компьютера;

- возможность применения C Sharp для написания динамических web-страниц.

Одной из сфер, в которой не используется данный язык программирования, считаются ограниченные по времени и высокопроизводительности программы, а также когда имеет значение объем занимаемый исполнением цикла от 1000 до 1050 машинных циклов, для которых освобождение ресурсов потребуется немедленно. В настоящее время из языков низкого уровня больше всего подходит под данные критерии C++. В C# отсутствует ряд главных моментов, нужных для создания высокопроизводительных приложений.

После выбора средств реализации приложения «ТекроCRM» можно переходить непосредственно к реализации приложения, «заточенного» под особенности предприятия ООО «Текро».

ГЛАВА 3. РЕАЛИЗАЦИЯ CRM ПРИЛОЖЕНИЯ ДЛЯ КОМПАНИИ ООО «ТЕКРО»

3.1 Создание интерфейса программного продукта

Для реализации CRM-приложения, предназначенного для организации ООО «Текро», был использован программный продукт Microsoft Visual Studio Enterprise 2017. При помощи данного продукта в системе было реализовано основное окно приложения, включающее в себя меню с ссылками на модули системы, такие как: документы, отгрузки, клиенты, анализ, настройки.

Основное окно CRM приложения для организации ООО «Текро» показано на рис. 3.1.

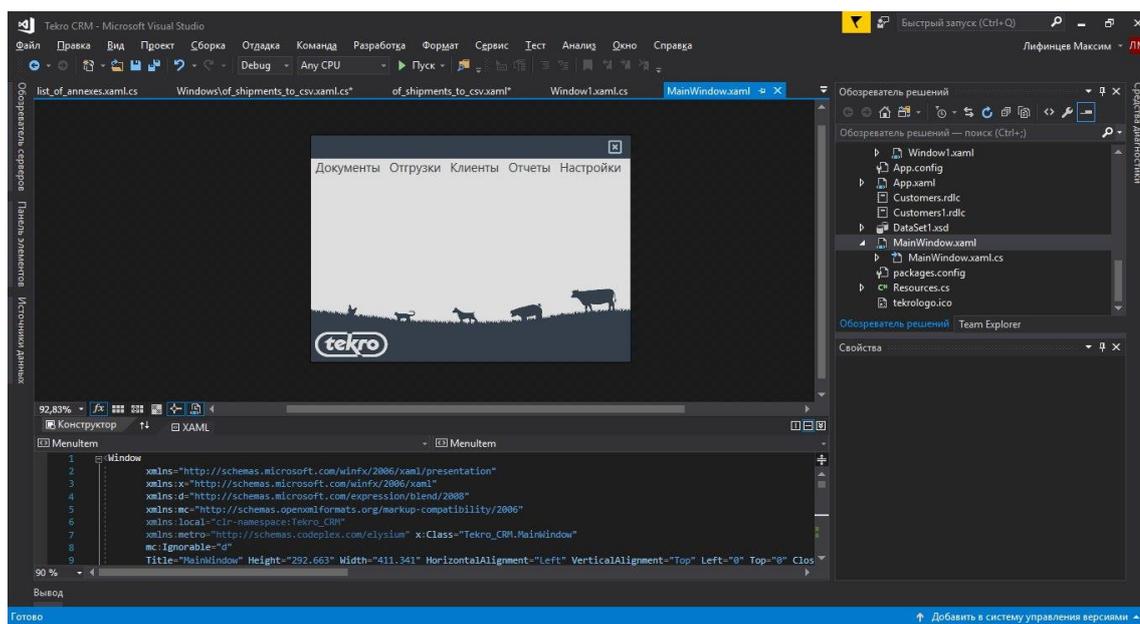


Рис. 3.1. Основное окно CRM-приложения

Для перехода между модулями системы был использован элемент управления «Main Menu». Данный элемент служит главным меню, хранящим в себе ссылки на модули приложения.

На рис. 3.2 представлено использование элемента управления «Main Menu».

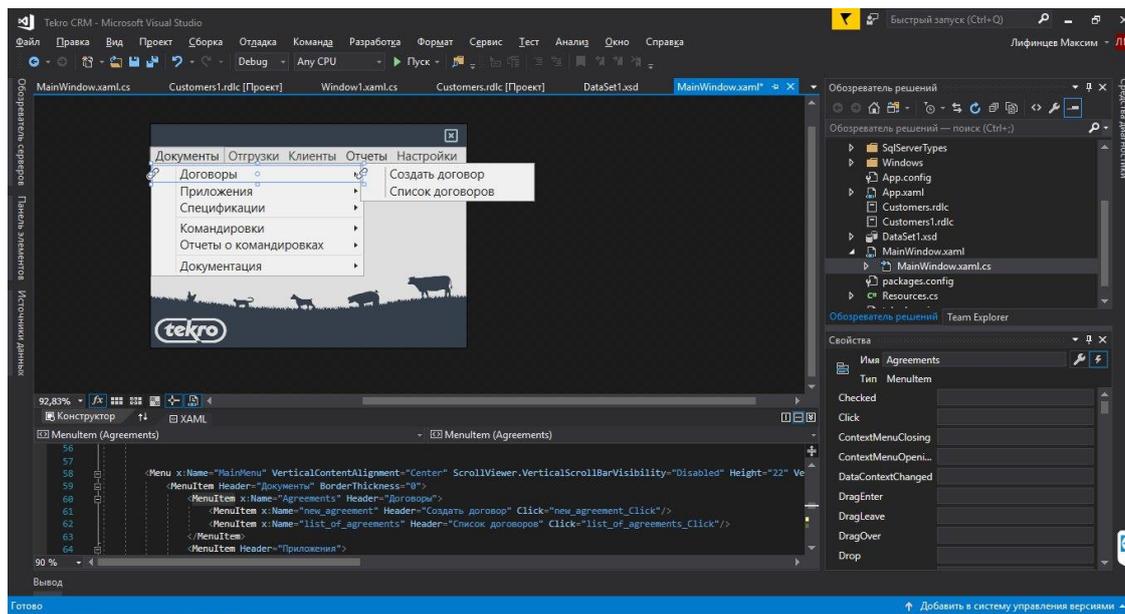


Рис. 3.2. Элемент управления «MainMenu»

Модуль CRM приложения «Создание договора» содержит в себе такие элементы как:

- Button – служит для выполнения событий при нажатии на кнопку;
- Text Box – служит для ввода информации в приложение;
- Date Picker – служит для выбора дат;
- Label – служит для вывода необходимого текста;
- Group Box – служит для группировки элементов управления.

На рис. 3.3 представлен модуль системы «Создание договора».

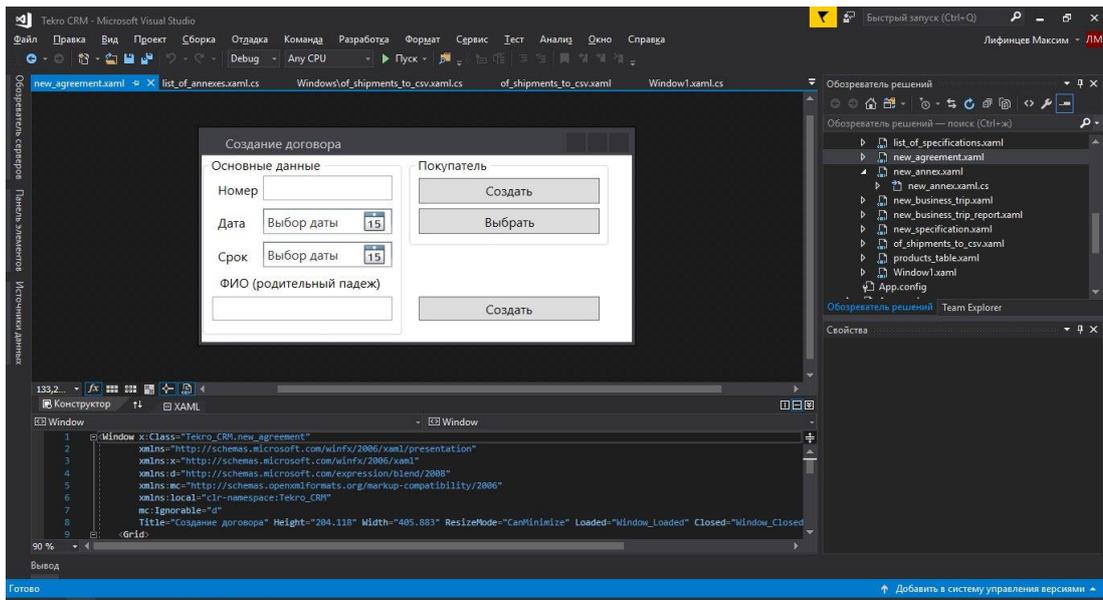


Рис. 3.3. Модуль «Создание договора»

Для создания приложения к договору был спроектирован модуль «Создание приложения». Данный модуль включает в себя те же элементы, что и модуль «Создание договора», а также Tab Control, который служит для перехода между вкладками, которые содержат элементы Text Box для ввода данных.

На рис. 3.4 показан модуль системы «Создание приложения».

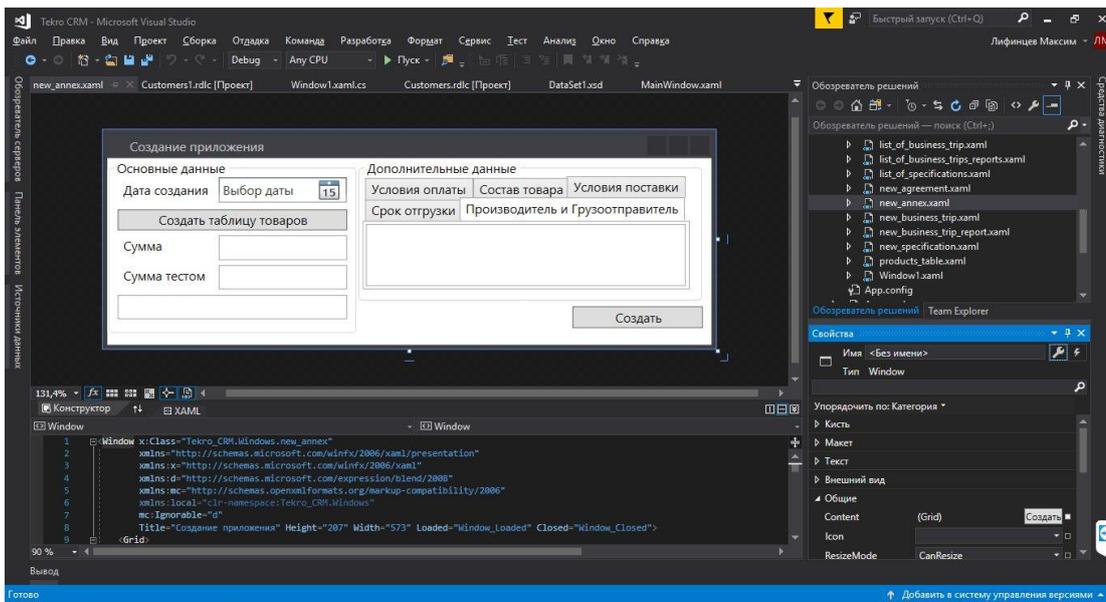


Рис. 3.4. Модуль «Создание приложения»

Для вывода необходимой информации и удобного экспорта данных используются отчеты. Данные шаблоны содержат переменные для вывода содержимого базы данных. Он содержит таблицу и переменную для учета нумерации страниц.

На рис. 3.5 показан процесс создания шаблона отчета.

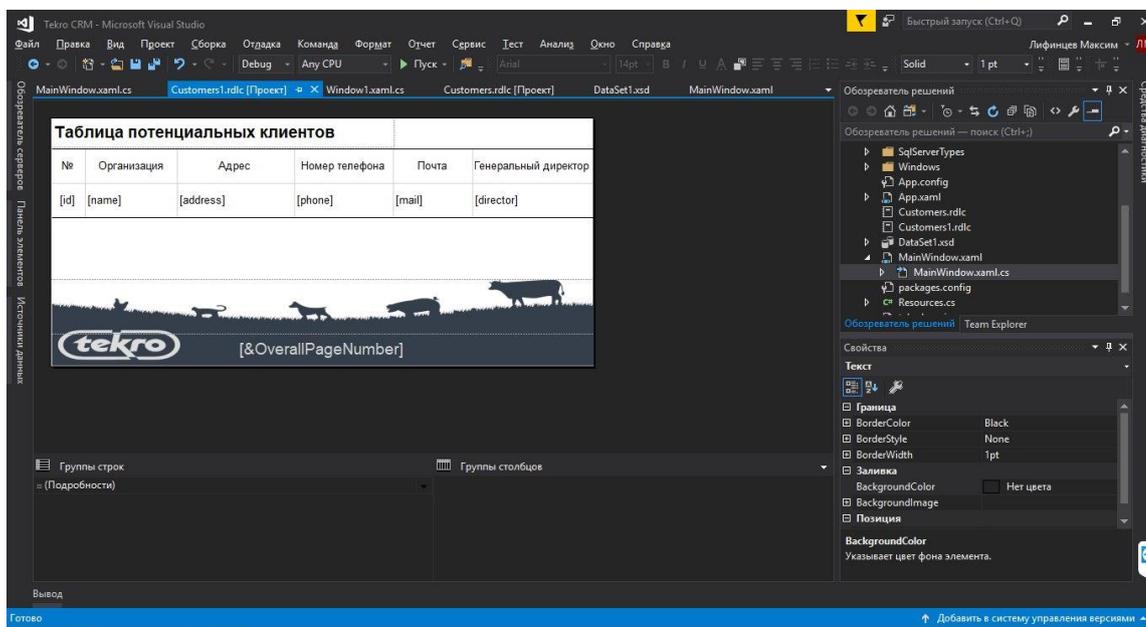


Рис. 3.5. Создание шаблона отчета

После создания интерфейса приложения «ТекроCRM» можно переходить к программной разработке модулей системы.

3.2 Программная разработка модулей CRM-приложения

После завершения разработки интерфейса программного продукта необходимо приступить к программной реализации модулей CRM приложения.

При запуске системы производится проверка на наличие файла конфигурации. В случае если файл не найден, открывается окно настроек, в котором пользователь задает параметры CRM приложения. После сохранения которых создается конфигурационный файл, а также все необходимые каталоги.

Программный код на проверку наличия файла конфигурации представлен в листинге 3.1.

Листинг 3.1. Проверка на наличие файла конфигурации

```
PrivatevoidWindow_Loaded(objectsender, RoutedEventArgse)
{ if (System.IO.File.Exists("Config.txt")) {
}
else
    {MessageBox.Show("Это первый запуск программы, необходимо создать файл
настроек", "Уведомление");
create_configcf = newcreate_config();
cf.ShowDialog();
string directory;
StreamReader file = new StreamReader("Config.txt");
directory = file.ReadLine() + @"\templates";
ZipFile.ExtractToDirectory("Templates.zip",directory);}}
```

Так как C# объектно-ориентированный язык программирования, то и все его элементы представлены классами и объектами классов. В данном CRM – приложении все окна являются объектами классов, которые создаются динамически.

Программный код объявления окна приложения и его отображения представлен в листинге 3.2.

Листинг 3.2. Создание объекта окна приложения

```
new_agreementnew_agreement = newnew_agreement();
new_agreement.Show();
```

В «ТекроCRM» необходимым функционалом является создание документов Word. Для каждого типа создаваемого документа имеется свой шаблон, который изменяется программно.

Программный код создания документа Word из шаблона на примере Договора представлен в листинге 3.3.

Листинг 3.3. Создание документа Word на примере Договора

```
templates temps = newtemplates();
string directory;
StreamReader file = newStreamReader("Config.txt");
    directory = file.ReadLine() + @"\templates";
temps.agreement_template = directory + @"\ + "agreement_template.dotx";
file.Close();
```

Листинг 3.3. Продолжение

```

stringdir_name;
dir_name = created_buyer.name;
StreamReaderfile_dir = newStreamReader("Config.txt");
Regexregex = newRegex(@"\W");
dir_name = regex.Replace(dir_name, "");
string path;
    path = file_dir.ReadLine() + @"\" + dir_name + @"\" + "Договоры\" +
date_picker1.SelectedDate.Value.Date.ToLongDateString();
Directory.CreateDirectory(path);
_Documentbusiness_trip_doc = GetDoc(temps.agreement_template);
business_trip_doc.SaveAs(path + @"\" + text_num.Text + ".docx");
stringdocumentation_id;
documentation_id = path + @"\" + text_num.Text + ".docx";
business_trip_doc.Close();

```

Для внесения данных в создаваемый документ Word используется диапазон (Range), в котором происходит замена текста. Простейшим методом определения диапазона является создание закладок (Bookmarks) в шаблоне Word. В дальнейшем созданные закладки меняются на необходимый текст.

Программный код замены закладок документа на необходимый текст представлен в листинге 3.4.

Листинг 3.4. Замена закладок документа

```

business_trip_doc.Bookmarks["address"].Range.Text = created_buyer.address;
business_trip_doc.Bookmarks["bank"].Range.Text = created_buyer.bank_name;
business_trip_doc.Bookmarks["bik"].Range.Text = created_buyer.bik;
business_trip_doc.Bookmarks["buyer"].Range.Text = created_buyer.name;
business_trip_doc.Bookmarks["buyer2"].Range.Text = created_buyer.name;

```

Для работы некоторых модулей CRM – приложения необходимо импортировать данные из Excel и обрабатывать их в программе. Для импорта также используется диапазон (Range), данные из которого сохраняются в массив объектов.

Программный код импорта данных из Excel в массив объектов представлен в листинге 3.5.

Листинг 3.5. Импорт данных из Excel в массив объектов

```

Microsoft.Office.Interop.Excel.Rangerange = ObjWorkSheet.get_Range("O27", "W" +
iLastRow.ToString());
values = (System.Array)range.Cells.Value;

```

В CRM-приложении необходима функция создания и заполнения таблиц в документах Word. Для этого используется диапазон ячеек, текст в которых заменяется программно.

Программный код заполнения ячеек таблицы Word представлен в листинге 3.6.

Листинг 3.6. Заполнение таблицы Word

```
for (inti = 2; i<items.Length + 2; i++)
    {s = items[i-2].ToString().Split(newchar[] { ';' });
_table.Cell(i, 1).Range.Text = s[0];
_table.Cell(i, 2).Range.Text = s[1];
_table.Cell(i, 3).Range.Text = s[2];
_table.Cell(i, 4).Range.Text = s[3];}
```

По завершению программной реализации всех модулей CRM-приложения можно переходить непосредственно к тестированию контрольного примера.

3.3 Тестирование контрольного примера

При первом запуске CRM приложения необходимо создать файл конфигурации, который создается из окна настроек. После заполнения необходимых полей, данные сохраняются в файл Config.txt.

На рис. 3.6. показано окно настроек приложения «ТекроCRM».

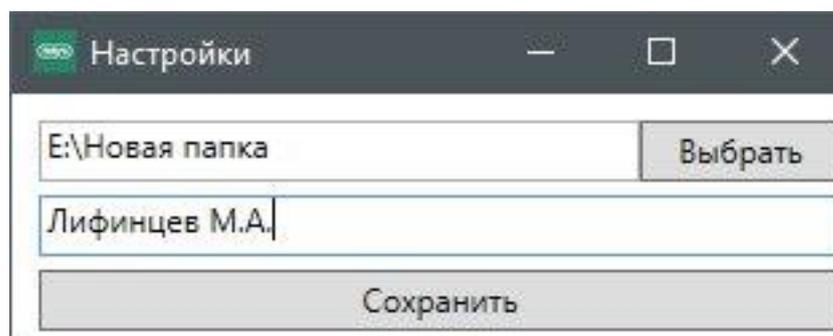


Рис. 3.6. Окно настроек приложения «ТекроCRM»

После сохранения настроек открывается главное окно программы, которое содержит ссылки на все модули приложения, такие как: документы, отгрузки, клиенты, анализ, настройки.

На рис. 3.7. показано основное окно приложения «ТекроCRM».

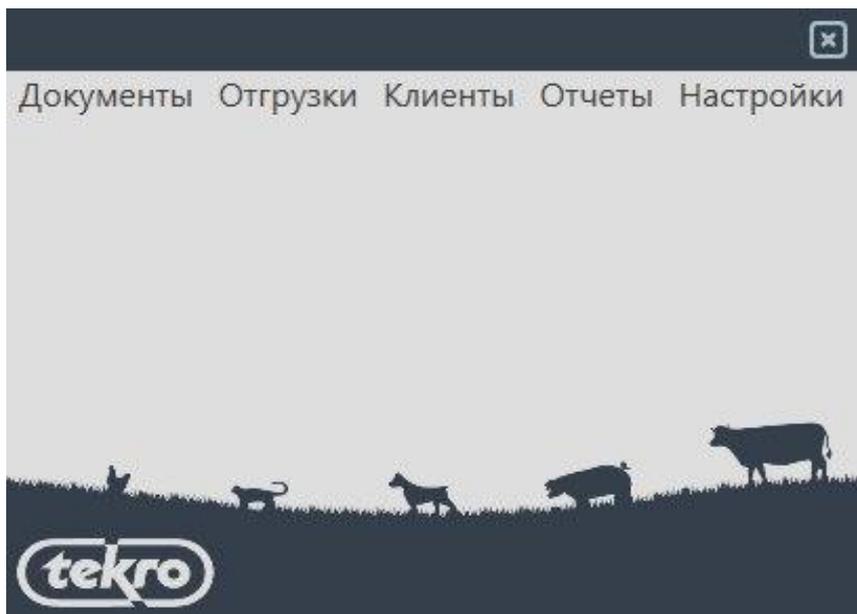


Рис. 3.7. Основное окно приложения «ТекроCRM»

Одним из модулей программы является модуль «Клиенты». В нем содержится информация о клиентах программы, а также средства для управления таблицей. В «ТекроCRM» клиенты разделены на «лиды» и контакты, которые отличаются цветом.

На рис. 3.8. показан модуль «Клиенты».

№	Название	Адрес	Ген. Директор	Контактное лицо	Контактная почта	Телефон	Факс	Почта
1	Малахов +	308009, г. Белгород, 5ый Заводской переулок, д.30	Антофьев Павел Валерьевич	Игрунов К.К.	chernov@td-malahov.ru	8 (4722) 34-57-48	8(4722)34-57-49	igrunov@mail.ru
2	Веста	308014, г. Белгород, улица Николая Чумичова, д.123	Шопска Лариса Анатольевна	Шопски В.Н.	shopska@yandex.ru	8(4722)32-01-02	8(4722)32-01-01	shopski@mail.ru
3	Левада	308009, г. Новый Оскол, улица Преображенская, д. 2	Константинов Владимир Анатольевич	Мишакин И. В.	konstantinov@mail.ru	8-910-564-78-92		misakin@yandex.ru
4	Новострой	314759, г. Новый Оскол, улица 1ая Котлозаводская, д. 45	Светлов Илья Игоревич	Семенов И. В.	semenov@yandex.ru	8(4722)95-78-32	8(4722)95-78-33	svetlov@yandex.ru
5	Титан	308004, г. Новомичуринск, улица Смирагина, д. 13	Корнеев Николай Николаевич	Корнеева Г. П.	korneeva@yandex.ru	8(4722)45-25-01	8(4722) 45-25-02	korneeva@yandex.ru
6	Зверюшки	304789, г. Крымск, улица Николая Чумичова, д.78	Чернов Анатолий Анатольевич	Ямшук Я. А.	yamshuk@mail.ru	8(4722)69-89-01	8(4722)69-89-02	chernov@mail.ru
7	Ветпитание	301259, г. Рязань, улица Рябиновая, д.71	Гольянова Юлия Николаевна	Вощкин А. А.	voskin@mail.ru	8-919-564-89-78		golyanova@mail.ru
8	ОптПлюс	308745, г. Москва, улица Первомайская, д.143	Егоров Петр Михайлович	Иванов С.В.	ivanov.s.v.75@yandex.ru	8-909-685-85-88	8-909-685-85-85	optplus@mail.ru
9	Дворянин	308015, г. Белгород, бульвар Народный, строение 2а	Никулин Владислав Петрович	Шаповалов И.И.	ivan.SH.19@google.com	8 (4722)35-15-15	8 (4722)35-15-16	nikulindvoryanin@gmail.com
10	Ферма	309110, Белгородская обл., п. Ивня, улица Советская, д.15	Мамаев Василий Викторович	Бабаев Е.А.	babaev2013@mail.ru	8 (4724)35-14-87	8 (4724)35-14-88	ivnyafarma@yandex.ru

Рис. 3.8. Модуль «Клиенты»

Для удобства в использовании данного модуля созданы «Фильтры». Одним из них является поиск.

На рис. 3.9 показано использование поиска в приложении «ТекроCRM».

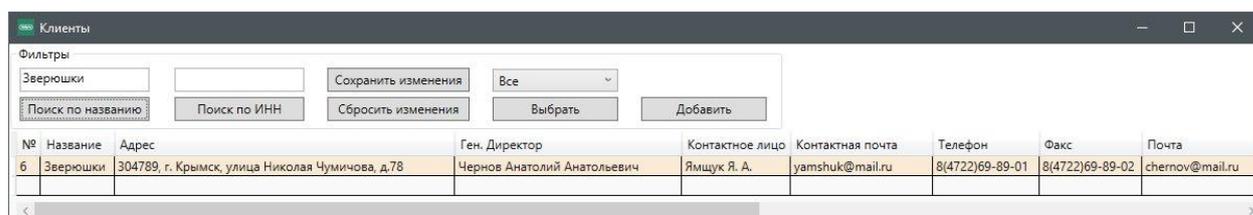


Рис. 3.9. Использование поиска в приложении «ТекроCRM»

Также с помощью фильтра можно выбрать тип клиента. После его применения отображается только один тип: «Лиды» либо Контакты.

На рис. 3.10 показан тип клиентов «Контакты».

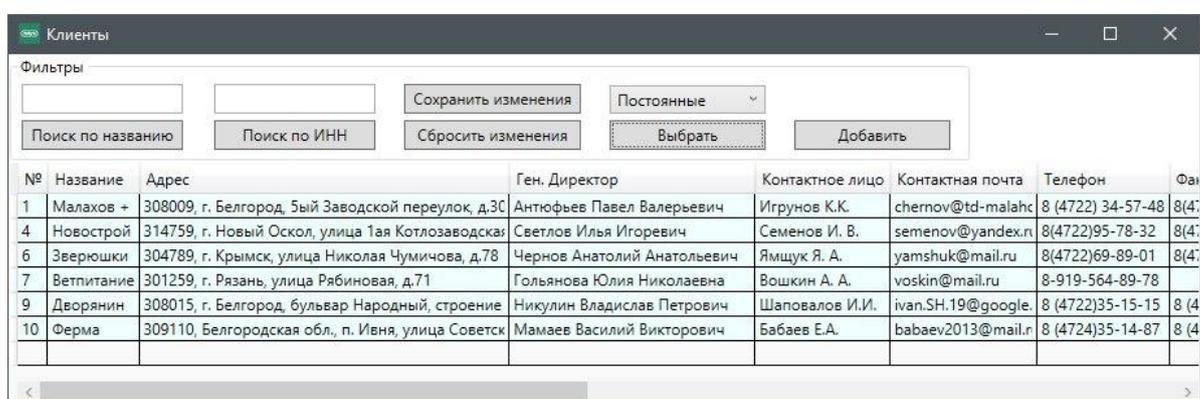


Рис. 3.10. Тип клиентов «Контакты»

В CRM приложении автоматизирована работа с документацией. Одним из главных документов является «Договор». Для создания которого пользователю достаточно указать даты и выбрать клиента. Результат создания договора показан на рисунке 3.1. в приложении 3.

На рис. 3.11 показан модуль создания договора.

Рис. 3.11. Модуль создания договора

После создания договора необходимо создать приложение к нему. В начале заполняется таблица товаров, и рассчитывается сумма. Эти данные передаются на форму создания приложения.

На рис. 3.12 показано окно создания таблицы товаров.

Рис. 3.12. Окно создания таблицы товаров

При вводе необходимой информации и заполнении таблицы товаров создается документ приложения. Кроме основных указываются дополнительные данные: условия оплаты, состав товара, условия поставки, срок отгрузки, производитель и грузоотправитель. Результат создания приложения показан на рисунке 3.2.в приложении 3.

На рис. 3.13 показано окно создания приложения.

Рис. 3.13. Окно создания приложения

Для приложения необходимо создать файл спецификации. Для этого выбирается файл рецепта и вводится его название. После нажатия кнопки «Создать» на диске создается файл спецификации к приложению. Результат создания спецификации показан на рисунке 3.3.в приложении 3.

На рис. 3.14. показано окно создания спецификации.

Рис. 3.14. Окно создания спецификации

Еще одним модулем CRM приложения является создание заявки на командировку. После заполнения необходимых данных создается документ, пример которого представлен на рис. 3.4 в приложении 3.

На рис. 3.15 показано окно создания заявки на командировку.

Рис. 3.15. Окно создания заявки на командировку

После завершения командировки необходимо создать отчет. Отчет включает в себя: цель командировки, вывод и результат. После заполнения всех данных создается документ отчета на диске, пример которого представлен на рис. 3.5 в приложении 3.

На рис. 3.16 показано окно создания отчета о командировке.

Рис. 3.16. Окно создания о командировке

Для необходимости наглядного вывода данных о клиентах с последующей отправкой на печать используются отчеты. Отчеты поддерживают экспорт в таких форматах: pdf, docx, xlsx.

На рис. 3.17 показан процесс создания отчета «Потенциальные клиенты».

№	Организация	Адрес	Номер телефона	Почта	Генеральный директор
2	Веста	308014, г. Белгород, улица Николая Чумичова, д. 123	8(4722)32-01-02	shopski@mail.ru	Шопска Лариса Анатольевна
3	Левада	308009, г. Новый Оскол, улица Преображенская, д. 2	8-910-564-78-92	misakin@yandex.ru	Константинов Владимир Анатольевич
5	Титан	308004, г. Новомичуринск, улица Смирягина, д. 13	8(4722)45-25-01	korneeva@yandex.ru	Корнеев Николай Николаевич
8	ОптПлюс	308745, г. Москва, улица Первомайская, д. 143	8-909-685-85-88	optplus@mail.ru	Егоров Петр Михайлович

Рис. 3.17. Создание отчета «Потенциальные клиенты»

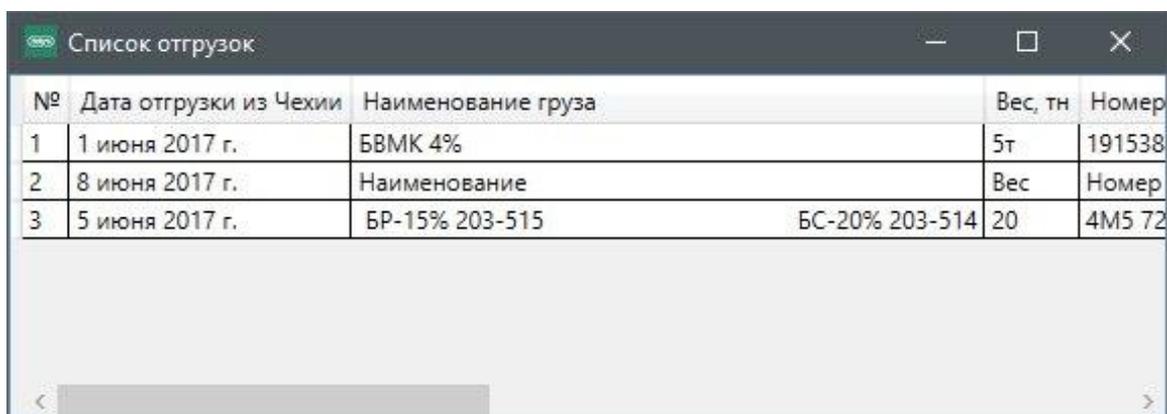
Также в программе реализован модуль отгрузок. Для записи данных в базу используется окно «Создание отгрузки», в котором вводится необходимая информация и выбирается получатель.

На рис. 3.18 показано окно создания отгрузки.

Рис. 3.18. Окно создания отгрузки

После создания отгрузки можно посмотреть записанные в базу данные. Для этого используется окно «Список отгрузок».

На рис.3.19 показано окно «Список отгрузок».



№	Дата отгрузки из Чехии	Наименование груза	Вес, тн	Номер
1	1 июня 2017 г.	БВМК 4%	5т	191538
2	8 июня 2017 г.	Наименование	Вес	Номер
3	5 июня 2017 г.	БР-15% 203-515	БС-20% 203-514	20

Рис. 3.19. Окно «Список отгрузок»

Все отгрузки можно экспортировать в файл CSV. Для этого выбирается необходимая папка, после чего все данные сохраняются в файл. Созданный файл можно открыть в Excel для наглядного представления. Пример файла, созданного в программе представлен на рис. 3.6 в приложении 3.

На рис. 3.20 показано окно экспорта.

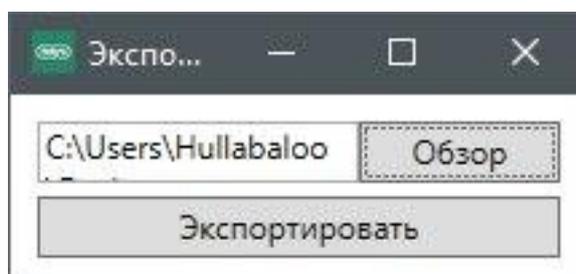


Рис. 3.20. Окно экспорта

Таким образом, внедрение разработанного приложения «ТекроCRM» позволит упростить работу с клиентами, а также автоматизировать документооборот предприятия ООО «Текро» в целом.

ЗАКЛЮЧЕНИЕ

В ходе написания выпускной квалификационной работы было спроектировано и разработано CRM – приложение для отдела продаж компании ООО «Текро».

Для решения данной задачи была подробно изучена деятельность предприятия ООО «Текро» и его структура, обоснована необходимость разработки программного продукта, а также проведен обзор современных информационных систем, и выявлены их недостатки.

Для проектирования и разработки CRM-приложения для организации ООО «Текро» были использованы следующие инструменты: SQLite, среда .NET и язык программирования C#, а также среда разработки Visual Studio 2017.

В результате внедрения разработанного CRM-приложения автоматизируется деятельность сотрудников отдела продаж, сократится время доступа и получения документов.

Разработанная автоматизированная система внедрена в практическую деятельность ООО «Текро» г. Белгорода, о чем имеется акт о внедрении.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 2.105. Общие требования к текстовым документам, 2014 г. – 2 с.
2. ГОСТ 34.601-90 Автоматизированные системы. Стадии создания.
3. ГОСТ 7.32. Отчет о научно-исследовательской работе. Структура и правила оформления, 2014 г. -22 с.
- 4.Абрамян, М. М. Visual С# на примерах (+ CD-ROM)/М. М. Абрамян. - Москва: БХВ-Петербург, 2012. - 496 с.
5. Агальцов, В. П. Базы данных. Распределенные и удаленные базы данных: Учебник/В. П. Агальцов. – Москва: ИД ФОРУМ: НИЦ Инфра, 2013. – 272 с.
6. Битрикс 24 – сервис для автоматизации бизнеса [Электронный ресурс]. – Режим доступа: <https://www.bitrix24.ru> (дата обращения 01.06.2017г).
7. Бухалков, М. И. Организация производства и управление предприятием: Учебник /М.И. Бухалков . - Москва: ИНФРА, 2013. - 506 с.
8. Варфоломеева, А. О. Информационные системы предприятия: Учебное пособие/А. О. Варфоломеева, А. В. Коряковский, В. П. Романов. - Москва: НИЦ ИНФРА, 2013. - 283 с.
9. Васильев, А. А. С#. Объектно-ориентированное программирование / А. А. Васильев. - Москва: Питер, 2012. - 320 с.
10. Васильков, А. В. Информационные системы и их безопасность: Учебное пособие/А. В. Васильков, А. А. Васильков, И. А. Васильков. - Москва: Форум, 2013. - 528 с.
11. Вдовин, В. М. Предметно-ориентированные экономические информационные системы: Учебное пособие/В. М. Вдовин, Л. Е. Суркова. - Москва: Дашков и К, 2016. - 388 с.

12. Вертоградов, В. К. Развертывание CRM-системы как рациональный первый шаг в постановке маркетинга [Электронный ресурс]/В. К. Вертоградов – Режим доступа: <http://www.pro-invest.com/it> (дата обращения 13.05.2017)

13. Гаврилов, Л. П. Управление предприятием: финансовые и инвестиционные решения: Учебное пособие/Л. П. Гаврилов. - Москва: Финансы и статистика, 2012. - 184 с.

14. Гвоздева, В. А. Информатика, автоматизированные информационные технологии и системы: Учебник/В. А. Гвоздева. - Москва: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 544 с.

15. Грекул, В. И. Управление внедрением информационных систем: учебник [Электронный ресурс]/В. И. Грекул, Н. Л. Коровкина, Г. Н. Денищенко. – Режим доступа: <http://www.intuit.ru/department/itmngt/isimman> (дата обращения: 13.05.2017)

16. Гриффитс, Иэн Программирование на C# 5.0/И. Гриффитс. - Москва, 2012. - 826 с.

17. Гукова, А. В. Управление предприятием: финансовые и инвестиционные решения: Учебное пособие/А. В. Гукова. - Москва: Финансы и статистика, 2012. –184 с.

18. Данелян, Т. Я. Экономические информационные системы (ЭИС) предприятий и организаций: Монография./Т. Я. Данелян. - Москва: ЮНИТИ, 2015. - 284 с.

19. Джеффри, Рихтер CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C#/Р. Джеффри. - Москва, 2016. - 896 с.

20. Емельянов, С. В. Информационные технологии и вычислительные системы: ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ. ПРИКЛАДНЫЕ АСПЕКТЫ ИНФОРМАТИКИ/С. В. Емельянов. - Москва: Ленанд, 2015. - 96 с.

21. Зайцев, Н. Л. Экономика, организация и управление предприятием: Учебное пособие /Н.Л. Зайцев. - Москва: ИНФРА-М, 2012. – 455 с.

22. Зиборов, В. В. Visual C# 2012 на примерах/В. В. Зиборов. - Москва: БХВ-Петербург, 2013. - 480 с.

23. Зиборов, В. С. MS Visual C++ 2010 в среде .NET/В. С. Зиборов. - Москва, 2012. - 320 с.

24. Кадыров, М. А. Особенности внедрения CRM при массовых продажах [Электронный ресурс]/М. А. Кадыров – Режим доступа: <http://www.cfin.ru/> (дата обращения 13.05.2017)

25. Курочкин, Д. В. Логистика: [транспортная, закупочная, производственная, распределительная, складирования, информационная]: курс лекций / Д. В. Курочкин. – Минск: ФУАинформ, 2012. – 268 с.

26. Леонтьев, В. С. "Windows 10 Новейший самоучитель"/В. С. Леонтьев. - Эксмо, 2015 год, 528 с.

27. Логическое проектирование [Электронный ресурс]. – Режим доступа: http://life-prog.ru/1_5589_lektsiya-.html (дата обращения 01.06.2017г).

28. Логическое проектирование баз данных [Электронный ресурс]. – Режим доступа: <http://www.studfiles.ru/preview/2152676/page:7/> (дата обращения 01.06.2017г).

29. Олейник, П. П. Корпоративные информационные системы: Учебник для вузов. Стандарт третьего поколения /П. П. Олейник. - СПб: Питер, 2012. - 176 с.

30. ООО «Текро» [Электронный ресурс]. – Режим доступа: <http://www.tekro-rf.ru> (дата обращения 23.05.2017г).

31. Петряшов, Д. А. Автоматизация работы с информацией о внешнем окружении фирмы /Д. А. Петряшов – 2013. – № 2. –138–145 с.

32. Погонин, В. А., Корпоративные информационные системы: учебное пособие/В. А. Погонин, А. Г. Схиртладзе, С. И. Татаренко, С. Б. Путин. – Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2012. – 144 с.

33. Подбельский, В. В. Язык C#. Базовый курс/В. В. Подбельский. - Москва: Финансы и статистика, 2013. - 408 с.

34. Рыжко, А. Л. Информационные системы управления производственной компанией: Учебник для академического бакалавриата / А. Л. Рыжко, А. И. Рыбников, Н. А. Рыжко. - Люберцы: Юрайт, 2016. - 354 с.

35. Сырецкий, Г. А. Информатика. Фундаментальный курс. Том II. Информационные технологии и системы /Г. А. Сырецкий. - СПб: ВHV, 2012. - 848 с.

36. Троелсен, Эндрю "Язык программирования C# 5.0 и платформа .NET 4.5"/ Э.Троелсен. - Москва, 2013 год. - 1311 с.

37. Уткин, В. Б. Информационные системы в экономике: Учебник для студентов высших учебных заведений/В. Б. Уткин, К. В. Балдин. - Москва: ИЦ Академия, 2012. - 288 с.

38.Физическое проектирование базы данных [Электронный ресурс]. – Режим доступа: <http://www.bourabai.kz/dbt/dbms/03.htm> (дата обращения 01.06.2017г).

39. Физическое моделирование. Особенности построения физической модели базы данных [Электронный ресурс]. – Режим доступа: http://studme.org/77262/informatika/izicheskoe_modelirovanie (дата обращения 01.06.2017г).

40. Фримен, Адам ASP.NET 4.5 с примерами на C# 5.0 для профессионалов; Диалектика / А. Фримен. - Москва, 2014. - 686 с.

41. Что такое SQLite. Плюсы и минусы. Заметки web-разработчика [Электронный ресурс]. – Режим доступа: <http://webnotes.by/docs/php/259> (дата обращения 01.06.2017г).

42. Шарп, Джон " MicrosoftVisual C#"/ Д. Шарп. – Питер, 2017 год, 848 с.

43. Ширяев, В.И. Управление предприятием: Моделирование, анализ, управление / В. И. Ширяев, И. А. Баев, Е. В. Ширяев. - Москва: КД Либроком, 2015. –272 с.

44. Эспозито, Дино Программирование на основе Microsoft ASP.NET MVC/Д. Эспозито. - Москва, 2012. - 275 с.

45. Яковлев, В. П. Корпоративные информационные системы: конспект лекций / В. П. Яковлев. - СПбГТУРП. – СПб., 2015. – 117 с.

46. 1С: Предприятие 8 CRMСТАНДАРТ [Электронный ресурс]. – Режим доступа: <http://center-comptech.ru/1c-crm-standart.html> (дата обращения 01.06.2017г).

47. CRM MANGO OFFICE – облачная CRM – система для бизнеса [Электронный ресурс]. – Режим доступа: <https://www.mango-office.ru/products/crm/> (дата обращения 01.06.2017г).

48. CRM-система для маркетинга, продаж, сервиса от Terrasoft [Электронный ресурс]. – Режим доступа: <https://www.terrasoft.ru> (дата обращения 01.06.2017г).

49. Windows 10 [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Windows_10 (дата обращения 23.05.2017г).

ПРИЛОЖЕНИЕ 1

Окно MainWindow

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Microsoft.Office.Interop.Excel;
using Microsoft.Office.Interop.Word;
using System.Data.SQLite;
using System.Data;
using Tekro_CRM.Windows;
using System.IO;
using System.IO.Compression;
namespace Tekro_CRM
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public class database {
        public String dbFileName;
        public SQLiteConnection dbConnection = new SQLiteConnection();
        public SQLiteCommand sqlCommand = new SQLiteCommand();
        public DataSet DataSet = new DataSet();
        public System.Data.DataTable DataTable = new System.Data.DataTable();
        public SQLiteDataAdapter DataAdapter;
        public string returnstr(string sql, int index) {
            dbFileName = MainWindow.dbname;
            dbConnection = new SQLiteConnection("Data Source=" + dbFileName + ";Version=3;");
            dbConnection.Open();
            sqlCommand.Connection = dbConnection;
            sqlCommand.CommandText = sql;
            DataAdapter = new SQLiteDataAdapter(sqlCommand.CommandText, dbConnection);
            DataSet.Reset();
            DataAdapter.Fill(DataTable);
            string toreturn;
            toreturn = DataTable.Rows[0][index].ToString();
            dbConnection.Close();
            return toreturn; } }
        public string agreement_template;
        public string business_trip_template; }

```

```

public class buyer {
public string name, director, address, phone, inn, kpp, ogrn, bank_name, rsch, korsch, bik,
id,okpo,fax; }
public partial class MainWindow : System.Windows.Window {
public static string dbname = "Sources/database.db";
public static templates templates = new templates();
public MainWindow() {
InitializeComponent(); }
private void new_agreement_Click(object sender, RoutedEventArgs e) {
new_agreement new_agreement = new new_agreement();
new_agreement.Show(); }
private void list_of_agreements_Click(object sender, RoutedEventArgs e) {
list_of_agreement list_of_agreements = new list_of_agreement();
list_of_agreements.Show();}
private void new_business_trip_Click(object sender, RoutedEventArgs e) {
new_business_trip new_business_trip = new new_business_trip();
new_business_trip.Show();}
private void list_of_business_trips_Click(object sender, RoutedEventArgs e){
list_of_business_trip list_of_business_trip = new list_of_business_trip();
list_of_business_trip.Show();}
App.Current.Shutdown();}
private void list_of_business_trips_reports_Click(object sender, RoutedEventArgs e){
list_of_business_trips_reports b_trips_rep = new Windows.list_of_business_trips_reports();
b_trips_rep.Show();}
private void buyers_Click(object sender, RoutedEventArgs e){
buyers window_buyer = new buyers();
window_buyer.Show();}
ButtonStyle btnstyle = new ButtonStyle();
private void Window_Loaded(object sender, RoutedEventArgs e){
if (File.Exists("Config.txt")){ }
else{
MessageBox.Show("Это первый запуск программы, необходимо создать файл
настроек", "Уведомление");
create_config cf = new create_config();
cf.ShowDialog();
string directory;
StreamReader file = new StreamReader("Config.txt");
directory = file.ReadLine() + @"\templates";
ZipFile.ExtractToDirectory("Templates.zip",directory);} }
this.Close();}
private void Window_MouseLeftButtonDown(object sender, MouseButtonEventArgs e){
this.DragMove();}
private void Close_btn_MouseMove(object sender, MouseEventArgs e) {
btnstyle.SelectStyle(Close_btn, "pack://siteoforigin:,,,/Resources/Close_white.png");}
private void Close_btn_MouseLeave(object sender, MouseEventArgs e)
{btnstyle.SelectStyle(Close_btn, "pack://siteoforigin:,,,/Resources/Close.png");}
private void settings_Click(object sender, RoutedEventArgs e){
create_config cf = new create_config();cf.Show();}
private void open_annexes_Click(object sender, RoutedEventArgs e)
{list_of_annexes loa = new list_of_annexes();loa.Show();}
private void list_of_specifications_Click(object sender, RoutedEventArgs e)
{list_of_specifications los = new list_of_specifications();}

```

```

los.Show();}
private void report_Click(object sender, RoutedEventArgs e){}
private void report_customers_Click(object sender, RoutedEventArgs e)
{Window1 w1 = new Window1();
w1.report = "customers";w1.Show();}
private void report_customers1_Click(object sender, RoutedEventArgs e)
{Window1 w1 = new Window1();w1.report = "customers1";
w1.Show();}
private void to_csv_Click(object sender, RoutedEventArgs e)
{of_shipments_to_csv to_csvw = new of_shipments_to_csv();to_csvw.Show();}
private void add_os_Click(object sender, RoutedEventArgs e)
{new_of_shipment nos = new new_of_shipment();
nos.Show();}
private void list_of_os_Click(object sender, RoutedEventArgs e)
{list_of_shipments los = new list_of_shipments();los.Show();}}

```

Окно buyer_create

```

using System;
using System.Collections.Generic;
using System.Data.SQLite;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using Tekro_CRM.Windows;
namespace Tekro_CRM.Windows
{/// <summary>
/// Логика взаимодействия для buyer_create.xaml
/// </summary>
/// public partial class buyer_create : Window
{public buyer_create(){InitializeComponent();}
public string Data()
{return "";}
void ClearTextBoxes(DependencyObject obj)
{for (int i = 0; i < VisualTreeHelper.GetChildrenCount(obj); i++)
{if (obj is TextBox)
{((TextBox)obj).Text = null;}
ClearTextBoxes(VisualTreeHelper.GetChild(obj, i));}}
private void clear_Click(object sender, RoutedEventArgs e)
{ClearTextBoxes(this);}

```

```

public buyer created_buyer = new buyer();
public void editgroupname(GroupBox name, TextBox name2)
{ name.Header = "Компания " + created_buyer.name;
name2.Text = created_buyer.director; }
private void create_Click(object sender, RoutedEventArgs e)
{ created_buyer.name = name.Text;
created_buyer.address = address.Text;
created_buyer.bank_name = bank.Text;
created_buyer.bik = bik.Text;
created_buyer.director = director.Text;
created_buyer.inn = inn.Text;
created_buyer.korsch = korsch.Text;
created_buyer.kpp = kpp.Text;
created_buyer.ogrn = ogrn.Text;
created_buyer.phone = phone.Text;
created_buyer.rsch = rsch.Text;
created_buyer.fax = fax.Text;
MessageBoxResult res = new MessageBoxResult();
res = MessageBox.Show("Сохранить данные о покупателе?", "Сохранение",
MessageBoxButton.YesNo,
MessageBoxImage.Question);
if(res == MessageBoxResult.Yes)
{ database db = new database();
db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText =
"insert into Customers" +
" ('name','address','phone','director','bank_name','rsch','korsch','bik','inn','kpp','ogrn','okpo','fax')" +
" values ('" + created_buyer.name + "', '" + created_buyer.address + "', '" + created_buyer.phone +
"', '" + created_buyer.director + "', '" + created_buyer.bank_name + "', '" + created_buyer.rsch +
"', '" + created_buyer.korsch + "', '" + created_buyer.bik + "', '" + created_buyer.inn + "', '" +
created_buyer.kpp + "', '" + created_buyer.ogrn + "', '" + created_buyer.okpo + "', '" +
created_buyer.fax + "')";
db.sqlCommand.ExecuteNonQuery();
db.sqlCommand.CommandText = "SELECT id FROM Customers WHERE
rowid=last_insert_rowid()";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
created_buyer.id = db.DataTable.Rows[0][0].ToString();
db.dbConnection.Close();
StreamReader file = new StreamReader("Config.txt");
string path = file.ReadLine();
string name = created_buyer.name;
Regex regex = new Regex(@"\W");
name = regex.Replace(name, "");
string direct = path + "\\" + name;
Directory.CreateDirectory(direct);
file.Close();this.Close();}else
{ this.Close();} }

```

Окно buyer_select

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SQLite;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{
    /// <summary>
    /// Логика взаимодействия для buyer_select.xaml
    /// </summary>
    public partial class buyer_select : Window
    {
        public buyer_select() { InitializeComponent(); }
        public database db = new database();
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            buyers_table.CanUserAddRows = false;
            buyers_table.CanUserDeleteRows = false;
            buyers_table.CanUserReorderColumns = false;
            buyers_table.CanUserResizeColumns = false;
            buyers_table.CanUserResizeRows = false;
            db.dbFileName = MainWindow.dbname;
            db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;
            db.sqlCommand.CommandText = "select * from Customers";
            db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
            db.DataSet.Reset();
            db.DataAdapter.Fill(db.DataTable);
            buyers_table.ItemsSource = db.DataTable.AsDataView();
            db.dbConnection.Close();
        }
        private void search_inn_Click(object sender, RoutedEventArgs e)
        {
            db.DataTable.Clear();
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;
            db.sqlCommand.CommandText = "select * from Customers where inn like '%" + inn.Text + "%'";
            db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
            db.DataSet.Reset();
            db.DataAdapter.Fill(db.DataTable);
            buyers_table.ItemsSource = db.DataTable.AsDataView();
            db.dbConnection.Close();
        }
    }
}

```

```

private void search_name_Click(object sender, RoutedEventArgs e)
{ db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "select * from Customers where name like '%" + name.Text +
"%";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
buyers_table.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void search_address_Click(object sender, RoutedEventArgs e)
{ db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "select * from Customers where address like '%" + address.Text
+ "%";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
buyers_table.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
public buyer created_buyer = new buyer();
public void editgroupname(GroupBox name, TextBox name2)
{ name.Header = "Компания " + created_buyer.name;
name2.Text = created_buyer.director;}
public database dbtosend = new database();
public string id_os = "", name_os;
public bool isos = false;
private void buyers_table_MouseDoubleClick(object sender, MouseButtonEventArgs e)
{ DataRowView row = (DataRowView)buyers_table.SelectedItems[0];
dbtosend.dbFileName = MainWindow.dbname;
dbtosend.dbConnection = new SQLiteConnection("Data Source=" + dbtosend.dbFileName +
";Version=3;");
dbtosend.dbConnection.Open();
dbtosend.sqlCommand.Connection = dbtosend.dbConnection;
dbtosend.sqlCommand.CommandText = "select * from Customers where id = '"+
row["id"].ToString() + "'";
dbtosend.DataAdapter = new SQLiteDataAdapter(dbtosend.sqlCommand.CommandText,
dbtosend.dbConnection);
dbtosend.DataSet.Reset();
dbtosend.DataAdapter.Fill(dbtosend.DataTable);
buyers_table.ItemsSource = dbtosend.DataTable.AsDataView();
created_buyer.name = dbtosend.DataTable.Rows[0].ItemArray[14].ToString();
created_buyer.id = dbtosend.DataTable.Rows[0].ItemArray[0].ToString();
created_buyer.address = dbtosend.DataTable.Rows[0].ItemArray[1].ToString();
created_buyer.bank_name = dbtosend.DataTable.Rows[0].ItemArray[7].ToString();
created_buyer.bik = dbtosend.DataTable.Rows[0].ItemArray[10].ToString();
created_buyer.director = dbtosend.DataTable.Rows[0].ItemArray[2].ToString();
created_buyer.inn = dbtosend.DataTable.Rows[0].ItemArray[11].ToString();
created_buyer.korsch = dbtosend.DataTable.Rows[0].ItemArray[9].ToString();
created_buyer.kpp = dbtosend.DataTable.Rows[0].ItemArray[12].ToString();

```

```

created_buyer.ogrn = dbtosend.DataTable.Rows[0].ItemArray[13].ToString();
created_buyer.phone = dbtosend.DataTable.Rows[0].ItemArray[5].ToString();
created_buyer.rsch = dbtosend.DataTable.Rows[0].ItemArray[8].ToString();
created_buyer.fax = dbtosend.DataTable.Rows[0].ItemArray[16].ToString();
created_buyer.okpo = dbtosend.DataTable.Rows[0].ItemArray[17].ToString();
if (isos){
id_os = dbtosend.DataTable.Rows[0].ItemArray[0].ToString();
name_os = dbtosend.DataTable.Rows[0].ItemArray[14].ToString();}
dbtosend.dbConnection.Close();
this.Close();}}}

```

Окно buyers

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SQLite;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows{
class AgeToColorConverter : IValueConverter{
public object Convert(object value, Type targetType, object parameter,
System.Globalization.CultureInfo culture){
return value.ToString() == "1" ?
new SolidColorBrush(Colors.Azure)
: new SolidColorBrush(Colors.AntiqueWhite);}
public object ConvertBack(object value, Type targetType, object parameter,
System.Globalization.CultureInfo culture){
throw new Exception("Не найдено необходимой функции.")}}
/// <summary>
/// Логика взаимодействия для buyers.xaml
/// </summary>
public partial class buyers : Window{public buyers()
{InitializeComponent();}
public database db = new database();
private void Window_Loaded(object sender, RoutedEventArgs e)
{db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "select * from Customers";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);

```

```

db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
datagrid.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void save_Click(object sender, RoutedEventArgs e)
{ db.dbConnection.Open();
db.sqlCommand.CommandText = string.Format("SELECT * FROM Customers",
db.DataTable.TableName);
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
SQLiteCommandBuilder builder = new SQLiteCommandBuilder(db.DataAdapter);
db.DataAdapter.Update(db.DataTable);
db.dbConnection.Close();}
private void reset_Click(object sender, RoutedEventArgs e)
{ db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "select * from Customers";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
datagrid.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
buyer buyer = new buyer();
private void create_agreement_Click(object sender, RoutedEventArgs e)
{ new_agreement window_new_agreement = new new_agreement();
DataRowView row = (DataRowView)datagrid.SelectedItems[0];
window_new_agreement.Show();
buyer.name = row["name"].ToString();
buyer.address = row["address"].ToString();
buyer.bank_name = row["bank_name"].ToString();
buyer.bik = row["bik"].ToString();
buyer.director = row["director"].ToString();
buyer.inn = row["inn"].ToString();
buyer.korsch = row["korsch"].ToString();
buyer.kpp = row["kpp"].ToString();
buyer.ogrn = row["ogrn"].ToString();
buyer.phone = row["phone"].ToString();
buyer.rsch = row["rsch"].ToString();
buyer.id = row["id"].ToString();
window_new_agreement.created_buyer = buyer;
window_new_agreement.group1.Header = window_new_agreement.group1.Header + " " +
window_new_agreement.created_buyer.name;
window_new_agreement.select_buyer.IsEnabled = false;
window_new_agreement.create_buyer.IsEnabled = false;
window_new_agreement.fio.Text = row["director"].ToString();}
private void name_Click(object sender, RoutedEventArgs e)
{ db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "select * from Customers where name like '%" + text_name.Text
+ "%'";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);

```

```

db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
datagrid.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void inn_Click(object sender, RoutedEventArgs e)
{ db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "select * from Customers where inn like '%" + text_inn.Text +
"%";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
datagrid.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void create_business_trip_Click(object sender, RoutedEventArgs e)
{ new_business_trip new_bt = new new_business_trip();
new_bt.date_picker3.SelectedDate = DateTime.Now;
DataRowView row = (DataRowView)datagrid.SelectedItems[0];
new_bt.created_buyer.address = row["address"].ToString();
new_bt.created_buyer.name = row["name"].ToString();
new_bt.created_buyer.id = row["id"].ToString();
new_bt.place_btn.IsEnabled = false;new_bt.Show();}
private void b_trips_Click(object sender, RoutedEventArgs e)
{ list_of_business_trip list_bt = new list_of_business_trip();
DataRowView row = (DataRowView)datagrid.SelectedItems[0];
list_bt.paramtrs = "and customers_id = " + row["id"].ToString();
list_bt.customer_search.IsEnabled = false;
list_bt.customer_text.IsEnabled = false;
list_bt.Show();}
private void agreements_Click(object sender, RoutedEventArgs e)
{ list_of_agreement list_a = new list_of_agreement();
DataRowView row = (DataRowView)datagrid.SelectedItems[0];
list_a.paramtrs = "and customer_id = " + row["id"].ToString();
list_a.id_search.IsEnabled = false;
list_a.id_text.IsEnabled = false;
list_a.Show();}
private void select_button_Click(object sender, RoutedEventArgs e)
{ db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
if (select_box.SelectedIndex == 0){
db.sqlCommand.CommandText = "select * from Customers";
to_customers.IsEnabled = true;
to_clients.IsEnabled = true;}
if (select_box.SelectedIndex == 1){
db.sqlCommand.CommandText = "select * from Customers where customer = 0 or customer is
null";
to_customers.IsEnabled = true;
to_clients.IsEnabled = false;}
if (select_box.SelectedIndex == 2)
{ db.sqlCommand.CommandText = "select * from Customers where customer = 1";

```

```

to_clients.IsEnabled = true;
to_customers.IsEnabled = false;}
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
datagrid.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void to_customers_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)datagrid.SelectedItems[0];
database db = new database();
db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "update Customers set customer = 1 where id = " +
row["id"].ToString();
db.sqlCommand.ExecuteNonQuery();
db.dbConnection.Close();
db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
if (select_box.SelectedIndex == 0){
db.sqlCommand.CommandText = "select * from Customers";
to_customers.IsEnabled = true;
to_clients.IsEnabled = true;}
if (select_box.SelectedIndex == 1){
db.sqlCommand.CommandText = "select * from Customers where customer = 0";
to_customers.IsEnabled = true;
to_clients.IsEnabled = false;}
if (select_box.SelectedIndex == 2)
{db.sqlCommand.CommandText = "select * from Customers where customer = 1";
to_clients.IsEnabled = true;
to_customers.IsEnabled = false;}
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
datagrid.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void to_clients_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)datagrid.SelectedItems[0];
database db = new database();
db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "update Customers set customer = 0 where id = " +
row["id"].ToString();
db.sqlCommand.ExecuteNonQuery();
db.dbConnection.Close();
db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;

```

```

if (select_box.SelectedIndex == 0){
db.sqlCommand.CommandText = "select * from Customers";
to_customers.IsEnabled = true;
to_clients.IsEnabled = true;}
if (select_box.SelectedIndex == 1)
{db.sqlCommand.CommandText = "select * from Customers where customer = 0";
to_customers.IsEnabled = true;
to_clients.IsEnabled = false;}
if (select_box.SelectedIndex == 2)
{db.sqlCommand.CommandText = "select * from Customers where customer = 1";
to_clients.IsEnabled = true;
to_customers.IsEnabled = false;}
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
datagrid.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void of_shipments_Click(object sender, RoutedEventArgs e)
{list_of_shipments los = new list_of_shipments();
DataRowView row = (DataRowView)datagrid.SelectedItems[0];
los.parametr = "and Customers.id = " + row["id"].ToString();
los.Show();} } }

```

Окно create_config

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using System.Windows.Forms;
using System.IO;
namespace Tekro_CRM.Windows
{
/// <summary>
/// Логика взаимодействия для create_config.xaml
/// </summary>
public partial class create_config : Window
{
public create_config()
{
InitializeComponent();
}
private void Button_Click(object sender, RoutedEventArgs e)
{
FolderBrowserDialog folder_dialog = new FolderBrowserDialog();
folder_dialog.ShowDialog();
}
}
}

```

```

folder_text.Text = folder_dialog.SelectedPath;}
private void save_btn_Click(object sender, RoutedEventArgs e)
{StreamWriter file = new StreamWriter("Config.txt");
file.Write(folder_text.Text + @"\\Документация TekroCRM");
file.WriteLine();
file.Write(fio_text.Text);
file.Close();
Directory.CreateDirectory(folder_text.Text + @"\\Документация TekroCRM");
System.Windows.MessageBox.Show("Настройки сохранены");
this.Close();}}

```

Окно fileshow

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{/// <summary>
/// Логика взаимодействия для fileshow.xaml
/// </summary>
public partial class fileshow : Window
{public fileshow()
{InitializeComponent();}
public string path;
private void Button_Click(object sender, RoutedEventArgs e)
{System.Diagnostics.Process.Start(path);}}

```

Окно list_of_agreement

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SQLite;
using System.Diagnostics;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;

```

```

using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using Tekro_CRM.Windows;
namespace Tekro_CRM
{
    /// <summary>
    /// Логика взаимодействия для list_of_agreement.xaml
    /// </summary>
    public partial class list_of_agreement : Window{
    public list_of_agreement(){
    InitializeComponent();}
    public database db = new database();
    public string parametr = "";
    private void Window_Loaded(object sender, RoutedEventArgs e)
    {
        agreements.CanUserAddRows = false;
        agreements.CanUserDeleteRows = false;
        agreements.CanUserReorderColumns = false;
        agreements.CanUserResizeColumns = false;
        agreements.CanUserResizeRows = false;
        db.dbFileName = MainWindow.dbname;
        db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
        db.dbConnection.Open();
        db.sqlCommand.Connection = db.dbConnection;
        db.sqlCommand.CommandText = "select * from Agreements, Customers where
        Agreements.customer_id = Customers.id " + parametr;
        db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
        db.DataSet.Reset();
        db.DataAdapter.Fill(db.DataTable);
        agreements.ItemsSource = db.DataTable.AsDataView();
        db.dbConnection.Close();}
    private void id_search_Click(object sender, RoutedEventArgs e)
    {
        db.DataTable.Clear();
        db.dbConnection.Open();
        db.sqlCommand.Connection = db.dbConnection;
        db.sqlCommand.CommandText = "select * from Agreements, Customers where
        Agreements.customer_id = Customers.id and num like '%" + id_text.Text+"%' ";
        db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
        db.DataSet.Reset();
        db.DataAdapter.Fill(db.DataTable);
        db.dbConnection.Close();}
    private void date_search_Click(object sender, RoutedEventArgs e)
    {
        db.DataTable.Clear();
        db.dbConnection.Open();
        db.sqlCommand.Connection = db.dbConnection;
        if(parametr == "")
        db.sqlCommand.CommandText = "select * from Agreements, Customers where
        Agreements.customer_id = Customers.id and date like '%" +
        date_text.SelectedDate.Value.Date.ToLongDateString() + "%' ";else

```

```

db.sqlCommand.CommandText = "select * from Agreements, Customers where
Agreements.customer_id = Customers.id " + parametrs +" and date like '%" +
date_text.SelectedDate.Value.Date.ToLongDateString() + "%' ";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
db.dbConnection.Close();}
private void Create_annex_Click(object sender, RoutedEventArgs e)
{new_annex na = new new_annex();
DataRowView row = (DataRowView)agreements.SelectedItems[0];
na.agreement_id = row["id"].ToString();
na.Show();}
private void list_of_annexes_Click(object sender, RoutedEventArgs e)
{list_of_annexes loa = new Windows.list_of_annexes();
DataRowView row = (DataRowView)agreements.SelectedItems[0];
loa.parametrs = "and agreement_id = " + row["id"].ToString();
loa.Show();}
private void Show_file_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)agreements.SelectedItems[0];
database dbf = new database();
dbf.dbFileName = MainWindow.dbname;
dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
path = path.Replace(@"\", @"/");
Process.Start("explorer.exe", @"/select, " + path);dbf.dbConnection.Close();}
private void Show_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)agreements.SelectedItems[0];
database dbf = new database();dbf.dbFileName = MainWindow.dbname;
dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
dbf.dbConnection.Close();
Process.Start(path);}}

```

Окно list_of_annexes

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SQLite;
using System.Diagnostics;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{
    /// <summary>
    /// Логика взаимодействия для list_of_annexes.xaml
    /// </summary>
    public partial class list_of_annexes : Window
    {
        public database db = new database();
        public string paramtrs = "";
        public list_of_annexes()
        {
            InitializeComponent();
        }
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            table.CanUserAddRows = false;
            table.CanUserDeleteRows = false;
            table.CanUserReorderColumns = false;
            table.CanUserResizeColumns = false;
            table.CanUserResizeRows = false;
            db.dbFileName = MainWindow.dbname;
            db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;
            db.sqlCommand.CommandText = "select * from Annexes, Agreements, Customers where
            Annexes.agreement_id = Agreements.id and Agreements.customer_id = Customers.id " +
            paramtrs;
            db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
            db.DataSet.Reset();
            db.DataAdapter.Fill(db.DataTable);
            table.ItemsSource = db.DataTable.AsDataView();
            db.dbConnection.Close();
        }
        private void agreement_id_search_Click(object sender, RoutedEventArgs e)
        {
            db.DataTable.Clear();
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;

```

```

db.sqlCommand.CommandText = "select * from Annexes, Agreements, Customers where
Annexes.agreement_id = Agreements.id and Agreements.customer_id = Customers.id and
agreement_id = " + agreement_id_text.Text;
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
db.dbConnection.Close(); }
private void date_search_Click(object sender, RoutedEventArgs e)
{ db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
if (paramtrs == "")
db.sqlCommand.CommandText = "select * from Annexes, Agreements, Customers where
Annexes.agreement_id = Agreements.id and Agreements.customer_id = Customers.id and
Annexes.date like '%" + date_text.SelectedDate.Value.Date.ToLongDateString() + "%' ";
else db.sqlCommand.CommandText = "select * from Annexes, Agreements, Customers where
Annexes.agreement_id = Agreements.id and Agreements.customer_id = Customers.id " + paramtrs
+ " and Annexes.date like '%" + date_text.SelectedDate.Value.Date.ToLongDateString() + "%' ";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
db.dbConnection.Close();}
private void Create_specification_Click(object sender, RoutedEventArgs e)
{ new_specification ns = new new_specification();
DataRowView row = (DataRowView)table.SelectedItems[0];
ns.annex_id = row["id"].ToString();
ns.Show();}
private void List_of_specifications_Click(object sender, RoutedEventArgs e)
{ list_of_specifications los = new list_of_specifications();
DataRowView row = (DataRowView)table.SelectedItems[0];
los.paramtrs = "and Specifications.annex_id = " + row["id"].ToString();
los.Show();}
private void Show_Click(object sender, RoutedEventArgs e)
{ DataRowView row = (DataRowView)table.SelectedItems[0];
database dbf = new database();
dbf.dbFileName = MainWindow.dbname;
dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
dbf.dbConnection.Close();
Process.Start(path);}
private void Show_file_Click(object sender, RoutedEventArgs e)
{ DataRowView row = (DataRowView)table.SelectedItems[0];
database dbf = new database();
dbf.dbFileName = MainWindow.dbname;

```

```

dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
path = path.Replace(@"\", @"\");
dbf.dbConnection.Close();
Process.Start("explorer.exe", @"/select, " + path);}}}

```

Окно list_of_business_trip

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SQLite;
using System.Diagnostics;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using Tekro_CRM.Windows;
namespace Tekro_CRM
{
    /// <summary>
    /// Логика взаимодействия для list_of_business_trip.xaml
    /// </summary>
    public partial class list_of_business_trip : Window
    {
        public list_of_business_trip()
        {
            InitializeComponent();
        }
        public database db = new database();
        public string parametr = "";
        private void DataGrid_Loaded(object sender, RoutedEventArgs e)
        {
            DataGrid.CanUserAddRows = false;
            DataGrid.CanUserDeleteRows = false;
            DataGrid.CanUserReorderColumns = false;
            DataGrid.CanUserResizeColumns = false;
            DataGrid.CanUserResizeRows = false;
            db.dbFileName = MainWindow.dbname;
            db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;

```

```

if(parameters == "")
db.sqlCommand.CommandText = "select distinct * from Business_trips, Customers where
Business_trips.Customers_id = Customers.id ";
else
db.sqlCommand.CommandText = "select distinct * from Business_trips, Customers where
Business_trips.Customers_id = Customers.id " + parameters;
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
DataGrid.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void create_report_Click(object sender, RoutedEventArgs e)
{new_business_trip_report report = new new_business_trip_report();
DataRowView row = (DataRowView)DataGrid.SelectedItems[0];
report.d.address = row["address"].ToString();
report.d.date1 = row["date1"].ToString();
report.d.date2 = row["date2"].ToString();
report.d.date3 = row["date3"].ToString();
report.d.name = row["name"].ToString();
report.d.id = row["id"].ToString();
report.Show();}
private void show_reports_Click(object sender, RoutedEventArgs e)
{list_of_business_trips_reports b_trip_r = new list_of_business_trips_reports();
DataRowView row = (DataRowView)DataGrid.SelectedItems[0];
b_trip_r.parameters = "and bt_id = " + row["id"].ToString();
b_trip_r.Show();}
private void DataGrid2_Loaded(object sender, RoutedEventArgs e){
}private void customer_search_Click(object sender, RoutedEventArgs e)
{db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "select distinct * from Business_trips, Customers where
Business_trips.Customers_id = Customers.id and name like '%" +customer_text.Text+"%";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
DataGrid.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void date_search_Click(object sender, RoutedEventArgs e)
{db.DataTable.Clear();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
if (parameters == "")
db.sqlCommand.CommandText = "select distinct * from Business_trips, Customers where
Business_trips.Customers_id = Customers.id and date3 like '%" +
date.SelectedDate.Value.Date.ToLongDateString() + "%";
else.sqlCommand.CommandText = "select distinct * from Business_trips, Customers where
Business_trips.Customers_id = Customers.id " + parameters + " and date3 like
%" +date.SelectedDate.Value.Date.ToLongDateString() + "%";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);

```

```

DataGridView.ItemsSource = db.DataTable.AsDataView();
db.dbConnection.Close();}
private void Show_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)DataGridView.SelectedItems[0];
database dbf = new database();
dbf.dbFileName = MainWindow.dbname;
dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
dbf.dbConnection.Close();
Process.Start(path);}
private void Show_file_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)DataGridView.SelectedItems[0];
database dbf = new database();
dbf.dbFileName = MainWindow.dbname;
dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
path = path.Replace(@"\", @"/");
dbf.dbConnection.Close();
Process.Start("explorer.exe", @"/select, " + path);}}

```

Okho list_of_business_trips_reports

```

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

```

```

using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{
    /// <summary>
    /// Логика взаимодействия для list_of_business_trips_reports.xaml
    /// </summary>
    public partial class list_of_business_trips_reports : Window{
    public list_of_business_trips_reports(){
    InitializeComponent();}
    public database db = new database();
    public string parametrs = "";
    private void DataGrid_Loaded(object sender, RoutedEventArgs e)
    {DataGrid.CanUserAddRows = false;
    DataGrid.CanUserDeleteRows = false;
    DataGrid.CanUserReorderColumns = false;
    DataGrid.CanUserResizeColumns = false;
    DataGrid.CanUserResizeRows = false;
    db.dbFileName = MainWindow.dbname;
    db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
    db.dbConnection.Open();
    db.sqlCommand.Connection = db.dbConnection;
    db.sqlCommand.CommandText = "select * from B_trips_reports, Customers, Business_trips where
    B_trips_reports.bt_id = Business_trips.id and Business_trips.customers_id = Customers.id " +
    parametrs;
    db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
    db.DataSet.Reset();
    db.DataAdapter.Fill(db.DataTable);
    DataGrid.ItemsSource = db.DataTable.AsDataView();
    db.dbConnection.Close();}
    private void date_Click(object sender, RoutedEventArgs e)
    {db.DataTable.Clear();
    db.dbConnection.Open();
    db.sqlCommand.Connection = db.dbConnection;
    db.sqlCommand.CommandText = "select * from B_trips_reports, Customers, Business_trips where
    B_trips_reports.bt_id = Business_trips.id and Business_trips.customers_id = Customers.id and
    B_trips_reports.date like '%" + date_text.SelectedDate.Value.ToLongDateString() + "%";
    db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
    db.DataSet.Reset();
    db.DataAdapter.Fill(db.DataTable);
    DataGrid.ItemsSource = db.DataTable.AsDataView();
    db.dbConnection.Close();}
    private void name_Click(object sender, RoutedEventArgs e)
    {db.DataTable.Clear();
    db.dbConnection.Open();
    db.sqlCommand.Connection = db.dbConnection;
    db.sqlCommand.CommandText = "select * from B_trips_reports, Customers, Business_trips where
    B_trips_reports.bt_id = Business_trips.id and Business_trips.customers_id = Customers.id and
    Customers.name like '%" + name_text.Text + "%";
    db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
    db.DataSet.Reset();
    db.DataAdapter.Fill(db.DataTable);
    DataGrid.ItemsSource = db.DataTable.AsDataView();
}
}

```

```

db.dbConnection.Close();}
private void show_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)DataGrid.SelectedItems[0];
database dbf = new database();
dbf.dbFileName = MainWindow.dbname;
dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
dbf.dbConnection.Close();
Process.Start(path);}
private void file_show_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)DataGrid.SelectedItems[0];
database dbf = new database();
dbf.dbFileName = MainWindow.dbname;
dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
path = path.Replace(@"\", @"\");
dbf.dbConnection.Close();
Process.Start("explorer.exe", @"/select, " + path);} }

```

Окно list_of_shipments

```

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

```

```

namespace Tekro_CRM.Windows
{
    /// <summary>
    /// Логика взаимодействия для list_of_shipments.xaml
    /// </summary>
    public partial class list_of_shipments : Window
    {
        public list_of_shipments()
        {
            InitializeComponent();
        }
        public database db = new database();
        public string parametr = "";
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            of_shipments.CanUserAddRows = false;
            of_shipments.CanUserDeleteRows = false;
            of_shipments.CanUserReorderColumns = false;
            of_shipments.CanUserResizeColumns = false;
            of_shipments.CanUserResizeRows = false;
            db.dbFileName = MainWindow.dbname;
            db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;
            db.sqlCommand.CommandText = "select distinct Of_shipment_transport.id as '№',
            Of_shipment_transport.date_of_shipment as 'ДатаотгрузкиизЧехии' ,
            Of_shipment_transport.name as 'Наименованиегруза'," +
            "Of_shipment_transport.weight as 'Вес, тн', Of_shipment_transport.car_number as
            'Номермашины', Of_shipment_transport.gtd as 'ГТД', Of_shipment_transport.date_of_arrival as
            'ДатаприходавОрел'," +
            "Of_shipment_transport.date_of_unloading as 'Датаразгрузки', Customers.name as 'Получатель',
            Of_shipment_transport.docs as 'Документы', Of_shipment_transport.freight_rate as
            'Фрахтоваяставка'," +
            "Of_shipment_transport.note as 'Примечание' from Of_shipment_transport, Customers where
            Of_shipment_transport.customer_id = Customers.id " + parametr;
            db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
            db.DataSet.Reset();
            db.DataAdapter.Fill(db.DataTable);
            of_shipments.ItemsSource = db.DataTable.AsDataView();
            db.dbConnection.Close();}}
}

```

Окно list_of_specifications

```

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

```

```

using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{
    /// <summary>
    /// Логика взаимодействия для list_of_specifications.xaml
    /// </summary>
    public partial class list_of_specifications : Window
    {
        public database db = new database();
        public string parametr = "";
        public list_of_specifications()
        {
            InitializeComponent();
        }
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            table.CanUserAddRows = false;
            table.CanUserDeleteRows = false;
            table.CanUserReorderColumns = false;
            table.CanUserResizeColumns = false;
            table.CanUserResizeRows = false;
            db.dbFileName = MainWindow.dbname;
            db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;
            db.sqlCommand.CommandText = "select * from Specifications, Agreements, Annexes where
            Specifications.annex_id = Annexes.id and Specifications.agreement_id = Agreements.id " +
            parametr;
            db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
            db.DataSet.Reset();
            db.DataAdapter.Fill(db.DataTable);
            table.ItemsSource = db.DataTable.AsDataView();
            db.dbConnection.Close();
        }
        private void agreement_id_search_Click(object sender, RoutedEventArgs e)
        {
            db.DataTable.Clear();
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;
            db.sqlCommand.CommandText = "select * from Specifications, Agreements, Annexes where
            Specifications.annex_id = Annexes.id and Specifications.agreement_id = Agreements.id and
            Specifications.agreement_id = " + agreement_id_text.Text;
            db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
            db.DataSet.Reset();
            db.DataAdapter.Fill(db.DataTable);
            db.dbConnection.Close();
        }
        private void date_search_Click(object sender, RoutedEventArgs e)
        {
            db.DataTable.Clear();
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;
            if (parametr == "") db.sqlCommand.CommandText = "select * from Specifications, Agreements,
            Annexes where Specifications.annex_id = Annexes.id and Specifications.agreement_id =
            Agreements.id and Specifications.date like '%" +
            date_text.SelectedDate.Value.Date.ToLongDateString() + "%' ";
            else db.sqlCommand.CommandText = "select * from Specifications, Agreements, Annexes where
            Specifications.annex_id = Annexes.id and Specifications.agreement_id = Agreements.id " +
            parametr + " and Specifications.date like '%" +
            date_text.SelectedDate.Value.Date.ToLongDateString() + "%' ";
        }
    }
}

```

```

db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
db.dbConnection.Close();}
private void Show_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)table.SelectedItems[0];
database dbf = new database();
dbf.dbFileName = MainWindow.dbname;
dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
dbf.dbConnection.Close();
Process.Start(path);}
private void Show_file_Click(object sender, RoutedEventArgs e)
{DataRowView row = (DataRowView)table.SelectedItems[0];
database dbf = new database();
dbf.dbFileName = MainWindow.dbname;
dbf.dbConnection = new SQLiteConnection("Data Source=" + dbf.dbFileName + ";Version=3;");
dbf.dbConnection.Open();
dbf.sqlCommand.Connection = dbf.dbConnection;
dbf.sqlCommand.CommandText = "select system_address from Documentation where id = " +
row["documentation_id"].ToString();
dbf.DataAdapter = new SQLiteDataAdapter(dbf.sqlCommand.CommandText, dbf.dbConnection);
dbf.DataSet.Reset();
dbf.DataAdapter.Fill(dbf.DataTable);
string path;
path = dbf.DataTable.Rows[0][0].ToString();
path = path.Replace(@"\", @"/");
dbf.dbConnection.Close();
Process.Start("explorer.exe", @"/select, " + path);}}}

```

Окно new_agreement

```

using Microsoft.Office.Interop.Word;
using System;
using System.Collections.Generic;
using System.Data.SQLite;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;

```

```

using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using Tekro_CRM.Windows;
namespace Tekro_CRM
{
    /// <summary>
    /// Логика взаимодействия для new_agreement.xaml
    /// </summary>
    public partial class new_agreement : System.Windows.Window{
    public new_agreement(){
    InitializeComponent();}
    public void Change_name(string name){
    this.group1.Header = name;}
    public buyer created_buyer = new buyer();
    private void create_buyer_Click(object sender, RoutedEventArgs e)
    {buyer_create create = new buyer_create();
    create.ShowDialog();created_buyer = create.created_buyer;
    create.editgroupname(group1, fio);}
    private void select_buyer_Click(object sender, RoutedEventArgs e)
    {buyer_select select = new buyer_select();
    select.ShowDialog();
    created_buyer = select.created_buyer;
    select.editgroupname(group1,fio);}
    _Application oWord = new Microsoft.Office.Interop.Word.Application();
    private _Document GetDoc(string path){
    _Document business_trip_doc = oWord.Documents.Add(path);
    SetTemplate(business_trip_doc);
    return business_trip_doc;}
    private void SetTemplate(_Document business_trip_doc)
    {business_trip_doc.Bookmarks["address"].Range.Text = created_buyer.address;
    business_trip_doc.Bookmarks["bank"].Range.Text = created_buyer.bank_name;
    business_trip_doc.Bookmarks["bik"].Range.Text = created_buyer.bik;
    business_trip_doc.Bookmarks["buyer"].Range.Text = created_buyer.name;
    business_trip_doc.Bookmarks["buyer2"].Range.Text = created_buyer.name;
    string date = date_picker1.SelectedDate.Value.Date.ToLongDateString();
    date.Substring(0, date.Length - 1);
    business_trip_doc.Bookmarks["date"].Range.Text =
    date_picker1.SelectedDate.Value.Date.ToLongDateString().Remove(date_picker1.SelectedDate.Va
    lue.Date.ToLongDateString().Length -1);
    business_trip_doc.Bookmarks["director"].Range.Text = fio.Text;
    string[] dir = created_buyer.director.Split(new char[] { ' ' });
    var dir1 = new StringBuilder(dir[1]);
    dir[1] = dir1[0].ToString() + ".";
    var dir2 = new StringBuilder(dir[2]);
    dir[2] = dir2[0].ToString() + ".";
    string str = dir[0] + " " + dir[1] + dir[2];
    business_trip_doc.Bookmarks["director2"].Range.Text = str;
    business_trip_doc.Bookmarks["korsch"].Range.Text = created_buyer.korsch;
    business_trip_doc.Bookmarks["kpp"].Range.Text = created_buyer.kpp;
    business_trip_doc.Bookmarks["num"].Range.Text = text_num.Text;

```

```

business_trip_doc.Bookmarks["ogrn"].Range.Text = created_buyer.ogrn;
business_trip_doc.Bookmarks["phone"].Range.Text = created_buyer.phone;
business_trip_doc.Bookmarks["rsch"].Range.Text = created_buyer.rsch;
business_trip_doc.Bookmarks["inn"].Range.Text = created_buyer.inn;
business_trip_doc.Bookmarks["okpo"].Range.Text = created_buyer.okpo;
business_trip_doc.Bookmarks["fax"].Range.Text = created_buyer.fax;
business_trip_doc.Bookmarks["date2"].Range.Text =
srok.SelectedDate.Value.Date.ToLongDateString().Remove(srok.SelectedDate.Value.Date.ToLong
DateString().Length - 1);}
private void Button_Click(object sender, RoutedEventArgs e)
{ templates temps = new templates();
string directory;
StreamReader file = new StreamReader("Config.txt");
directory = file.ReadLine() + @"\templates";
temps.agreement_template = directory + @"\ + "agreement_template.dotx";
file.Close();
string dir_name;
dir_name = created_buyer.name;
StreamReader file_dir = new StreamReader("Config.txt");
Regex regex = new Regex(@"\W");
dir_name = regex.Replace(dir_name, "");
string path;
path = file_dir.ReadLine() + @"\ + dir_name + @"\Договоры\ +
date_picker1.SelectedDate.Value.Date.ToLongDateString();
Directory.CreateDirectory(path);
_Document business_trip_doc = GetDoc(temps.agreement_template);
business_trip_doc.SaveAs(path + @"\ + text_num.Text + ".docx");
string documentation_id;
documentation_id = path + @"\ + text_num.Text + ".docx";
business_trip_doc.Close();
database db1 = new database();
db1.dbFileName = MainWindow.dbname;
db1.dbConnection = new SQLiteConnection("Data Source=" + db1.dbFileName + ";Version=3;");
db1.dbConnection.Open();
db1.sqlCommand.Connection = db1.dbConnection;
db1.sqlCommand.CommandText = "insert into Documentation ('system_address','document_type')
values (" + documentation_id + "','output')";
db1.sqlCommand.ExecuteNonQuery();
db1.sqlCommand.CommandText = "SELECT id FROM Documentation WHERE
rowid=last_insert_rowid()";
db1.DataAdapter = new SQLiteDataAdapter(db1.sqlCommand.CommandText, db1.dbConnection);
db1.DataSet.Reset();
db1.DataAdapter.Fill(db1.DataTable);
string last_id;
last_id = db1.DataTable.Rows[0][0].ToString();
db1.dbConnection.Close();
database db = new database();
db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText =

```

```

"insert into Agreements" +
" ('num','customer_id','date','documentation_id'" +
"      values      ('"+      text_num.Text      +      ",      '"+created_buyer.id+"',      '"+
date_picker1.SelectedDate.Value.Date.ToLongDateString() +      ",      '"+ last_id +      "');"
db.sqlCommand.ExecuteNonQuery();
db.dbConnection.Close();
fileshow fs = new fileshow();
fs.path = path;
fs.ShowDialog();}
private void Window_Loaded(object sender, RoutedEventArgs e)
{DateTime num = new DateTime();
num = DateTime.Now;
text_num.Text = num.ToString("dd") + num.ToString("MM");}
private void Window_Closed(object sender, EventArgs e)
{oWord.Quit();}}

```

Окно new_annex

```

using Microsoft.Office.Interop.Word;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SQLite;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{/// <summary>
/// Логика взаимодействия для new_annex.xaml
/// </summary>
public partial class new_annex : System.Windows.Window
{public string agreement_id, agreement_num, agreement_date, customer_name,
customer_director,path;
int annex_num,sum_i;
float sum;
_Application oWord = new Microsoft.Office.Interop.Word.Application();
private _Document GetDoc(string path)
{Object missingObj = System.Reflection.Missing.Value;
_Document business_trip_doc = oWord.Documents.Add(path);
SetTemplate(business_trip_doc);
Range wordRange = business_trip_doc.Bookmarks["table"].Range;
Microsoft.Office.Interop.Word.Table _table;

```

```

_table = business_trip_doc.Tables.Add(wordRange, items.Length + 1, 4, ref missingObj, ref
missingObj);
_table.Rows.Add(ref missingObj);
_table.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleSingle;
_table.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleSingle;
_table.Cell(1, 1).Range.Text = "Наименование товара";
_table.Cell(1, 2).Range.Text = "Кол-во, тонн";
_table.Cell(1, 3).Range.Text = "Цена за 1 тонну, руб., в т.ч. НДС 10%";
_table.Cell(1, 4).Range.Text = "Сумма в руб., в т.ч. НДС 10%";
string[] s;
for (int i = 2; i < items.Length + 2; i++)
{s = items[i-2].ToString().Split(new char[] { ';' });
_table.Cell(i, 1).Range.Text = s[0];
_table.Cell(i, 2).Range.Text = s[1];
_table.Cell(i, 3).Range.Text = s[2];
_table.Cell(i, 4).Range.Text = s[3];}
_table.Cell(items.Length + 2, 1).Range.Text = "Всего:";
_table.Cell(items.Length + 2, 2).Range.Text = sum_i.ToString();
_table.Cell(items.Length + 2, 3).Range.Text = "";
_table.Cell(items.Length + 2, 4).Range.Text = sum.ToString() + " руб" ;
for (int i = 1; i<=3+1; i++)
for (int j = 1; j<=4; j++)
_table.Cell(i, j).VerticalAlignment = WdCellVerticalAlignment.wdCellAlignVerticalCenter;
return business_trip_doc;}
object[] items;
private void object_create_Click(object sender, RoutedEventArgs e)
{products_table products = new products_table();
products.ShowDialog();
items = products.items;
object_create.Content = "Таблица товаров создана";
object_create.IsEnabled = false;
summa.Text = products.sum.ToString();
summa.IsEnabled = false;
sum = products.sum;
string[] s;
for (int i = 0; i < items.Length; i++)
{s = items[i].ToString().Split(new char[] { ';' });
sum_i = sum_i + Convert.ToInt32(s[1]);}
private void Window_Closed(object sender, EventArgs e)
{oWord.Quit();}
private void SetTemplate(_Document business_trip_doc)
{business_trip_doc.Bookmarks["date"].Range.Text
date1.SelectedDate.Value.ToLongDateString() ;
business_trip_doc.Bookmarks["dated"].Range.Text = agreement_date;
business_trip_doc.Bookmarks["director"].Range.Text = fio.Text;
string[] dir = customer_director.Split(new char[] { ' ' });
var dir1 = new StringBuilder(dir[1]);
dir[1] = dir1[0].ToString() + ".";
var dir2 = new StringBuilder(dir[2]);
dir[2] = dir2[0].ToString() + ".";
business_trip_doc.Bookmarks["director2"].Range.Text = str;
business_trip_doc.Bookmarks["dnum"].Range.Text = agreement_num;
}

```

```

business_trip_doc.Bookmarks["name"].Range.Text = customer_name;
business_trip_doc.Bookmarks["name2"].Range.Text = customer_name;
business_trip_doc.Bookmarks["num"].Range.Text = annex_num.ToString();
business_trip_doc.Bookmarks["proisvoditel"].Range.Text = proizv1.Text;
business_trip_doc.Bookmarks["sostav"].Range.Text = sostav1.Text;
business_trip_doc.Bookmarks["srok"].Range.Text = srok1.Text;
business_trip_doc.Bookmarks["sum"].Range.Text = summa.Text;
business_trip_doc.Bookmarks["sum2"].Range.Text = summa_t.Text;
business_trip_doc.Bookmarks["uslovia"].Range.Text = uslovia.Text;
business_trip_doc.Bookmarks["usloviap"].Range.Text = uslovia_p1.Text;}
private void create_Click(object sender, RoutedEventArgs e)
{
    templates temps = new templates();
    string directory;
    StreamReader file = new StreamReader("Config.txt");
    directory = file.ReadLine() + @"\templates";
    temps.agreement_template = directory + @"\" + "annex_template.dotx";
    file.Close();
    string dir_name;
    dir_name = customer_name;
    StreamReader file_dir = new StreamReader("Config.txt");
    Regex regex = new Regex(@"\W");
    dir_name = regex.Replace(dir_name, "");
    string path;
    path = file_dir.ReadLine() + @"\" + dir_name + @"\Договоры\" + agreement_date +
    @"\Приложения";
    Directory.CreateDirectory(path);
    _Document business_trip_doc = GetDoc(temps.agreement_template);
    business_trip_doc.SaveAs(path + @"\Приложение №" + annex_num.ToString() + ".docx");
    string documentation_id;
    documentation_id = path + @"\Приложение №" + annex_num.ToString() + ".docx";
    business_trip_doc.Close();
    database db1 = new database();
    db1.dbFileName = MainWindow.dbname;
    db1.dbConnection = new SQLiteConnection("Data Source=" + db1.dbFileName + ";Version=3;");
    db1.dbConnection.Open();
    db1.sqlCommand.Connection = db1.dbConnection;
    db1.sqlCommand.CommandText = "insert into Documentation ('system_address','document_type')
    values ('" + documentation_id + "','output')";
    db1.sqlCommand.ExecuteNonQuery();
    db1.sqlCommand.CommandText = "SELECT id FROM Documentation WHERE
    rowid=last_insert_rowid()";
    db1.DataAdapter = new SQLiteDataAdapter(db1.sqlCommand.CommandText, db1.dbConnection);
    db1.DataSet.Reset();
    db1.DataAdapter.Fill(db1.DataTable);
    string last_id;
    last_id = db1.DataTable.Rows[0][0].ToString();
    db1.dbConnection.Close();
    database db = new database();
    db.dbFileName = MainWindow.dbname;
    db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
    db.dbConnection.Open();
    db.sqlCommand.Connection = db.dbConnection;

```

```

db.sqlCommand.CommandText = "insert into Annexes (agreement_id, date, documentation_id,
num) values ('+ agreement_id +'", "'+date1.SelectedDate.Value.ToLongDateString()+'",
"' +last_id+'", "'+ annex_num.ToString() + "')";
db.sqlCommand.ExecuteNonQuery();
db.dbConnection.Close();
fileshow fs = new fileshow();
fs.path = path;
fs.ShowDialog();}
private void Window_Loaded(object sender, RoutedEventArgs e)
{db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "select Agreements.id, Agreements.date, Agreements.num,
Customers.name, Customers.director, Documentation.system_address from Agreements,
Customers, Documentation where Agreements.customer_id = Customers.id and
Agreements.documentation_id = Documentation.id and Agreements.id = "+ agreement_id + "'";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
agreement_num = db.DataTable.Rows[0].ItemArray[2].ToString();
agreement_date = db.DataTable.Rows[0].ItemArray[1].ToString();
customer_name = db.DataTable.Rows[0].ItemArray[3].ToString();
customer_director = db.DataTable.Rows[0].ItemArray[4].ToString();
path = db.DataTable.Rows[0].ItemArray[5].ToString();
db.dbConnection.Close();
path = path.Substring(0, path.LastIndexOf("\\"));
database db1 = new database();
db1.dbFileName = MainWindow.dbname;
db1.dbConnection = new SQLiteConnection("Data Source=" + db1.dbFileName + ";Version=3;");
db1.dbConnection.Open();
db1.sqlCommand.Connection = db1.dbConnection;
db1.sqlCommand.CommandText = "select count(id) from Annexes where agreement_id = " +
agreement_id;
db1.DataAdapter = new SQLiteDataAdapter(db1.sqlCommand.CommandText, db1.dbConnection);
db1.DataSet.Reset();
db1.DataAdapter.Fill(db1.DataTable);
annex_num = Int32.Parse(db1.DataTable.Rows[0].ItemArray[0].ToString());
annex_num = annex_num + 1;
db1.dbConnection.Close();
fio.Text = customer_director;}
buyer annex_buyer = new buyer();
database db = new database();
public new_annex(){
InitializeComponent();}}

```

Окно new_business_trip

```

using Microsoft.Office.Interop.Word;
using System;

```

```

using System.Collections.Generic;
using System.Data.SQLite;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using Tekro_CRM.Windows;
namespace Tekro_CRM
{
    /// <summary>
    /// Логика взаимодействия для new_business_trip.xaml
    /// </summary>
    public partial class new_business_trip : System.Windows.Window
    {
        public new_business_trip()
        {
            InitializeComponent();
        }
        void ClearTextBoxes(DependencyObject obj)
        {
            if (obj is TextBox)
            {
                ((TextBox)obj).Text = null;
            }
            ClearTextBoxes(VisualTreeHelper.GetChild(obj, 1));
        }
        private void ClearButton_Click(object sender, RoutedEventArgs e)
        {
            ClearTextBoxes(this);
        }
        _Application oWord = new Microsoft.Office.Interop.Word.Application();
        private _Document GetDoc(string path)
        {
            _Document business_trip_doc = oWord.Documents.Add(path);
            SetTemplate(business_trip_doc);
            return business_trip_doc;
        }
        private void SetTemplate(_Document business_trip_doc)
        {
            business_trip_doc.Bookmarks["data"].Range.Text = text_data.Text.ToString() +
            Environment.NewLine;
            business_trip_doc.Bookmarks["date1"].Range.Text =
            date_picker1.SelectedDate.Value.Date.ToLongDateString();
            business_trip_doc.Bookmarks["date2"].Range.Text =
            date_picker2.SelectedDate.Value.Date.ToLongDateString();
            business_trip_doc.Bookmarks["date3"].Range.Text =
            date_picker3.SelectedDate.Value.Date.ToShortDateString();
            business_trip_doc.Bookmarks["fio"].Range.Text = text_fio.Text.ToString();
            business_trip_doc.Bookmarks["place"].Range.Text = created_buyer.address + " " +
            created_buyer.name ;
            business_trip_doc.Bookmarks["time"].Range.Text = text_time.Text.ToString();
        }
        private void CreateButton_Click(object sender, RoutedEventArgs e)
        {
            templates temps = new templates();
            string directory;
            StreamReader file = new StreamReader("Config.txt");
            directory = file.ReadLine() + @"\templates";
        }
    }
}

```

```

temps.business_trip_template = directory + @"\" + "business_trip_template.dotx";
file.Close();
string dir_name;
dir_name = created_buyer.name;
StreamReader file_dir = new StreamReader("Config.txt");
Regex regex = new Regex(@"\W");
dir_name = regex.Replace(dir_name, "");
string path;
path = file_dir.ReadLine() + @"\" + dir_name + @"\Командировки\" +
date_picker3.SelectedDate.Value.Date.ToLongDateString();
Directory.CreateDirectory(path);
_Document business_trip_doc = GetDoc(temps.business_trip_template);
business_trip_doc.SaveAs(path + @"\Заявка.docx");
string documentation_id;
documentation_id = path + @"\Заявка.docx";
business_trip_doc.Close();
database db1 = new database();
db1.dbFileName = MainWindow.dbname;
db1.dbConnection = new SQLiteConnection("Data Source=" + db1.dbFileName + ";Version=3;");
db1.dbConnection.Open();
db1.sqlCommand.Connection = db1.dbConnection;
db1.sqlCommand.CommandText = "insert into Documentation ('system_address','document_type')
values ('" + documentation_id + "','output')";
db1.sqlCommand.ExecuteNonQuery();
db1.sqlCommand.CommandText = "SELECT id FROM Documentation WHERE
rowid=last_insert_rowid()";
db1.DataAdapter = new SQLiteDataAdapter(db1.sqlCommand.CommandText, db1.dbConnection);
db1.DataSet.Reset();
db1.DataAdapter.Fill(db1.DataTable);
string last_id;
last_id = db1.DataTable.Rows[0][0].ToString();
db1.dbConnection.Close();
database db = new database();
db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "insert into Business_trips (date1, date2, date3, customers_id,
closed, 'documentation_id') " +
"values ('" + date_picker1.SelectedDate.Value.Date.ToLongDateString() + "', '" +
date_picker2.SelectedDate.Value.Date.ToLongDateString() + "', '" +
date_picker3.SelectedDate.Value.Date.ToLongDateString() + "', '" + created_buyer.id + "', '0', '" +
last_id + "')";
db.sqlCommand.ExecuteNonQuery();
db.dbConnection.Close();
fileshow fs = new fileshow();
fs.path = path;
fs.ShowDialog();}
public buyer created_buyer = new buyer();
private void place_btn_Click(object sender, RoutedEventArgs e)
{buyer_select select = new buyer_select();
select.ShowDialog();
}

```

```

created_buyer = select.created_buyer;}
private void Window_Closed(object sender, EventArgs e)
{oWord.Quit();}}

```

Окно new_business_trip_report

```

using Microsoft.Office.Interop.Word;
using System;
using System.Collections.Generic;
using System.Data.SQLite;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{/// <summary>
/// Логика взаимодействия для new_business_trip_report.xaml
/// </summary>
public partial class new_business_trip_report : System.Windows.Window
{public new_business_trip_report()
{InitializeComponent();}
public class data
{public string date1, date2, date3, address, name,id;}
void ClearTextboxes(DependencyObject obj)
{for (int i = 0; i < VisualTreeHelper.GetChildrenCount(obj); i++)
{if (obj is TextBox)
{((TextBox)obj).Text = null;}
ClearTextboxes(VisualTreeHelper.GetChild(obj, i));}}
public data d = new data();
_Application oWord = new Microsoft.Office.Interop.Word.Application();
private _Document GetDoc(string path)
{ _Document business_trip_doc = oWord.Documents.Add(path);
SetTemplate(business_trip_doc);
return business_trip_doc;}
private void SetTemplate(_Document business_trip_doc)
{business_trip_doc.Bookmarks["address"].Range.Text = d.address;
business_trip_doc.Bookmarks["date1"].Range.Text = d.date1;
business_trip_doc.Bookmarks["date2"].Range.Text = d.date2;
business_trip_doc.Bookmarks["date3"].Range.Text = d.date1;
StreamReader file = new StreamReader("Config.txt");
string str;

```

```

str = file.ReadLine();
str = file.ReadLine();
business_trip_doc.Bookmarks["fio"].Range.Text = str;
business_trip_doc.Bookmarks["name"].Range.Text = d.name;
business_trip_doc.Bookmarks["name2"].Range.Text = d.name;
business_trip_doc.Bookmarks["result"].Range.Text = result.Text;
business_trip_doc.Bookmarks["result2"].Range.Text = result2.Text;
business_trip_doc.Bookmarks["role"].Range.Text = role.Text;
business_trip_doc.Bookmarks["target"].Range.Text = target.Text;}
private void Create_Click(object sender, RoutedEventArgs e)
{ templates temps = new templates();
string directory;
StreamReader file = new StreamReader("Config.txt");
directory = file.ReadLine() + @"\templates";
temps.business_trip_template = directory + @"\ + "bt_report_template.dotx";
file.Close();
string dir_name;
dir_name = d.name;
StreamReader file_dir = new StreamReader("Config.txt");
Regex regex = new Regex(@"\W");
dir_name = regex.Replace(dir_name, "");
string path;
path = file_dir.ReadLine() + @"\ + dir_name + @"\Командировки\" + d.date3;
Directory.CreateDirectory(path);
_Document business_trip_doc = GetDoc(temps.business_trip_template);
business_trip_doc.SaveAs(path + @"\Отчет.docx");
string documentation_id;
documentation_id = path + @"\Отчет.docx";
business_trip_doc.Close();
database db1 = new database();
db1.dbFileName = MainWindow.dbname;
db1.dbConnection = new SQLiteConnection("Data Source=" + db1.dbFileName + ";Version=3;");
db1.dbConnection.Open();
db1.sqlCommand.Connection = db1.dbConnection;
db1.sqlCommand.CommandText = "insert into Documentation ('system_address','document_type')
values ('" + documentation_id + "','output')";
db1.sqlCommand.ExecuteNonQuery();
db1.sqlCommand.CommandText = "SELECT id FROM Documentation WHERE
rowid=last_insert_rowid()";
db1.DataAdapter = new SQLiteDataAdapter(db1.sqlCommand.CommandText, db1.dbConnection);
db1.DataSet.Reset();
db1.DataAdapter.Fill(db1.DataTable);
string last_id;
last_id = db1.DataTable.Rows[0][0].ToString();
db1.dbConnection.Close();
database db = new database();
db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "insert into B_trips_reports (bt_id, date, 'documentation_id') " +
"values ('" + d.id + "',' + DateTime.Today.ToLongDateString() + "',' + last_id + "')";

```

```

db.sqlCommand.ExecuteNonQuery();
db.dbConnection.Close();
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "update Business_trips set closed = 1 where id = " + d.id;
db.sqlCommand.ExecuteNonQuery();
db.dbConnection.Close();
fileshow fs = new fileshow();
fs.path = path;
fs.ShowDialog();}
private void Clear_Click(object sender, RoutedEventArgs e)
{ClearTextBoxes(this);}}

```

Окно new_of_shipment

```

using System;
using System.Collections.Generic;
using System.Data.SQLite;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{/// <summary>
/// Логика взаимодействия для new_of_shipment.xaml
/// </summary>
public partial class new_of_shipment : Window
{public new_of_shipment()
{InitializeComponent();}
void ClearTextBoxes(DependencyObject obj)
{for (int i = 0; i < VisualTreeHelper.GetChildrenCount(obj); i++)
{if (obj is TextBox)
{((TextBox)obj).Text = null;}
ClearTextBoxes(VisualTreeHelper.GetChild(obj, i));}}
string id = "",name = "";
private void button_clear_Click(object sender, RoutedEventArgs e)
{ClearTextBoxes(this);}
private void button_create_Click(object sender, RoutedEventArgs e)
{database db = new database();
db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;

```

```

db.sqlCommand.CommandText = "insert into Of_shipment_transport
(customer_id,'date_of_shipment','name','weight','car_number','gtd','date_of_arrival','date_of_unloa
ding','freight_rate','note','docs') " +
"values ("'+id+ "'','"+date1.SelectedDate.Value.ToLongDateString()+ "'','"+text_name.Text+
"'','"+text_weight.Text+ "'','"+text_carnum.Text+ "'','"+text_gtd.Text+
"'','"+date2.SelectedDate.Value.ToLongDateString()+
"'','"+date3.SelectedDate.Value.ToLongDateString()+ "'','"+text_rfrate.Text+
"'','"+text_comment.Text+"', '"+text_docs.Text+")";
db.sqlCommand.ExecuteNonQuery();
db.dbConnection.Close();
MessageBox.Show("Данные об отгрузке сохранены");}
private void select_customer_Click(object sender, RoutedEventArgs e)
{buyer_select bs = new buyer_select();
bs.isos = true;
bs.ShowDialog();
id = bs.id_os;
name = bs.name_os;
text_customer.Text = name;
button_create.IsEnabled = true;}}

```

Окно new_specification

```

using Microsoft.Office.Interop.Word;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Data.SQLite;
using System.IO;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Forms;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{/// <summary>
/// Логика взаимодействия для new_specification.xaml
/// </summary>
public partial class new_specification : System.Windows.Window
{public new_specification()
{InitializeComponent();
}
Microsoft.Office.Interop.Excel.Application ObjWorkExcel = new
Microsoft.Office.Interop.Excel.Application();
Microsoft.Office.Interop.Excel.Workbook ObjWorkBook;

```

```

Array values;
int iLastRow, count, countnotnull = 0;
private void file_select_Click(object sender, RoutedEventArgs e)
{ OpenFileDialog file = new OpenFileDialog();
file.ShowDialog();
file_name.Text = file.FileName;
ObjWorkBook = ObjWorkExcel.Workbooks.Open(file.FileName, Type.Missing, Type.Missing,
Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,
Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing);
Microsoft.Office.Interop.Excel.Worksheet ObjWorkSheet =
(Microsoft.Office.Interop.Excel.Worksheet)ObjWorkBook.Sheets[1];
iLastRow =
ObjWorkSheet.Cells.SpecialCells(Microsoft.Office.Interop.Excel.XlCellType.xlCellTypeLastCell).
Row - 13;
Microsoft.Office.Interop.Excel.Range range = ObjWorkSheet.get_Range("O27", "W" +
iLastRow.ToString());
values = (Array)range.Cells.Value;
count = values.Length / 9;
for (int i = 1; i <= count; i++)
if (values.GetValue(i, 1) != null)
_Application oWord = new Microsoft.Office.Interop.Word.Application();
private _Document GetDoc(string path)
{ Object missingObj = System.Reflection.Missing.Value;
_Document business_trip_doc = oWord.Documents.Add(path);
SetTemplate(business_trip_doc);
Range wordRange = business_trip_doc.Bookmarks["table"].Range;
Microsoft.Office.Interop.Word.Table _table;
_table = business_trip_doc.Tables.Add(wordRange, countnotnull, 3, ref missingObj, ref
missingObj);
_table.Rows.Add(ref missingObj);
_table.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleSingle;
_table.Borders.OutsideLineStyle = WdLineStyle.wdLineStyleSingle;
_table.AutoFitBehavior(WdAutoFitBehavior.wdAutoFitContent);
_table.Cell(1, 1).Range.Text = "Показатель";
_table.Cell(1, 1).Range.Bold = 2;
_table.Cell(1, 2).Range.Text = "Ед. изм.";
_table.Cell(1, 2).Range.Bold = 2;
_table.Cell(1, 3).Range.Text = recipe_name.Text ;
_table.Cell(1, 3).Range.Bold = 2;
int i1 = 1, i2 = 1; while (i1 <= count){
if (values.GetValue(i1, 1) != null && values.GetValue(i1, 7) != null)
{ _table.Cell(i2+1, 1).Range.Text = values.GetValue(i1, 1).ToString();
_table.Cell(i2 + 1, 1).Range.ParagraphFormat.Alignment =
WdParagraphAlignment.wdAlignParagraphLeft;
_table.Cell(i2+1, 2).Range.Text = values.GetValue(i1, 7).ToString();
_table.Cell(i2+1, 3).Range.Text = values.GetValue(i1, 9).ToString(); i1++; i2++;
continue; }
if (values.GetValue(i1, 1) != null && values.GetValue(i1, 7) == null)
{ object begCell = _table.Cell(i2+1, 1).Range.Start;
object endCell = _table.Cell(i2+1, 2).Range.End;
Range wordcellrange = business_trip_doc.Range(ref begCell, ref endCell);
wordcellrange.Select(); oWord.Selection.Cells.Merge();
}
}
}

```

```

_table.Cell(i2 + 1, 1).Range.Text = values.GetValue(i1, 1).ToString();
_table.Cell(i2 + 1, 2).Range.Text = values.GetValue(i1, 9).ToString();
_table.Cell(i2 + 1, 1).Range.ParagraphFormat.Alignment =
WdParagraphAlignment.wdAlignParagraphLeft;
i1++;i2++;continue;}
if (values.GetValue(i1, 1) == null){i1++;
continue;}}return business_trip_doc;}
private void SetTemplate(_Document business_trip_doc)
{string[] dir = director.Split(new char[] { ' ' });
var dir1 = new StringBuilder(dir[1]);
dir[1] = dir1[0].ToString() + ".";
var dir2 = new StringBuilder(dir[2]);
dir[2] = dir2[0].ToString() + ".";
string str = dir[1] + dir[2] + " " + dir[0];
business_trip_doc.Bookmarks["agreementdate"].Range.Text = giveinfo(sqldate, 2);
business_trip_doc.Bookmarks["agreementdate2"].Range.Text = giveinfo(sqldate, 2);
business_trip_doc.Bookmarks["agreementnum"].Range.Text = giveinfo(sqldate, 1);
business_trip_doc.Bookmarks["agreementnum2"].Range.Text = giveinfo(sqldate, 1);
business_trip_doc.Bookmarks["annexdate"].Range.Text = giveinfo(sqldate, 4);
business_trip_doc.Bookmarks["annexdate2"].Range.Text = giveinfo(sqldate, 4);
business_trip_doc.Bookmarks["annexnum"].Range.Text = giveinfo(sqldate, 5);
business_trip_doc.Bookmarks["annexnum2"].Range.Text = giveinfo(sqldate, 5);
business_trip_doc.Bookmarks["buyer"].Range.Text = name;
business_trip_doc.Bookmarks["fio"].Range.Text = str;
business_trip_doc.Bookmarks["specnum"].Range.Text = spec_num.ToString();
business_trip_doc.Bookmarks["specnum2"].Range.Text = spec_num.ToString();}
string director, name;public string annex_id;
public int spec_num, annex_num;
private void Window_Closed(object sender, EventArgs e)
{ObjWorkExcel.Quit();oWord.Quit();}
string sqldate;string giveinfo(string sql, int id)
{database db = new database();
string str = db.returnstr(sql, id);db = null;return str;
GC.Collect();GC.WaitForPendingFinalizers();}
private void create_Click(object sender, RoutedEventArgs e)
{director = giveinfo("select * from Customers, Agreements, Annexes where Annexes.agreement_id
= Agreements.id and Agreements.customer_id = Customers.id and Annexes.id = " + annex_id, 2);
name = giveinfo("select * from Customers, Agreements, Annexes where Annexes.agreement_id =
Agreements.id and Agreements.customer_id = Customers.id and Annexes.id = " + annex_id, 14);
templates temps = new templates();
string directory;
StreamReader file = new StreamReader("Config.txt");
directory = file.ReadLine() + @"\templates";
temps.agreement_template = directory + @"\" + "specification_template.dotx";
file.Close();string dir_name;dir_name = name;
StreamReader file_dir = new StreamReader("Config.txt");
Regex regex = new Regex(@"\W");dir_name = regex.Replace(dir_name, "");
string path;
path = file_dir.ReadLine() + @"\" + dir_name + @"\Договоры\" + giveinfo(sqldate, 2) +
@"\Приложения\Спецификации";
Directory.CreateDirectory(path);
_Document business_trip_doc = GetDoc(temps.agreement_template);

```

```

business_trip_doc.SaveAs(path + @"\\Спецификация №" + spec_num.ToString() + "
кПриложению №" + annex_num + ".docx");
string documentation_id;
documentation_id = path + @"\\Спецификация №" + spec_num.ToString() + " кПриложению №"
+ annex_num + ".docx";
business_trip_doc.Close();database db1 = new database();
db1.dbFileName = MainWindow.dbname;
db1.dbConnection = new SQLiteConnection("Data Source=" + db1.dbFileName + ";Version=3;");
db1.dbConnection.Open();
db1.sqlCommand.Connection = db1.dbConnection;
db1.sqlCommand.CommandText = "insert into Documentation ('system_address','document_type')
values ('" + documentation_id + "','output')";
db1.sqlCommand.ExecuteNonQuery();
db1.sqlCommand.CommandText = "SELECT id FROM Documentation WHERE
rowid=last_insert_rowid()";
db1.DataAdapter = new SQLiteDataAdapter(db1.sqlCommand.CommandText, db1.dbConnection);
db1.DataSet.Reset();
db1.DataAdapter.Fill(db1.DataTable);
string last_id;
last_id = db1.DataTable.Rows[0][0].ToString();
db1.dbConnection.Close();
db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.sqlCommand.Connection = db.dbConnection;
db.sqlCommand.CommandText = "insert into Specifications (annex_id, agreement_id, date,
documentation_id) values ('" + giveinfo(sqldate, 3) + "', '" + giveinfo(sqldate, 0) + "', '" +
DateTime.Today.ToLongDateString() + "', '"+last_id+"')";
db.dbConnection.Close();
fileshow fs = new fileshow();
fs.path = path;fs.ShowDialog();}
private void Window_Loaded(object sender, RoutedEventArgs e)
{sqldate = "select Agreements.id as aid, Agreements.num as anum, Agreements.date as adate,
Annexes.id as anid, Annexes.date as andate, Annexes.num as annum from Agreements, Annexes
where Annexes.agreement_id = Agreements.id and Annexes.id = " + annex_id;
database db1 = new database();
db1.dbFileName = MainWindow.dbname;
db1.dbConnection = new SQLiteConnection("Data Source=" + db1.dbFileName + ";Version=3;");
db1.dbConnection.Open();
db1.sqlCommand.Connection = db1.dbConnection;
db1.sqlCommand.CommandText = "select count(id) from Specifications where annex_id = " +
annex_id;
db1.DataAdapter = new SQLiteDataAdapter(db1.sqlCommand.CommandText, db1.dbConnection);
db1.DataSet.Reset();
db1.DataAdapter.Fill(db1.DataTable);
spec_num = Int32.Parse(db1.DataTable.Rows[0].ItemArray[0].ToString());
spec_num = spec_num + 1;db1.dbConnection.Close();
annex_num = Int32.Parse(giveinfo(sqldate,5));}}}

```

Окно of_shipments_to_csv

```

using Microsoft.Office.Interop.Excel;
using System;

```

```

using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Data.SQLite;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Forms;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{
    /// <summary>
    /// Логика взаимодействия для of_shipments_to_csv.xaml
    /// </summary>
    public partial class of_shipments_to_csv : System.Windows.Window
    {
        public of_shipments_to_csv()
        {
            InitializeComponent();
        }

        database db = new database();
        private void to_csv_Click(object sender, RoutedEventArgs e)
        {
            db.dbFileName = MainWindow.dbname;
            db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
            db.dbConnection.Open();
            db.sqlCommand.Connection = db.dbConnection;
            db.sqlCommand.CommandText = "select distinct Of_shipment_transport.id as '№',
            Of_shipment_transport.date_of_shipment as 'ДатаотгрузкиизЧехии' ,
            Of_shipment_transport.name as 'Наименованиегруза'," +
            "Of_shipment_transport.weight as 'Вес, тн', Of_shipment_transport.car_number as
            'Номермашины', Of_shipment_transport.gtd as 'ГТД', Of_shipment_transport.date_of_arrival as
            'ДатаприходавОрел'," +
            "Of_shipment_transport.date_of_unloading as 'Датаразгрузки', Customers.name as 'Получатель',
            Of_shipment_transport.docs as 'Документы', Of_shipment_transport.freight_rate as
            'Фрахтоваяставка'," +
            "Of_shipment_transport.note as 'Примечание' from Of_shipment_transport, Customers where
            Of_shipment_transport.customer_id = Customers.id";
            db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
            db.DataSet.Reset();
            db.DataAdapter.Fill(db.DataTable);
            StringBuilder sb = new StringBuilder();
            IEnumerable<string> columnNames = db.DataTable.Columns.Cast<DataColumn>().
            Select(column => column.ColumnName);
            sb.AppendLine(string.Join(";", columnNames));
            foreach (DataRow row in db.DataTable.Rows)
            {
                IEnumerable<string> fields = row.ItemArray.Select(field => field.ToString());
                sb.AppendLine(string.Join(";", fields));
            }
        }
    }
}

```

```

using (StreamWriter writer = new StreamWriter(text_path.Text + @"\Отгрузки.csv", true,
Encoding.GetEncoding("Windows-1251"))){
writer.Write(sb.ToString());
writer.Close();}
System.Windows.MessageBox.Show("Данные были экспортированы");}
private void select_path_Click(object sender, RoutedEventArgs e)
{FolderBrowserDialog folder_dialog = new FolderBrowserDialog();
folder_dialog.ShowDialog();
text_path.Text = folder_dialog.SelectedPath;}}

```

Окно products_table

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows
{/// <summary>
/// Логика взаимодействия для products_table.xaml
/// </summary>
public partial class products_table : Window
{public products_table()
{InitializeComponent();}
DataTable dt = new DataTable();
DataColumn column;
DataRow row;
public void create_column(DataTable t, Type type, string name )
{column = new DataColumn();
column.DataType = type;
column.ColumnName = name;
t.Columns.Add(column);}
public class Add_str
{private string name, count, price, cost;
public string Name { get { return name; } set { name = value; } }
public string Count { get { return count; } set { count = value; } }
public string Price { get { return price; } set { price = value; } }
public void setcost()
{cost = (Convert.ToSingle(count) * Convert.ToSingle(price)).ToString();}
private void Button_Click(object sender, RoutedEventArgs e)
{row = dt.NewRow();

```

```

row["Name"] = name_text.Text;row["Count"] = count_text.Text;
string str = price_text.Text;
str = str.Replace(".", ",");
float cost = Convert.ToSingle(count_text.Text) * Convert.ToSingle(str);
row["Price"] = str;row["Cost"] = cost.ToString();
dt.Rows.Add(row);}
private void Window_Loaded(object sender, RoutedEventArgs e)
{ table.CanUserAddRows = false;
table.ItemsSource = dt.AsDataView();
create_column(dt, Type.GetType("System.String"), "Name");
create_column(dt, Type.GetType("System.String"), "Count");
create_column(dt, Type.GetType("System.String"), "Price");
create_column(dt, Type.GetType("System.String"), "Cost"); }
private void update_Click(object sender, RoutedEventArgs e)
{ DataRowView row = (DataRowView)table.SelectedItems[0];
int rowid = dt.Rows.IndexOf(row.Row);
string str = dt.Rows[rowid]["Price"].ToString();
str = str.Replace(".", ",");
float cost = Convert.ToSingle(dt.Rows[rowid]["Count"].ToString()) * Convert.ToSingle(str);
dt.Rows[rowid]["Price"] = str;
dt.Rows[rowid]["Cost"] = cost.ToString();}
public object[] items;
public float sum = 0;private void save_Click(object sender, RoutedEventArgs e){
ArrayList rows = new ArrayList();
foreach (DataRow dataRow in dt.Rows)
rows.Add(string.Join(";", dataRow.ItemArray.Select(item => item.ToString())));
items = rows.ToArray();string[] s;
for (int i = 0; i < items.Length; i++)
{s = items[i].ToString().Split(new char[] { ';' });
sum = sum + Convert.ToSingle(s[3]);}
this.Close();}}

```

Окно Window1

```

using Microsoft.Reporting.WinForms;
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SQLite;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Forms;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace Tekro_CRM.Windows

```

```

{/// <summary>
/// Логика взаимодействия для Window1.xaml
/// </summary>
public partial class Window1 : Window
{public Window1()
{InitializeComponent();}
private void Window_Loaded(object sender, RoutedEventArgs e){}
public string report = "";
private bool _isReportViewerLoaded;
public database db = new database();
private void _reportViewer_Load(object sender, EventArgs e)
{if (!_isReportViewerLoaded){
db.dbFileName = MainWindow.dbname;
db.dbConnection = new SQLiteConnection("Data Source=" + db.dbFileName + ";Version=3;");
db.dbConnection.Open();
db.sqlCommand.Connection = db.dbConnection;
if(report == "customers")
db.sqlCommand.CommandText = "select * from Customers where customer = 1";
if (report == "customers1")
db.sqlCommand.CommandText = "select * from Customers where customer = 0 or customer is
NULL";
db.DataAdapter = new SQLiteDataAdapter(db.sqlCommand.CommandText, db.dbConnection);
db.DataSet.Reset();
db.DataAdapter.Fill(db.DataTable);
db.dbConnection.Close();
if(report == "customers")
_reportViewer.LocalReport.ReportPath = "Customers.rdlc";
if (report == "customers1")
_reportViewer.LocalReport.ReportPath = "Customers1.rdlc";
LocalReport localreport = _reportViewer.LocalReport;
localreport.DataSources.Add(new ReportDataSource("DataSet1", db.DataTable));
_reportViewer.RefreshReport();
_isReportViewerLoaded = true;}}}}

```

ПРИЛОЖЕНИЕ 2

Таблица 1 – Атрибуты сущности «Отгрузки»

Название	Имя атрибута	Тип атрибута
ID_отгрузки	Первичный ключ таблицы «Отгрузки»	Integer
ID_клиенты	Первичный ключ таблицы «Клиенты»	Integer
Дата отгрузки	Дата отгрузки товара из Чехии	Text
Дата прибытия	Дата прибытия товара в России	Text
Наименование	Название товара	Text
Рецепт	Кормовой рецепт	Text
Вес нетто	Вес товара	Text
Цена	Цена товара	Text
Транспорт итог	Сумма затрат на транспорт	Text
Транспорт	Цена перевозки за тонну	Text
Доход без транспорта	Чистый доход за тонну	Text
КК Итог	Цена концентрата, всего	Text
Дата	Дата разгрузки товара в России	Text
Отсрочка	Временной интервал между заключением договора и загрузкой товара	Text

Таблица 2 – Атрибуты сущности «Отгрузки транспорт»

Название	Имя атрибута	Тип атрибута
ID_отгрузки тс	Первичный ключ таблицы «Отгрузки транспорт»	Integer
ID_клиенты	Первичный ключ таблицы «Клиенты»	Integer
Дата отгрузки	Дата отгрузки товара из Чехии	Text
Наименование	Название товара	Text
Вес	Вес товара	Text
Номер машины	Регистрационный номер машины	Text
ГТД	Номер грузовой таможенной декларации	Text
Дата прибытия	Дата прибытия товара в России	Text
Дата разгрузки	Дата разгрузки товара в России	Text
Фрахтовая ставка	Цена морской перевозки товара	Text
Примечание	Транспортная компания и тип доставки	Text

Таблица 3 – Атрибуты сущности «Клиенты»

Название	Имя атрибута	Тип атрибута
ID_клиенты	Первичный ключ таблицы «Клиенты»	Integer
Адрес	Адрес компании	Text
Директор	ФИО генерального директора	Text
Контактное лицо	ФИО контакта для связи	Text
Почта контактного лица	Электронная почта контакта для связи	Text
Телефон	Телефон контакта для связи	Text
Почта компании	Электронная почта компании	Text
Банк	Обслуживающий банк компании	Text
Расчетный счет	Расчетный счет компании	Text
Корреспондентский счет	Корреспондентский счет компании	Text
БИК	Банковский идентификационный код компании	Text

ИНН	Идентификационный номер налогоплательщика	Text
КПП	Код причины постановки на учет в налоговых органах	Text
ОГРН	Основной государственный регистрационный номер	Text
Наименование организации	Название компании	Text

Таблица 4 – Атрибуты сущности «Комментарии для клиентов»

Название	Имя атрибута	Тип атрибута
ID_комментарии	Первичный ключ таблицы «Комментарии для клиентов»	Integer
ID_клиента	Первичный ключ таблицы «Клиенты»	Integer
Комментарий	Отчет о работе с клиентом	Text
Дата	Дата создания комментария	Text

Таблица 5 – Атрибуты сущности «Командировки»

Название	Имя атрибута	Тип атрибута
ID_командировки	Первичный ключ таблицы «Командировки»	Integer
ID_документации	Первичный ключ таблицы «Документации»	Integer
ID_клиенты	Первичный ключ таблицы «Клиенты»	Integer
Дата создания	Дата создания заявки на командировки	Text
Дата начала	Дата начала командировки	Text
Дата окончания	Дата окончания командировки	Text
Статус командировки	Проверка на создание отчета	Text

Таблица 6 – Атрибуты сущности «Отчеты командировки»

Название	Имя атрибута	Тип атрибута
ID_отчеты	Первичный ключ таблицы «Отчеты»	Integer
ID_командировки	Первичный ключ таблицы «командировки»	Integer
ID_документации	Первичный ключ таблицы «документации»	Integer
Дата	Дата создания отчета о командировке	Text

Таблица 7 – Атрибуты сущности «Документация»

Название	Имя атрибута	Тип атрибута
ID_документации	Первичный ключ таблицы «документации»	Integer
Адрес в системе	Расположение файла в системе	Text
Тип документа	Проверка на источник документа (входящий/исходящий)	Text

Таблица 8 – Атрибуты сущности «Договоры»

Название	Имя атрибута	Тип атрибута
ID_договоры	Первичный ключ таблицы «договоры»	Integer
ID_документации	Первичный ключ таблицы «документации»	Integer

ID_клиенты	Первичный ключ таблицы «клиенты»	Integer
Номер	Номер договора	Text
Дата	Дата составления договора	Text

Таблица 9 – Атрибуты сущности «Спецификации»

Название	Имя атрибута	Тип атрибута
ID_спецификации	Первичный ключ таблицы «спецификации»	Integer
ID_документация	Первичный ключ таблицы «документация»	Integer
ID_приложения	Первичный ключ таблицы «приложения»	Integer
дата	Дата создания спецификации	Text

Таблица 10 – Атрибуты сущности «Приложения»

Название	Имя атрибута	Тип атрибута
ID_приложения	Первичный ключ таблицы «приложения»	Integer
ID_договоры	Первичный ключ таблицы «договоры»	Integer
ID_документации	Первичный ключ таблицы «документации»	Integer
Дата	Дата создания приложения	Text

Таблица 11 – Атрибуты сущности «Расчеты»

Название	Имя атрибута	Тип атрибута
ID_расчеты	Первичный ключ таблицы «расчеты»	Integer
ID_документация	Первичный ключ таблицы «документация»	Integer
ID_клиенты	Первичный ключ таблицы «клиенты»	Integer
Дата	Дата создания расчета	Text

ПРИЛОЖЕНИЕ 3

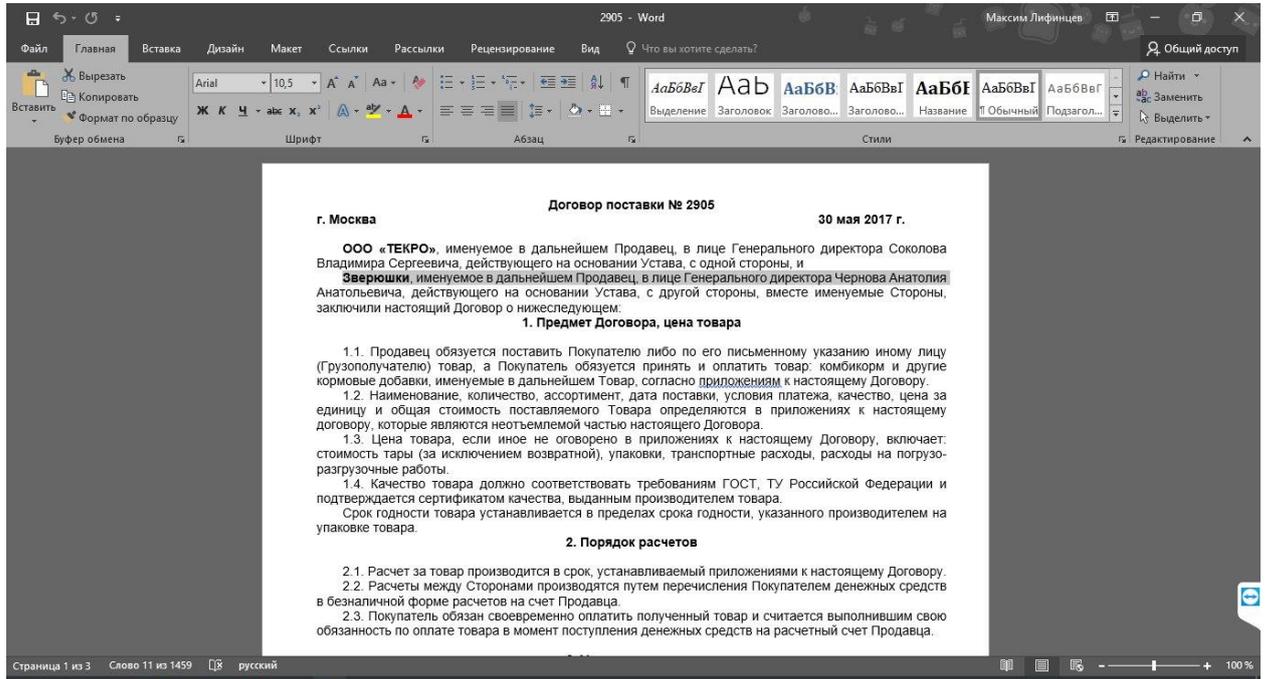


Рис. 3.1. Пример договора

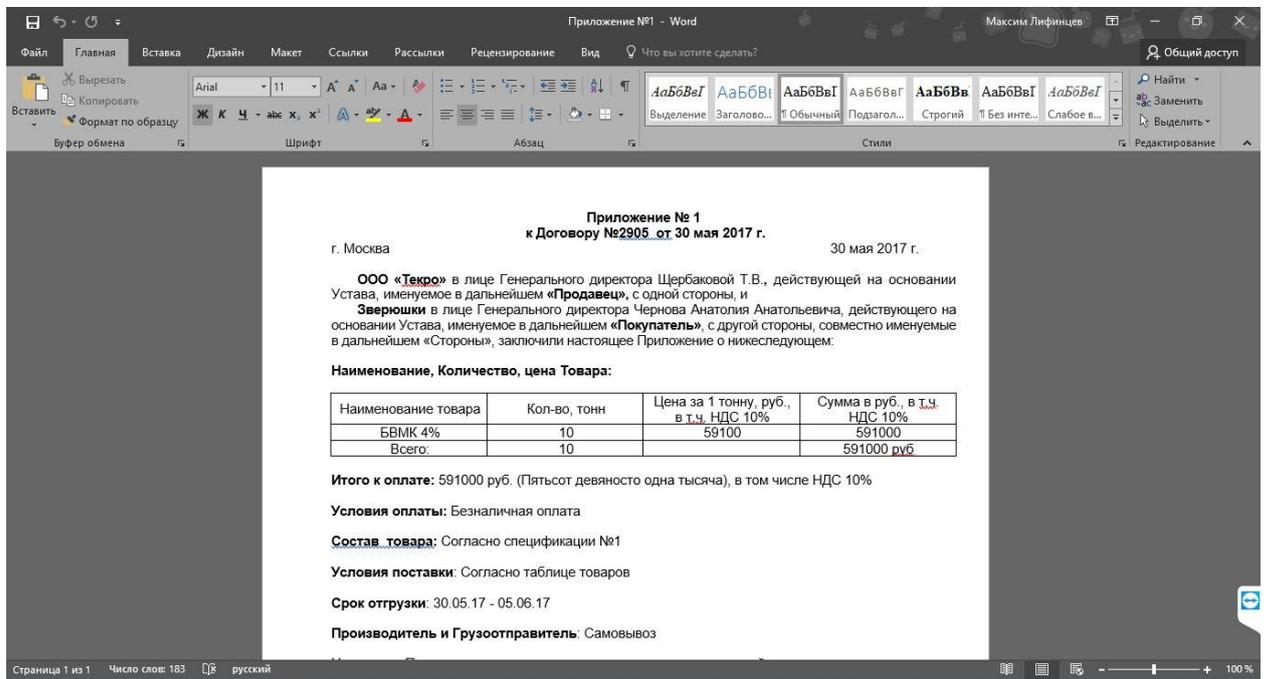
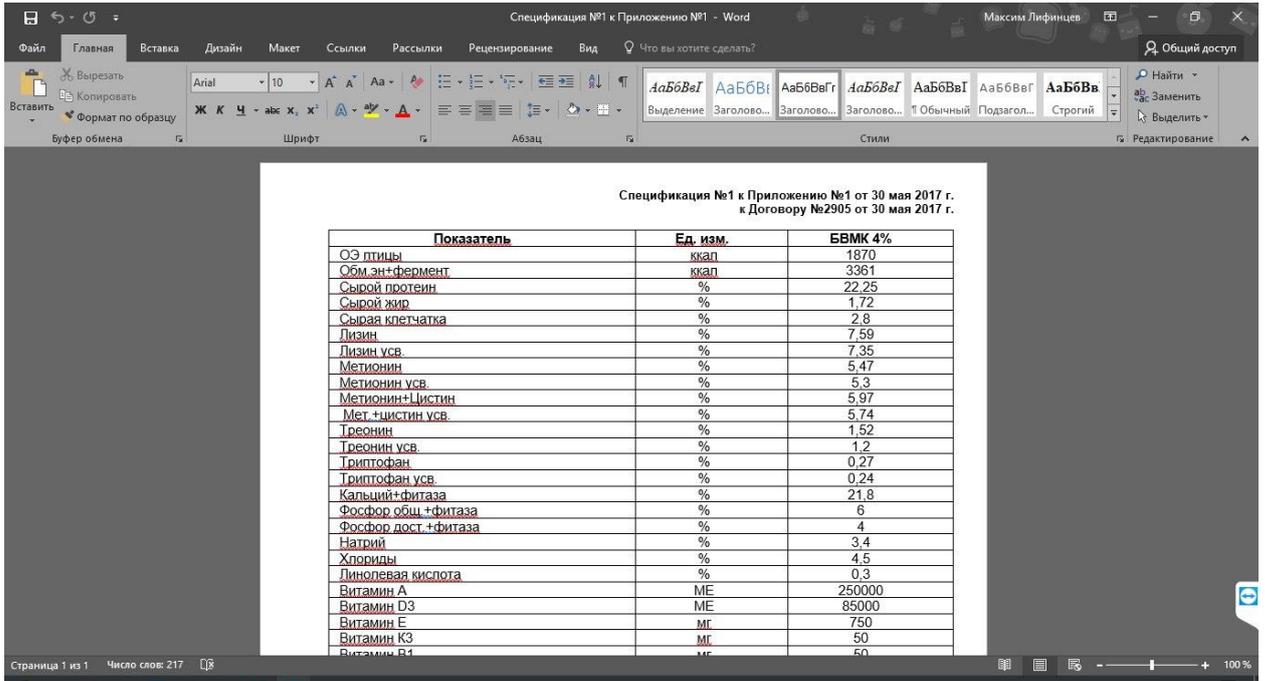


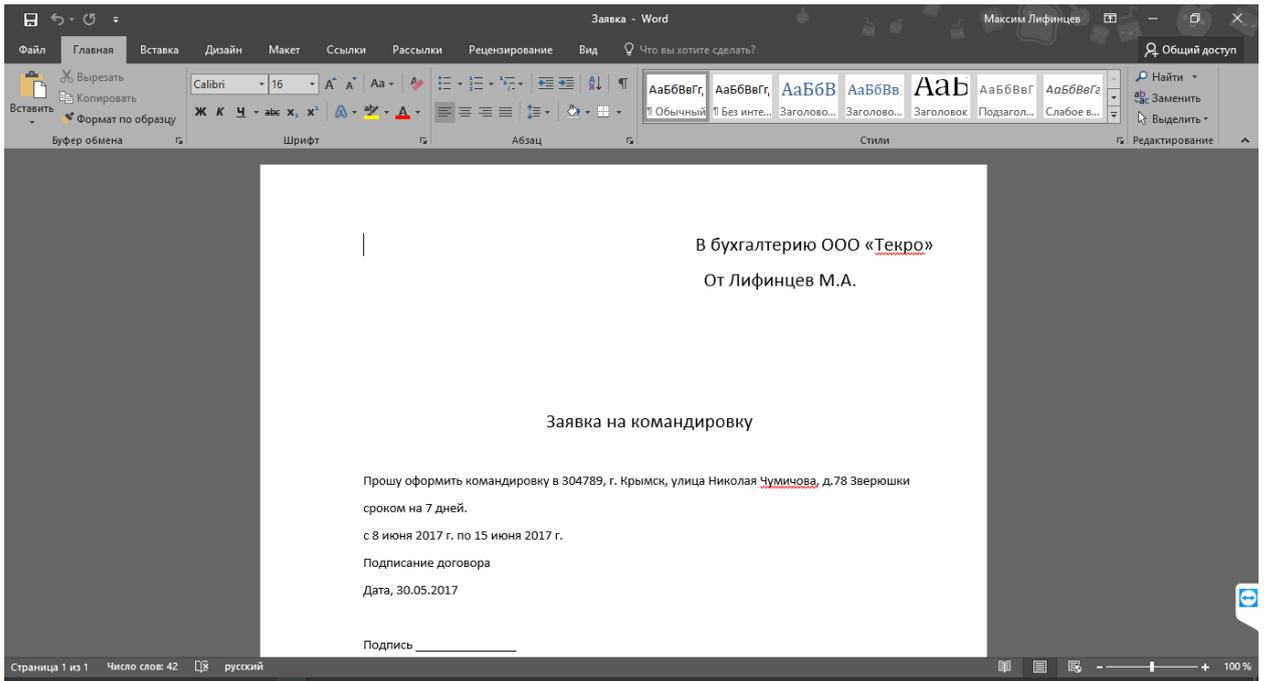
Рис. 3.2. Пример приложения



Спецификация №1 к Приложению №1 от 30 мая 2017 г.
к Договору №2905 от 30 мая 2017 г.

Показатель	Ед. изм.	БВМК 4%
ОЭ птицы	ккал	1870
Обм эн+фермент	ккал	3361
Сырой протеин	%	22,25
Сырой жир	%	1,72
Сырая клетчатка	%	2,8
Лизин	%	7,59
Лизин усв.	%	7,35
Метионин	%	5,47
Метионин усв.	%	5,3
Метионин+Цистин	%	5,97
Мет.+цистин усв.	%	5,74
Треонин	%	1,52
Треонин усв.	%	1,2
Триптофан	%	0,27
Триптофан усв.	%	0,24
Кальций+фитаза	%	21,8
Фосфор общ.+фитаза	%	6
Фосфор дост.+фитаза	%	4
Натрий	%	3,4
Хлориды	%	4,5
Линолевая кислота	%	0,3
Витамин А	МЕ	250000
Витамин Д3	МЕ	85000
Витамин Е	МГ	750
Витамин К3	МГ	50
Витамин В4	МГ	50

Рис. 3.3. Пример спецификации



В бухгалтерию ООО «Текро»
От Лифинцев М.А.

Заявка на командировку

Прошу оформить командировку в 304789, г. Крымск, улица Николая Чумичова, д.78 Зверушки сроком на 7 дней.
с 8 июня 2017 г. по 15 июня 2017 г.
Подписание договора
Дата, 30.05.2017
Подпись _____

Рис. 3.4. Пример заявки на командировку

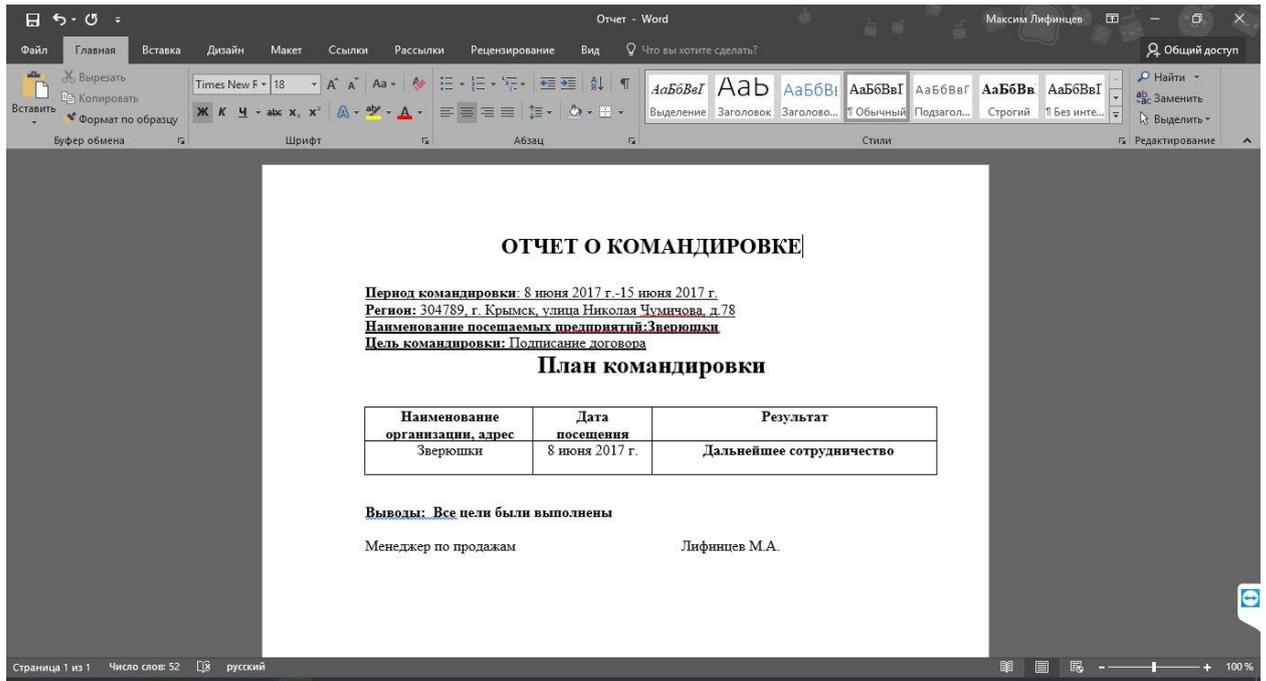


Рис. 3.5. Пример создания отчета о командировке

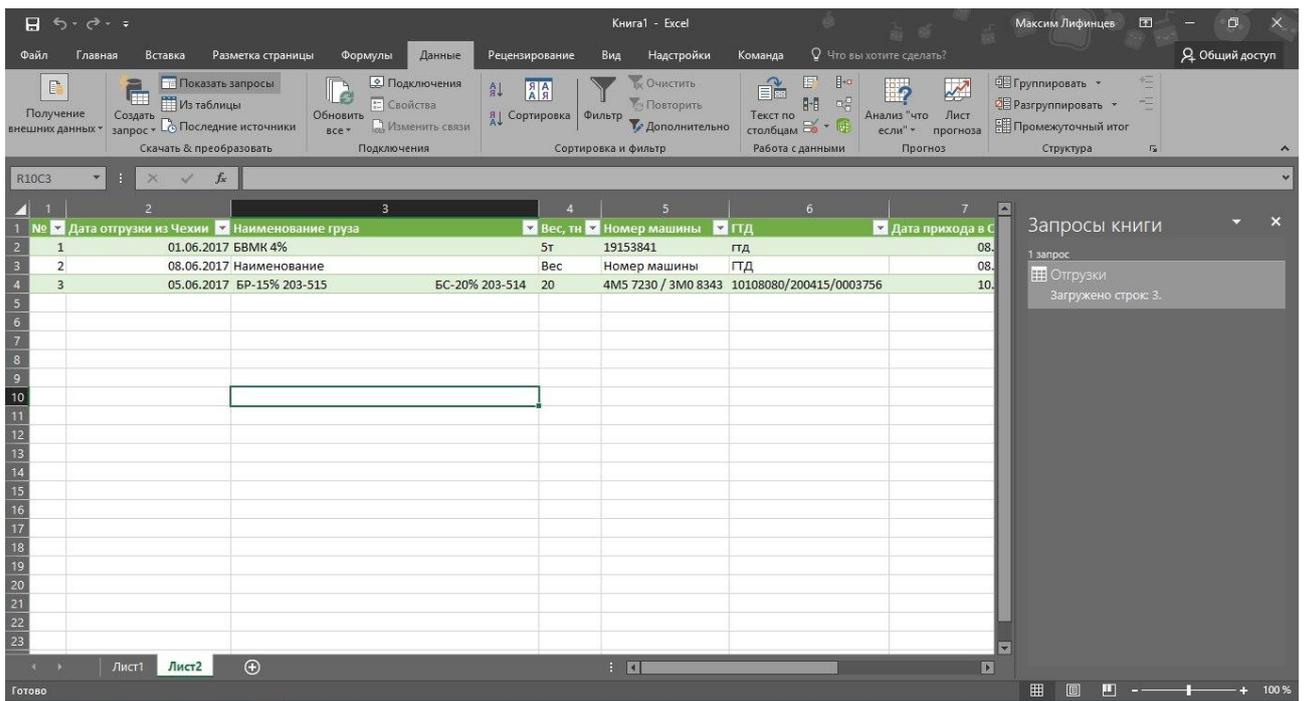


Рис. 3.6. Экспортированные данные

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« » _____ Г.

(подпись)

(Ф.И.О.)