

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа экономики и управления
Кафедра «Информационные технологии в экономике»

РАБОТА ПРОВЕРЕНА

Рецензент, ведущий аналитик
ООО «РитейлИнформ»

_____/ Р.А. Курьянов

«____»_____ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой, д.т.н., с.н.с.

_____/ Б. М. Суховилов

«____»_____ 2018 г.

Автоматизация формирования отчетности о проданных новых автомобилях в ООО
«Тойота Мотор»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ -09.03.03.2018.224.ВКР

Руководитель, доцент

_____/ Е.А. Конова

«____»_____ 2018 г.

Автор,

студент группы ЗэУ-539

_____/ М.В. Рожкова

«____»_____ 2018 г.

Нормоконтролер, доцент

_____/ Е.А. Конова

«____»_____ 2018 г.

Челябинск 2018

АННОТАЦИЯ

Рожкова М.В. Автоматизация предоставления отчетности в ООО Тойота Мотор – Челябинск: ЮУрГУ, Зэу-539, 67 с., 22 ил., 12 табл., библиогр. список – 8 наименований.

Структура выпускной квалификационной работы обусловлена целью и задачами, поставленными и решенными в ходе работы. Работа состоит из введения, трех глав и заключения.

Во введении обосновывается актуальность выбранного направления, описана степень важности существующей проблемы, формулируются цель и задачи, определяется объект и предмет, раскрывается новизна и практическая значимость.

В первой главе «Аналитическая часть» проведен анализ технико-экономической структуры организации, раскрыта его сущность, состав и специфика направления деятельности, проведен анализ существующей информационной системы организации, поставлены задачи для автоматизации процесса формирования отчетности, проведен обзор существующих решений.

Во второй главе «Разработка программного продукта» приведено обоснование выбора инструментов для разработки, произведен сравнительный анализ дополнительных плагинов и библиотек, необходимых для реализации требуемого функционала. Построена схема данных дипломного проекта с описанием таблиц и полей, приведено подробное описание разработанного интерфейса и функционала программного обеспечения.

В третьей главе «Организационно-экономический раздел» рассчитаны затраты на разработку и внедрение программного продукта, анализ экономической эффективности проекта.

Заключение включает в себя подведение итогов, результаты разработки информационной системы. Сформулированы основные выводы и рекомендации, сделанные на основе проведенной работы.

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	3
ВВЕДЕНИЕ	5
1 1 АНАЛИТИЧЕСКАЯ ЧАСТЬ.....	7
1.1 Техничко-экономическая характеристика предметной области.....	7
1.1.1 Краткая характеристика организации ООО «Автомир-74»	7
1.1.2 Анализ информационной системы организации	8
1.1.3 Анализ процесса формирования отчетности	9
1.2 Постановка задачи.....	13
1.3 Анализ существующих программных решений	15
1.3.1 Система Электронного документооборота «DIRECTUM»	17
1.3.2 Система Электронного документооборота «DocsVision»	19
1.3.3 1С: Документооборот 8	20
Выводы по разделу	21
2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА	22
1.4 Выбор средств реализации программного продукта.....	22
1.5 Выбор дополнительных плагинов и библиотек для обеспечения требуемого функционала.....	24
1.5.1 Google Text Recognition API	25
1.5.2 Tesseract.....	26
1.5.3 Anyline	26
1.6 Построение модели базы данных проекта.....	27
1.7 Описание таблиц базы данных	30
1.8 Разработка системы формирования отчетности	35
1.8.1 Интерфейс «Менеджер»	37
1.8.2 Интерфейс «Оформитель»	46

1.8.3	Интерфейс «Логист».....	47
1.8.4	Интерфейс Администратора	51
1.9	Правовые аспекты и меры безопасности	54
	Выводы по главе	Ошибка! Закладка не определена.
3	ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ.....	60
1.10	Расчет затрат	60
1.11	Анализ убытков	61
	Выводы по разделу	61
	ЗАКЛЮЧЕНИЕ	62
	ПРИЛОЖЕНИЕ А Бланк согласия на обработку персональных данных	63
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК	64

ВВЕДЕНИЕ

Цель работы: разработка проекта по автоматизации документооборота департамента отдела оформления и предоставления отчетности о проданных автомобилях в ООО Тойота-Мотор для получения компенсации денежных скидок.

Актуальность разработки обусловлена сложностью и монотонностью составления отчетности, непрерывным и не организованным взаимодействием между отделами организации для сбора информации, не структурированным способом хранения электронных документов и строгой проверкой предоставляемых документов на правильность. Существуют огромные финансовые риски, которые организация несет вследствие наличия ошибок, которые являются следствием наличия большей доли человеческого фактора.

В соответствии с поставленной целью сформулированы следующие задачи разработки системы.

- Изучение предметной области учета продаж новых автомобилей, принятого на предприятии стандарта и механизма документооборота, установившейся системы отчетности.
- Комплексный анализ основной существующей информационной системы предприятия.

- Анализ существующих разработок для решения поставленной цели.
- Определение списка задач для автоматизации.
- Разработка информационного обеспечения задачи.
- Разработка программного проекта решения задачи.
- Тестирование полученного проекта, устранение недостатков.

Теоретико-методической основой исследования являются современные подходы и разработки в области автоматизации деятельности предприятия.

Объектом автоматизации является предприятие ООО «Автомир-74».

Оценка актуальности разработки данного проекта включает в себя несколько пунктов.

- Оценку количества продаж новых автомобилей.
- Оценку удовлетворённости дистрибьютора предоставляемой отчетностью дилерского центра.
- Оценку скорости и трудозатрат на формирование отчетности.
- Оценку материально-технического оснащения дилерского центра.
- Оценку затрат на внедрение данной разработки.

Практическая значимость выпускной квалификационной работы заключается в том, что проект может быть использован холдингом предприятий и организаций при работе в жесткой конкурентной среде, а также для совершенствования системы ведения отчетности, как важной составляющей эффективного менеджмента. Использование разработанного программного обеспечения позволит решить основную массу проблем и неудобств, с которым на повседневной основе сталкиваются сотрудники организации, повысить оперативность и качество формирования отчетности, сократить экономические риски.

1 1 АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1 Технико-экономическая характеристика предметной области

~~1.1.1~~

1.1.21.1.1 Краткая характеристика организации ООО «Автомир-74»

ООО «Автомир-74» – официальный дилер автомобилей марки Toyota, является обособленным подразделением группы компаний «Автомир», которая в настоящее время является одной из самых крупных автомобильных холдингов на российском рынке.

Организация существует в городе Челябинске с 1993г. Основные виды деятельности предприятия – торговля автотранспортными средствами, их техническое обслуживание и ремонт. Краткий перечень направлений деятельности предприятия приведен ниже.

- Продажа новых автомобилей марки Toyota и автомобилей прочих марок с пробегом.
- Гарантийное и сервисное обслуживание автомобилей марки Toyota.
- Предоставление услуг по кузовному и слесарному ремонту.
- Тюнинг, установка охранных устройств и дополнительного оборудования.
- Обмен и выкуп подержанных автомобилей.
- Предоставление услуг по автострахованию совместно со страховыми компаниями.
- Предоставление кредитных и лизинговых программ совместно с банками-партнерами.

Штат подразделения насчитывает около 150 сотрудников и имеет 4 основных департамента. Директор обособленного подразделения является главным лицом филиала и находится в непосредственном подчинении директору дирекции группы компаний «Автомир» и вышестоящему руководству головного офиса в г.

Москва. Управление каждым департаментом осуществляет руководитель. Деятельность каждого департамента, а также самого директора регламентируется приказами и распоряжениями, утверждаемыми на более высоком уровне управления. Связь (в том числе и донесение приказов, распоряжений) между сотрудниками организации, департаментами и филиалами осуществляется средствами электронной рассылки и онлайн переговоров. Каждое рабочее место имеет доступ к корпоративному web-хранилищу, содержащему электронный архив приказов и распоряжений.

1.1.31.1.2 Анализ информационной системы организации

Каждое рабочее место в организации оснащено необходимым офисным набором вычислительной техники: персональные компьютеры, копировальный аппарат и сканер, утилизатор документов. В каждом отделе установлены телефоны, проведена сеть интернет, доступ к сети интернет и к локальной сети имеется на каждом компьютере. Необходимо отметить, что на предприятии повышены меры информационной безопасности: каждый ПК закреплен за конкретным работником, и вход в систему осуществляется через личную учетную запись с паролем. Данные, передаваемые по электронной почте, защищены с помощью PGP-инструментов.

Программное обеспечение, используемое в текущей работе и составляющее основную информационную систему организации.

- 1С: Предприятие 8.2. Конфигурация МТСК используется для оформления сделок и пакетов документов по продаже, обмену, выкупу автомобилей. Конфигурация МУВТ используется для управления взаимодействия с клиентами.

- 1С: Предприятие 7. Конфигурация 1С: Сервис для оформления пакета документов на приобретенное дополнительное оборудование, подарочное дополнительное оборудование и работы по его установке.

- Операционная система Windows 7 Enterprise.

- MS Office. Для осуществления оперативной коммуникации между сотрудниками компании используется Microsoft Outlook и Microsoft Link.
- Интернет браузеры Google Chrome и Internet Explorer.

В процесс автоматизации вовлечены все процессы, выполняемые на предприятии. Единственным исключением является процесс формирования и контроль отчетности о проданных автомобилях представительству ООО Тойота Мотор.

1.1.41.1.3 .Анализ процесса формирования отчетности

Продажа новых автомобилей марки Toyota является ключевым направлением деятельности предприятия.

Каждый месяц по результатам работы дилер отчитывается своему дистрибьютору по каждой оформленной сделке по продаже новых автомобилей.

Сделка по продаже нового автомобиля клиенту осуществляется по нескольким схемам:

- выдача нового автомобиля, купленного за наличные средства;
- выдача нового автомобиля, купленного за кредитные средства;
- выдача нового автомобиля по системе обмена старого автомобиля с доплатой наличными средствами (Trade-in;)
- выдача нового автомобиля по системе обмена старого автомобиля с доплатой кредитными средствами (Trade-in- кредит).

Схема сделки по продаже автомобиля является критерием, влияющим на конечный оформляемый пакет документов, и, соответственно на процесс формирования отчета по этой сделке.

Для оформления продажи автомобиля менеджером департамента оформления продаж готовится стандартный набор документов: договор купли продажи с покупателем, акт приема-передачи автомобиля, запись в ПТС о новом собственнике (паспорт транспортного средства). При оформлении сделки по схеме трейд-ин пакет документов дополняется договором купли-продажи, актом приема пере-

дачи на старый автомобиль, в ПТС компания «Автомир-74» вписывается, как новый собственник.

В процессе формирования отчетности участвуют менеджер департамента оформления продаж, менеджер департамента Trade-in и логист. Конечную ответственность за достоверность и правильность предоставляемых данных несет логист. Логист занимается отправкой конечного файла отчета представительству Toyota Motor Russia.

Отчетность предоставляется не позднее 5 числа текущего месяца за базовый период, равный полному предыдущему месяцу. Файл отчета представляет собой zip архив, включающий набор файлов в формате .pdf по каждой оформленной сделке.

Один файл включает набор документов по одной сделке. Имя каждого файла включает в себя VIN-номер нового автомобиля, наименование модели автомобиля, а также фамилию и инициалы собственника. Например, при продаже автомобиля модели RAV4, VIN – JTMDR3DJ3456365 Иванову Ивану Ивановичу файл будет называться R343_RAV4_JTMDR3DJ3456365 _Иванов И.И, где R343 – это код дилера, который является неизменным.

Для четкого разделения и разграничения должностных обязанностей, а в последующем формирования различных ролей доступа к функциональным модулям системы необходимо провести анализ организационной структуры предприятия и определить ключевых пользователей, отвечающих за определенные степени протекающих бизнес-процессов.

На рисунке 1.1 приведена UML диаграмма сущностей, описывающая сотрудников организации, участвующих в процессе формирования отчетности.

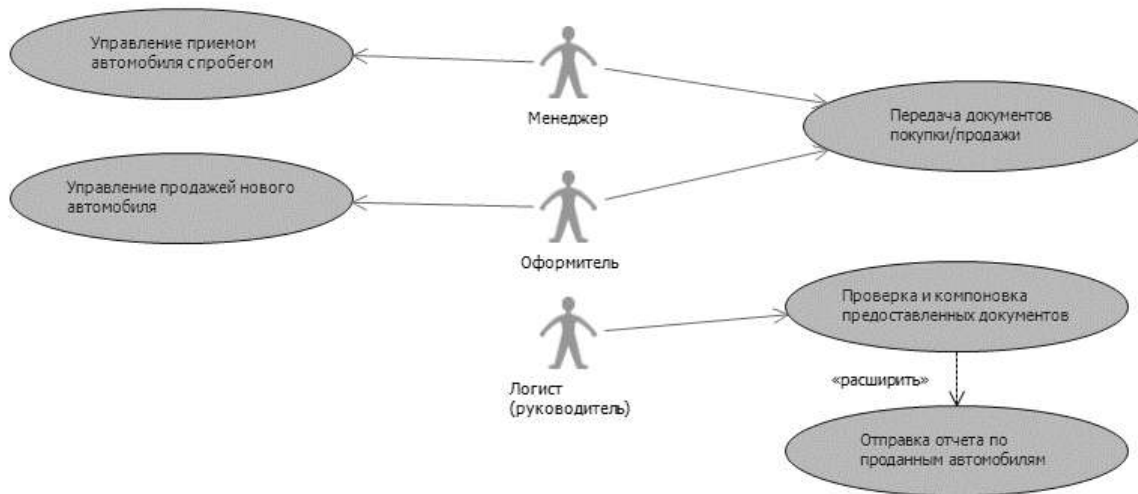


Рисунок 1.1 – Диаграмма сущностей процесса формирования отчетности 0.1

Диаграмма потоков данных (DFD) на рисунке 1.2 описывает процесс формирования отчетности в том виде, как это происходит в настоящее время.

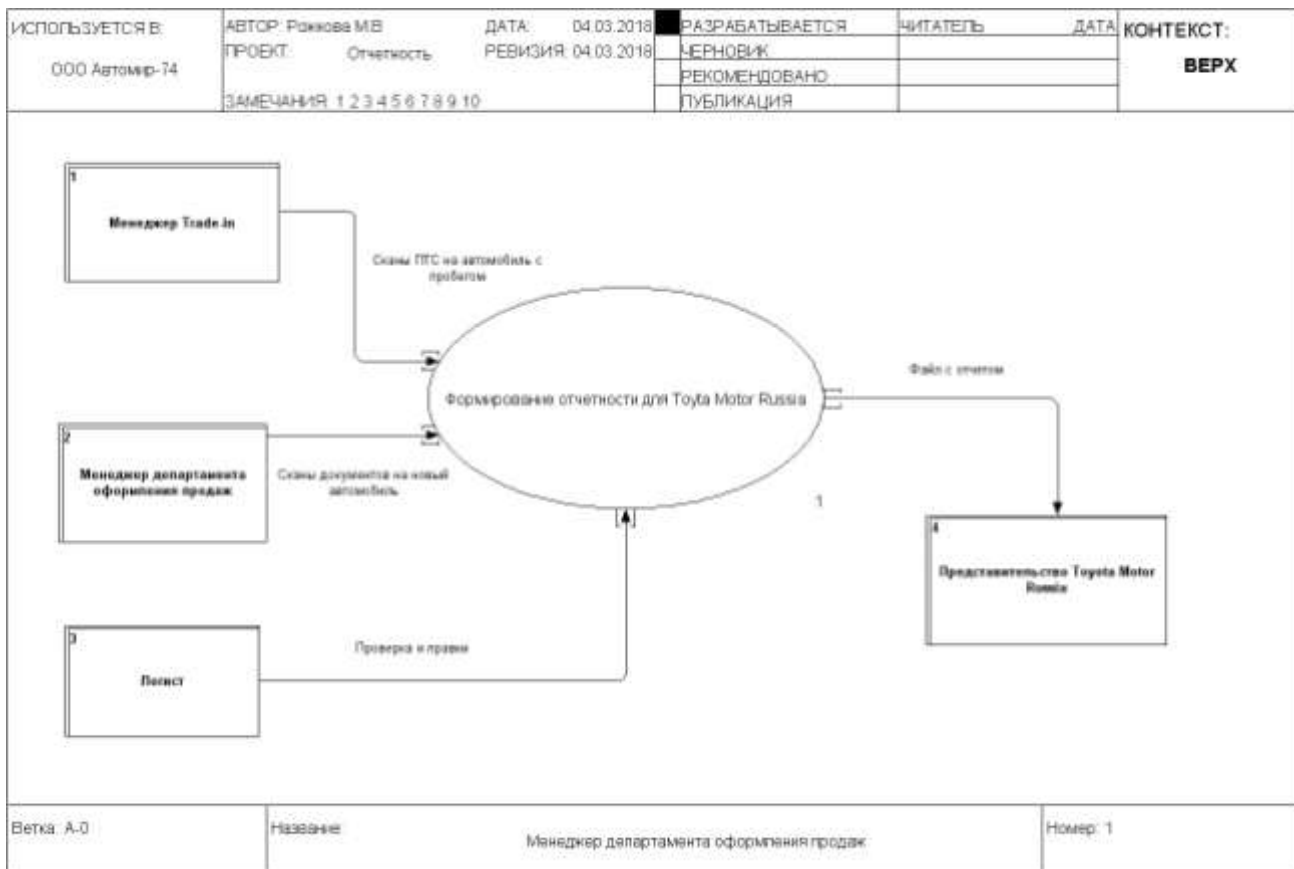


Рисунок 1.2 – Диаграмма потоков данных процесса формирования отчетности

На рисунке 1.3 приведена диаграмма второго уровня. **0.2**

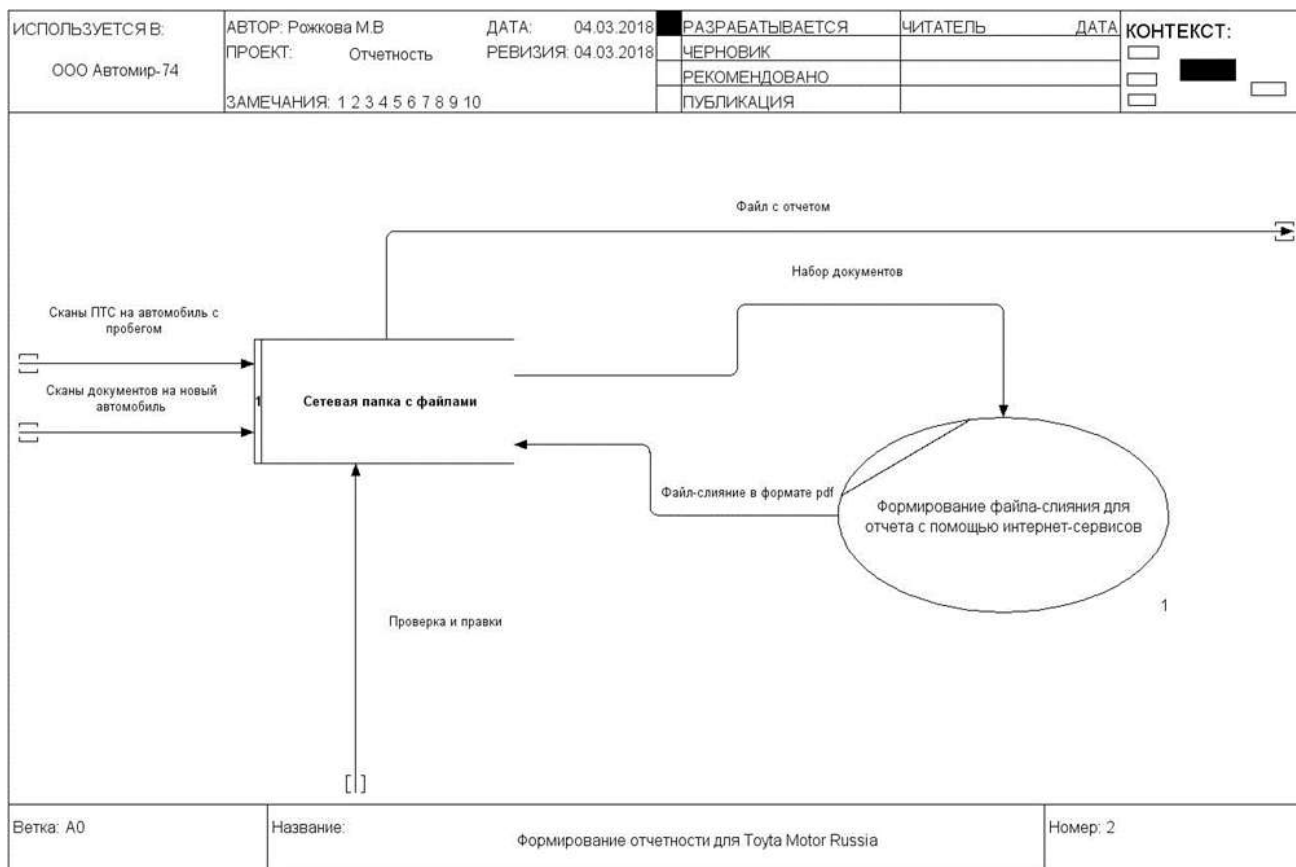


Рисунок 1.3 – Диаграмма потоков данных процесса формирования отчетности **0.3**

Из представленных диаграмм видно, что формирование отчетности на данное время происходит вручную. Для хранения скан-копий документов по каждой сделке используется выделенная папка на сетевом диске. Также скан-копии могут быть высланы логисту по электронной почте.

Для формирования файла-слияния (единый файл в формате .pdf, состоящий из последовательности документов по новому автомобилю и автомобилю с пробегом, если он есть в наличии) по каждой сделке логист пользуется различными интернет-сервисами, предоставляющими услуги по объединению и разделению .pdf файлов. Проверку данных в начальных файлах логист осуществляет самостоятельно и исключительно визуально. Логист должен проверить цены в договорах, правильность даты, параметров автомобиля (VIN- номер, номер двигателя и т.д.). Затем файл-слияние сохраняется в ту же сетевую папку, хранящую на-

чальные файлы. В конце периода логист вручную формирует архив с документами по каждой сделке.

После исследования данного алгоритма выявлены его недостатки.

- Отсутствие удобного хранилища для файлов, присылаемых сотрудниками по каждой сделке, отсутствие структуры и принадлежности набора файлов к одной сделке.

- Сложности при идентификации нужного файла, так как используются длинные имена, присваиваемые сканером.

- Использование сторонних ненадежных интернет-сервисов для разделения, объединения файлов. Большинство из них значительно ухудшают качество файла при обработке. Также существует вероятность того, что тот или иной интернет-сервис может прекратить свою работу.

Для исключения существующих в настоящее время неудобств при формировании отчетности, необходимо сформировать цели и комплекс задач по его автоматизации.

1.2 ~~Формирование целей и задач для автоматизации~~ Постановка задачи

Сформированы следующие ~~цели~~ задачи для ~~ее~~ автоматизации процесса формирования отчетности.

- Исключение сложности и монотонности выполняемых операций, а также сокращение доли человеческих манипуляций с данными и файлами.

- Настройка организованного взаимодействия между отделами организации для сбора информации по предоставлению отчетности.

- Структуризация хранения электронных документов и проверка предоставляемых документов на правильность.

- Сокращение финансовых рисков организации.

На рисунках 1.4 и 1.5 представлены DFD диаграммы потоков данных, описывающие процесс предложенного программного формирования отчетности.

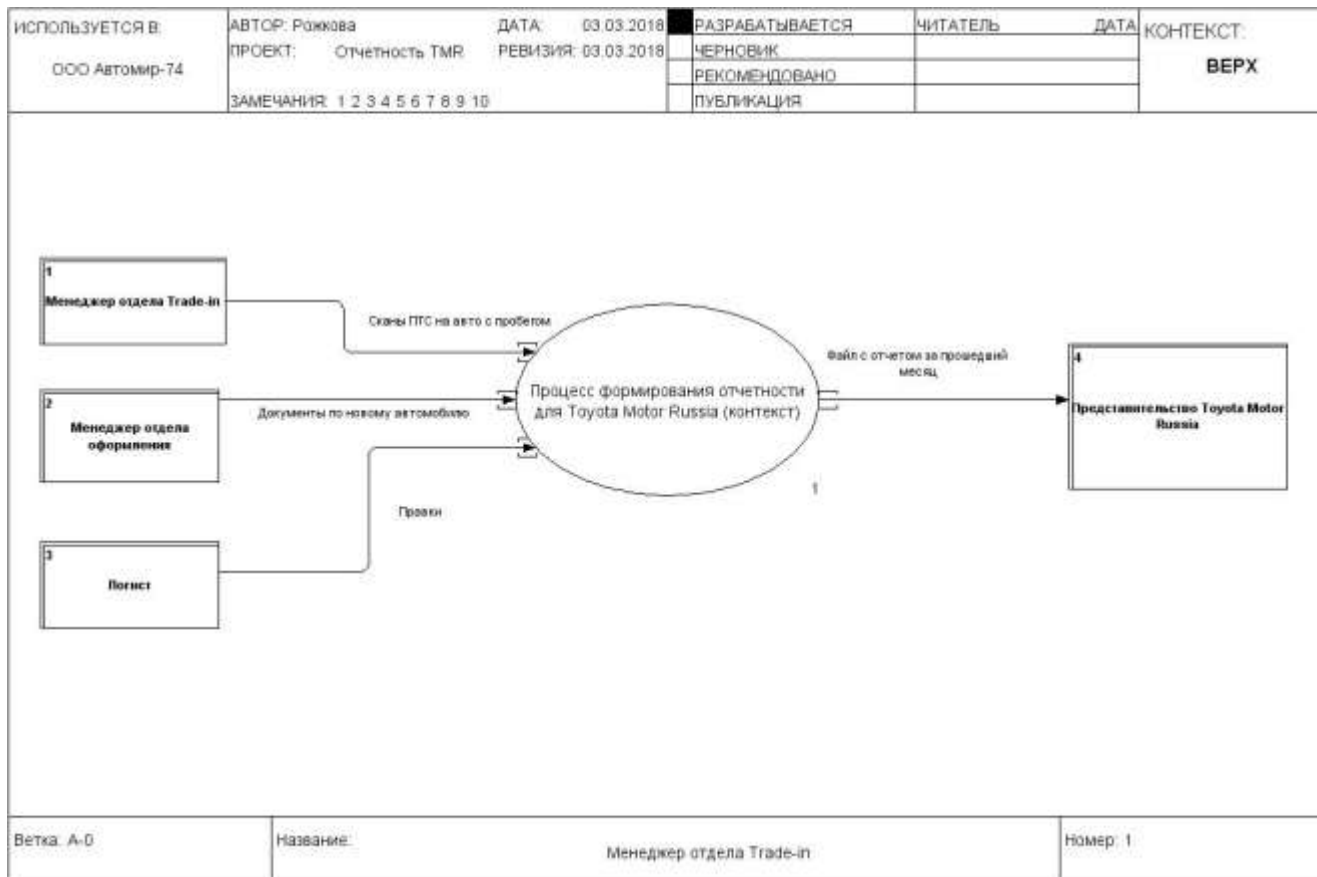


Рисунок 1.4 – Диаграмма потоков данных разработанной системы

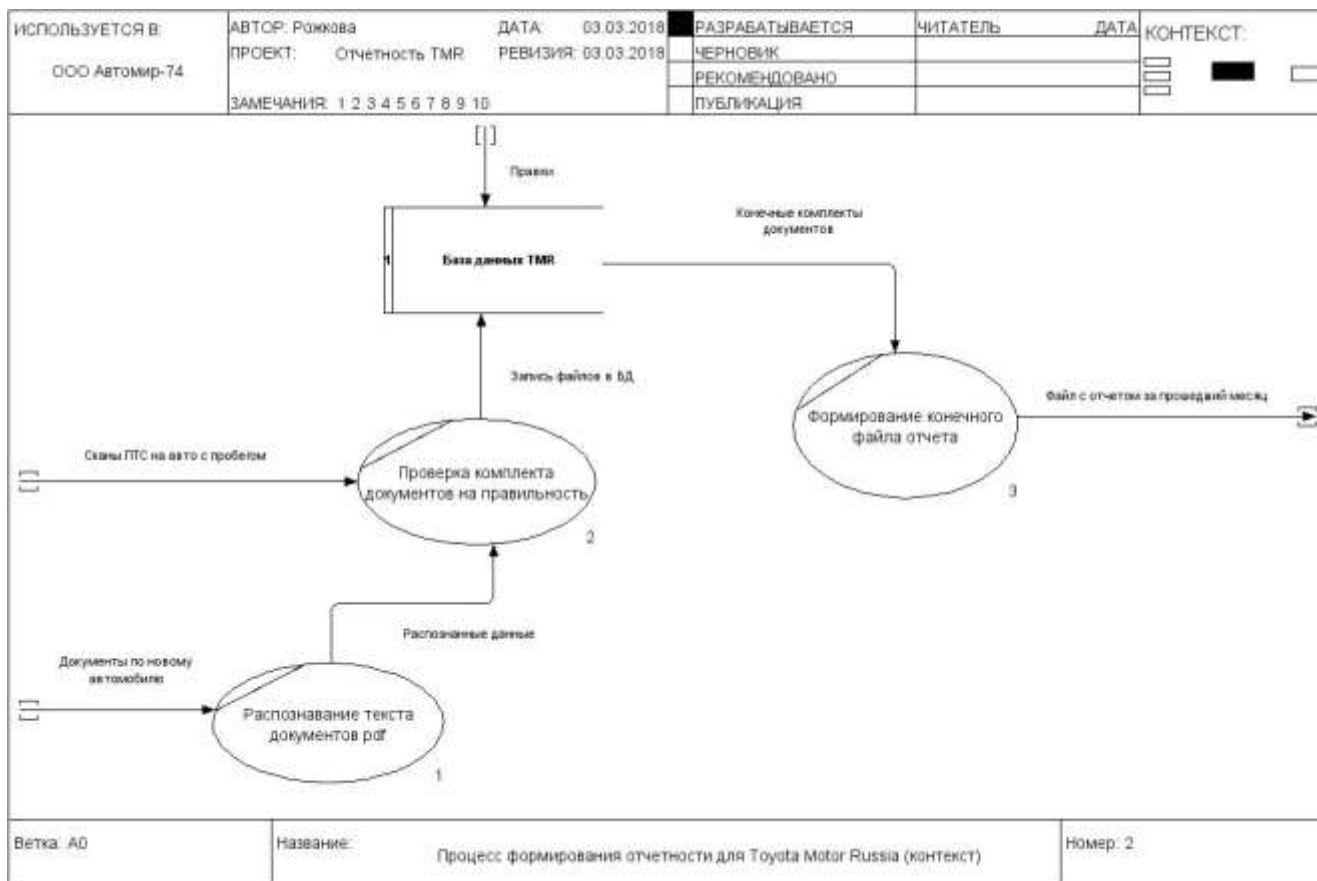


Рисунок 1.5 – Диаграмма потоков данных разработанной системы

Для упорядочивания имен и структурированного хранения файлов по каждой сделке используется база данных.

Для сокращения числа ошибок, возникающих вследствие визуальной проверки документов, проверка осуществляется программно с помощью автоматического распознавания текста.

Исключена необходимость использования сторонних интернет-сервисов. Формирование файла-слияния в формате .pdf осуществляется программно.

В разработанном программном продукте должно быть учтено то, что форма договора, а также требования по предоставляемому набору документов в отчете могут измениться. Для этого обеспечен функционал по формированию правил распознавания текста в загружаемых документах.

1.3 Анализ существующих программных решений

Рассмотрим существующие программные продукты, применяемые для решения подобных задач по предоставлению отчетности и проанализируем достоинства и недостатки существующих систем. Учитывая требуемый функционал для нового программного обеспечения, особое внимание в анализе существующих разработок уделено системам электронного документооборота (СЭД).

Существует множество определений того, что такое электронный документ, СЭД и т.д. Ниже приведены самые емкие из определений.

Электронный документ – это некий набор информации (текст, изображение, звукозапись), сохраненный на компьютере. Этот набор информации сопровождается карточкой с атрибутами, подобно тому, как книги в библиотеке сопровождаются картотекой. По атрибутам документ можно быстро найти.

Workflow (поток работ) – это последовательность действий сотрудников в рамках определенного бизнес-процесса. Например, последовательность действий – это получение документа, регистрация документа, рассмотрение документа, исполнение документа, а бизнес-процесс – работа с обращениями граждан.

Электронный документооборот (ЭДО) – это способ организации работы с документами, при котором основная масса документов используется в электронном виде и хранится централизованно.

Система электронного документооборота (СЭД) – это компьютерная программа (программное обеспечение, система), которая позволяет организовать работу с электронными документами (создание, изменение, поиск), а также взаимодействие между сотрудниками (передачу документов, выдачу заданий, отправку уведомлений и т.п.).

К преимуществам использования систем электронного документооборота можно отнести следующие.

- Прозрачность бизнес-процессов. Система обеспечивает возможность отслеживания этапов выполнения бизнес-процессов, что делает всю деятельность в организации абсолютно прозрачной для руководства.

- Повышение исполнительской дисциплины. Предоставляя полный контроль всех этапов работ для руководства, система напрямую влияет на исполнительскую дисциплину сотрудников.

- Сокращение затрат времени руководителей и сотрудников. Использование системы сокращает временные затраты практически на все рутинные операции с документами (создание, поиск, согласование и т.д.). Кроме того, происходит ускорение документооборота и, как следствие, всех процессов в организации.

По заявлениям исследовательской компании Gartner, к ЕСМ могут быть отнесены системы, поддерживающие не менее трех из шести функций.

- Управление документами (выписка/возврат, контроль версий, безопасность, группировка документов и т.д.).

- Совместная работа над общими документами и поддержка проектных команд.

- Сканирование документов и управление образами бумажных документов.

- правление записями для долгосрочного архивного хранения, автоматизации правил и нормативов хранения, гарантирование соответствия записей законодательству и регулирующим правилам.

- Workflow для поддержки бизнес-процессов, маршрутизации контента, назначение рабочих задач и состояний, трассировка маршрутов и контроль исполнения.

- Управление веб-контентом для автоматизации публикаций, управление динамическим контентом и взаимодействием пользователей для этих задач.

Приведены наиболее популярные на российском рынке Системы Электронного документооборота.

1.3.1 Система Электронного документооборота «DIRECTUM»

Система DIRECTUM поддерживает полный жизненный цикл управления документами, обеспечивает организацию и контроль деловых процессов на основе технологии Workflow.

Состав системы DIRECTUM.

- Управление электронными документами. Создание и хранение различных неструктурированных документов (тексты Microsoft Word, таблицы Microsoft Excel, CorelDraw, видео и пр.).

- Поддержка версий документов и ЭЦП; структурирование документов по папкам; назначение прав доступа на документы; история работы с документами; полнотекстовый и атрибутивный поиск документов.

- Управление деловыми процессами. Поддержка процессов согласования и обработки документов на всех стадиях их жизненного цикла (docflow); выдача электронных заданий и контроль их исполнения; взаимодействие между сотрудниками в ходе бизнес-процессов; поддержка свободных и жёстких маршрутов (Workflow).

– Управление договорами. Организация процесса согласования и регистрации договоров и сопутствующих документов, а также оперативной работы с ними (поиск, анализ, редактирование и т.д.).

– Управление совещаниями. Организация подготовки и проведения совещаний (согласование места и времени, состава участников, повестки).

– Канцелярия. Регистрация бумажных документов в соответствии с требованиями ГСДОУ¹; ведение номенклатуры дел с гибкими правилами нумерации; рассылка и контроль местонахождения бумажных документов; организация обмена электронными документами с ЭЦП с другими организациями.

– Управление взаимодействием с клиентами. Ведение единой базы организаций и контактных лиц; ведение истории встреч, звонков и переписки с клиентами; сопровождение процесса продаж в соответствии с регламентированными стадиями; планирование маркетинговых мероприятий; анализ эффективности продаж и маркетинговых воздействий.

Недостатки, выявленные по отношению к области применения (ООО Автомир-74):

- нагруженный интерфейс;
- необходимость обучения персонала работе в программном обеспечении (дополнительные расходы организации);
- стоимость лицензии (112 100 рублей за пакет Standard);
- минимальное количество приобретаемых рабочих мест: 20-25;
- есть языковые проблемы;
- низкое быстродействие просмотра;
- Отсутствие потребности использования большинства функциональных блоков данного продукта;
- отсутствие функционала по настройке правил распознавания текста.

¹ Государственная система документационного обеспечения управления

1.3.2 Система Электронного документооборота «DocsVision»

Система Docsvision – программный продукт, предназначенный для создания автоматизированных корпоративных решений по управлению документами и бизнес-процессами. Включает предметно-ориентированную платформу с открытыми интерфейсами прикладного программирования для разработки заказных приложений и готовые типовые приложения с возможностями параметрической настройки.

Состав системы DocsVision.

– Делопроизводство. Все для организации делопроизводства: от упрощенного до полнофункционального, по стандартам и методикам Росархива². Регистрация, рассмотрение, резолюции, согласования, контроль исполнительской дисциплины и номенклатура дел.

– Договорной документооборот. Решения для работы с поставщиками и покупателями – разработка, согласование и учет договоров, технических заданий, смет и других договорных документов, переписка, пролонгация, контроль исполнения.

– Оперативное управление. Организация управления подразделениями, организация совещаний.

– Электронный архив. Регистрация документов любыми способами – ручным вводом, сканированием с обычного или мощного потокового сканера.

Недостатки, выявленные по отношению к области применения (ООО Автомир-74):

- нагруженный интерфейс;
- необходимость обучения персонала работе в программном обеспечении (дополнительные расходы организации);
- стоимость лицензии (95 000 рублей за пакет Базовый);

² Федеральное архивное агентство (Росархив) — федеральный орган исполнительной власти, находящийся в ведении Президента Российской Федерации. Осуществляет функции по выработке и реализации государственной политики и нормативно-правовому регулированию в сфере архивного дела и делопроизводства, а также по контролю, оказанию государственных услуг и управлению федеральным имуществом в этой сфере.

- минимальное количество приобретаемых рабочих мест: 20;
- отсутствие функционала по настройке правил распознавания текста;
- отсутствие потребности использования большинства функциональных блоков данного продукта.

1.3.3 1С: Документооборот 8

Система обеспечивает автоматизацию полного цикла работы с документами, позволяет упорядочить взаимодействие между сотрудниками и осуществлять контроль использования рабочего времени.

Учёт документов реализован в соответствии с положениями действующей нормативной документации.

Программа обеспечивает многопользовательскую работу как в локальной сети, так и через интернет (в том числе через веб-браузеры). Система отличается большой гибкостью, высокой степенью детализации сведений о хранящихся данных и широким спектром возможностей. Позволяет повысить эффективность управления рабочим временем, стандартизировать процессы, обеспечить полный контроль и сохранность документации и любой иной необходимой информации. функциональность системы постоянно расширяется.

Состав системы 1С: Документооборот 8.

- Учет и хранение документов.
- Регистрация входящих и исходящих документов.
- Регистрация внутренних документов (инструкций, служебных записок, поручений и т. д.).
- Работа с документами любых типов (тексты, изображения, видеофайлы, аудиофайлы, офисные документы, архивы, приложения и пр.).
- Учёт в разрезе видов документов.
- Хранение документов в каталогах любой структуры по усмотрению пользователя.
- Загрузка документов из электронной почты, со сканера.

- Поддержка бумажного делопроизводства (ведение номенклатуры дел, контроль перемещений документов и т. д.).
- Учёт обращений граждан.
- Учёт персональных данных.
- Работа с документами.
- Работа с документами на всех стадиях жизненного цикла (от черновиков до утверждения или уничтожения).
- Механизм резолюций.
- Быстрый поиск, в том числе полнотекстовый.
- Возможность коллективной работы с одним и тем же файлом (согласование, контроль исполнения, принятие и т. д.).
- Формирование задач исполнителям.

Недостатки, выявленные по отношению к области применения (ООО Автомир-74):

- нагруженный интерфейс;
- необходимость обучения персонала ;
- стоимость лицензии (95 000 рублей за пакет ПРОФ);
- минимальное количество приобретаемых рабочих мест: 20;
- отсутствие потребности использования большинства функциональных блоков данного продукта.
- отсутствие функционала по настройке правил распознавания текста.

Выводы по разделу

Приведена технико-экономическая характеристика предметной области, изучена организационная структура компании ООО Автомир-74, проведен обзор и анализ информационной системы предприятия, выявлены процессы, которые до сих пор не были автоматизированы, приведена аргументация необходимости изменения этих процессов, а также поставлены цели и задачи для их автоматизации.

Принято решение о разработке собственного программного продукта, соответствовать следующим требованиям.

- Простой и интуитивно понятный интерфейс.
- Возможность коллективной работы.
- Минимум затрат на внедрение проекта и обучение персонала работе с программным продуктом.

2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

1.4 Выбор средств реализации программного продукта

В рамках дипломного проекта, исходя из поставленных целей для разработки принято решение построить многопользовательское клиент-серверное приложение, осуществляющее документооборот в реальном времени. Учитывая имеющуюся информационную систему на предприятии и уже установленное на рабочих местах оборудование и программное обеспечение, необходимо создать современный программный продукт, избегая дополнительных трат на покупку новых комплектующих для компьютеров или дополнительных программных продуктов. Также необходимо уделить особое внимание надежности приложения и простоте его интерфейса.

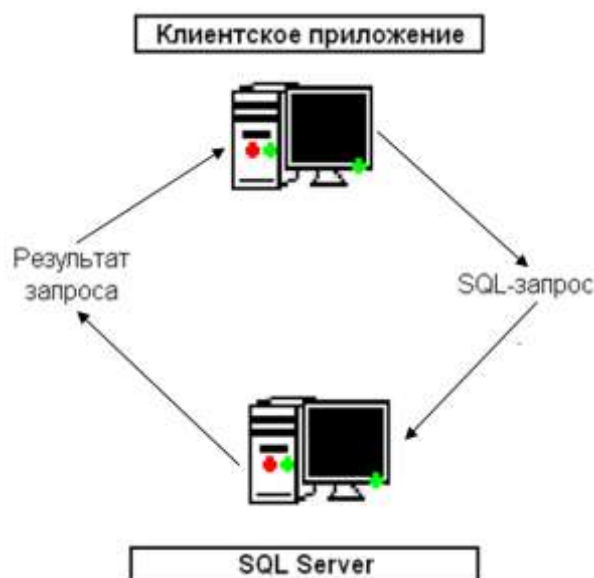
Для разработки выбраны следующие инструменты и ПО.

СУБД Microsoft SQL Server

Предназначен для работы в клиент-серверных системах. СУБД отвечает за поддержку отношений между данными, согласованность данных, а также отказоустойчивость. В клиент-серверной архитектуре клиентское приложение обычно выполняется на персональном компьютере пользователя, который является частью локальной сети. Локальная сеть применяется для соединения с ядром БД SQL Server, а также для передачи результатов и запросов.

Данная СУБД на текущий момент используется организацией Автомир-74 и полностью подходит для разработанного проекта – позволяет реализовывать хранимые процедуры и различные функции приложениям, разработанным на языках платформы .NET. В SQL Server предусмотрена безопасная и надежная работа многих пользователей со многими приложениями на одном компьютере.

На рисунке 2.1 приведена схема взаимодействия клиента с сервером.



– Рисунок 2.1 – Архитектура клиент-серверного приложения [2.1](#)

Среда Microsoft Visual Studio 2013 и язык программирования C#

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, добавление новых наборов инструментов, например, для редактирования и визуального проектирования кода. На данный продукт действует лицензия, предоставляемая Microsoft Office студентам высших учебных заведений.

Язык программирования C# выбран как родной язык среды Microsoft Visual Studio.

.NET Framework

Платформа для разработки и развертывания различных типов приложений с использованием единых стандартов и библиотек. Такой подход к разработке

программного обеспечения (ПО) позволяет упростить процесс обучения специалистов, сократить время ввода нового участника в проект, эффективно использовать накопленные специалистами знания, опыт и разработанные ранее компоненты. Создание .NET приложений возможно на любом .NET Framework-совместимом языке программирования, но на сегодняшний момент в большинстве случаев используется язык программирования C#. Такой выбор во многом обусловлен тем, что язык C# создан компанией Microsoft специально для платформы .NET и унаследовал все лучшее от C++, Java и других объектно-ориентированных языков.

1.5 Выбор дополнительных плагинов и библиотек для обеспечения требуемого функционала

В соответствии с поставленными целями и задачами, разработанное приложение должно обладать следующим функционалом:

- возможность формирования файла слияния из двух и более pdf файлов с возможностью добавления новых страниц.
- распознавание текста загруженных документов и поиск значений в нем.
- функция предварительного просмотра загружаемых pdf файлов.

Данные функции в проекте обеспечивают следующие инструменты.

- Adobe AxAcroPDFLib обеспечивает возможность предварительного просмотра документа, а также содержит набор методов, обеспечивающих доступ к элементам управления браузером pdf. Это создаваемый интерфейс. Этот объект позволяет загружать файл, перемещаться на разные страницы в файле и задавать различные параметры отображения и печати.

- PDF FocusLib обеспечивает постраничную конвертацию pdf документа в изображение с расширением .jpg для возможности дальнейшего распознавания текста.

Особое внимание уделено выбору инструмента по распознаванию текста в загружаемых документах. Оптическое распознавание символов (англ. Optical

Character Recognition – OCR) – это технология, которая позволяет преобразовывать различные типы документов, такие как отсканированные документы, pdf файлы или фото с цифровой камеры, в редактируемые форматы с возможностью поиска.

Проведен анализ трех наиболее популярных OCR библиотек с целью определить наиболее подходящий вариант для решения поставленных задач.

Наиболее популярные OCR библиотеки:

- Google Text Recognition API;
- Tesseract;
- Anyline.

1.5.1 Google Text Recognition API

Google Text Recognition API –это процесс обнаружения и распознавания текста в изображениях и видео. После обнаружения распознаватель определяет фактический текст в каждом блоке и разбивает его на слова и строки. Он обнаруживает текст различных языков (французский, немецкий, английский и т.д.) в режиме реального времени.

Данная OCR дает возможность распознавать текст как в реальном времени, так и с уже готовых изображений текстовых документов. В ходе анализа данной библиотеки выявлены как преимущества, так и недостатки ее использования.

Преимущества:

- возможность распознавания текста в реальном времени;
- возможность распознавания текста с изображений;
- небольшой размер библиотеки;
- высокая скорость распознавания.

Недостатки:

- большой размер файлов с обученными данными (~30Mb).

1.5.2 Tesseract

Tesseract – это OCR библиотека с открытым исходным кодом для разных операционных систем. Представляет собой бесплатное программное обеспечение, выпущенное под лицензией Apache³, версия 2.0, поддерживает различные языки.

Разработка библиотеки финансировалась компанией Google с 2006 года, время, когда она считалась одной из наиболее точных и эффективных OCR библиотек с открытым исходным кодом.

Преимущества:

- имеет открытый исходный код;
- имеет доступную документацию;
- соответственно, достаточно легко обучить OCR распознавать нужные шрифты и повысить качество распознаваемой информации.

Недостатки:

- недостаточная быстрота распознавания, что устраняется путем тренировки и обучения алгоритма распознавания.

1.5.3 Anyline

Anyline предоставляет многоплатформенный SDK, который позволяет разработчикам легко интегрировать функции OCR в приложения. Данная OCR библиотека привлекает многочисленными возможностями настройки параметров распознавания и предоставляемыми моделями для решения конкретных прикладных задач. Библиотека платная, и предназначена для коммерческого использования.

Преимущества:

- довольно простая настройка распознавания нужных шрифтов;
- распознавание текста в реальном времени;
- легкая и удобная настройка параметров распознавания;
- библиотека может распознавать штрихкоды и QR-коды;

³ (англ. Apache License) — лицензия на свободное программное обеспечение Apache Software Foundation.

- предоставляет готовые модули для решения разных задач.

Недостатки:

- невысокая скорость распознавания;
- для получения удовлетворительных результатов требуется первоначальная настройка шрифтов для распознавания;
- библиотека является платной .

Для решения поставленных задач выбрана библиотека Tesseract.

Разработанное решение позволяет распознавать текст в загруженных документах, записывать результат распознавания в текстовый файл, и далее в единую базу данных. Качество распознаваемой информации составляет около 90, что является очень хорошим результатом.

1.6 Построение модели базы данных проекта

Проектирование БД – одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы. В результате решения этой задачи должны быть определены содержание БД, эффективный для всех её будущих пользователей способ организации данных и инструментальные средства управления данными.

В крупных системах проектирование БД требует особой тщательности, поскольку цена допущенных на этой стадии просчётов и ошибок особенно велика. Некоторые ошибки проектирования можно скорректировать позже в процессе эксплуатации с помощью средств реструктуризации и реорганизации БД, но такие операции являются весьма трудоемкими и дорогостоящими

Основная цель процесса проектирования базы данных состоит в получении такого проекта, который удовлетворяет следующим требованиям.

- Корректность схемы БД, т.е. база должна быть гомоморфным образом моделируемой ПО, где каждому объекту ПО соответствуют данные в памяти ЭВМ, а каждому процессу – адекватные процедуры обработки данных.

- Обеспечение ограничений на объёмы внешней и оперативной памяти и другие ресурсы вычислительной системы.
- Эффективность функционирования (соблюдение ограничений на время реакции системы на запрос и обновление данных).
- Защита данных от сбоев и несанкционированного доступа.
- Простота и удобство эксплуатации.
- Гибкость, т.е. возможность развития и адаптации к изменениям ПО и/или требований пользователей.

Удовлетворение первых 4-х требований обязательно для принятия проекта базы данных.

Процесс проектирования БД включает в себя следующие этапы.

- Информационно-логическое (инфологическое) проектирование.
- Определение требований к операционной обстановке, в которой будет функционировать информационная система.
- Логическое проектирование БД.
- Физическое проектирование БД.

Ниже перечислены некоторые объекты инфологической модели данных

- **Группа пользователей** – содержит информацию о группах пользователей, участвующих в процессе документооборота и формирования отчетности.
- **Пользователь** – содержит информацию о пользователях и их правах на загрузку, возврат или утверждение документов.
- **Правило** – содержит информацию о критериях, правилах по которым распознается текст в документах.
- **Сделка trade-in** – содержит основные сведения о сделках по обмену старого автомобиля на новый.
- **Журнал сделок** – содержит информацию о всех загруженных сделках (чистая продажа и trade-in) и их критериях

– **Архив сделок** – содержит информацию об утвержденных сделках предыдущих периодов.

Каждый из этих объектов имеет свои свойства и связи с другими объектами. Схема разработанной базы данных представлена на рисунке 2.2.

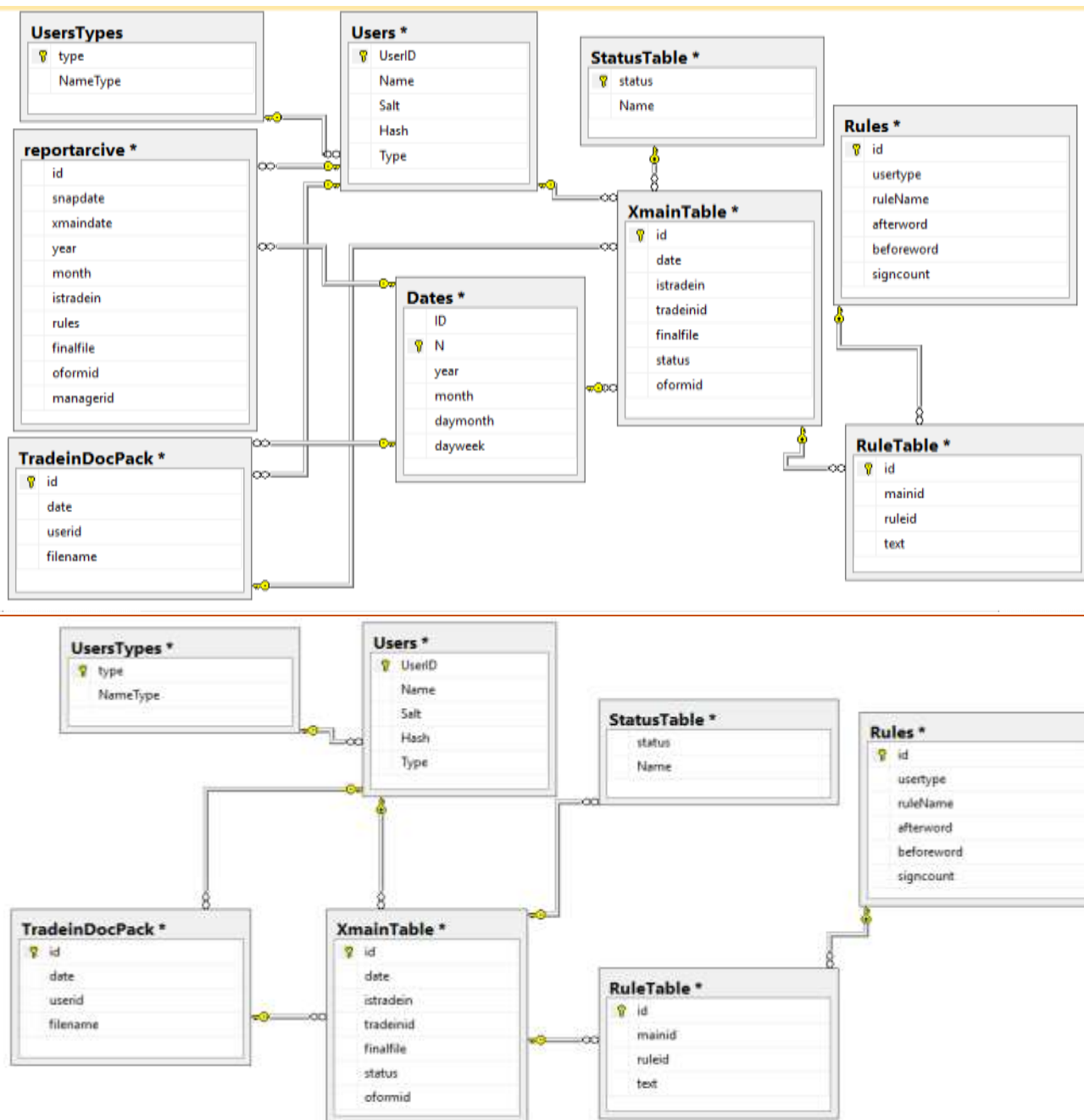


Рисунок 2.2 – Схема базы данных

1.7–2.2

1.7 Описание таблиц базы данных

Таблица Users предназначена для хранения информации о пользователях системы. В таблице 1 приведено описание ее полей. Для обеспечения мер безопасности, в таблице хранятся соль и хэш паролей.

Таблица 1 – Поля таблицы Users

Название поля	Описание	Тип поля
UserId	Уникальный код	Идентификатор
Name	Имя сотрудника	Текстовое
Salt	Соль пароля	Целое Текстовое
Hash	Хэш пароля	Текстовое
Type	Тип пользователя	Целое

Таблица UsersTypes предназначена для хранения типов пользователей системы. Для каждого типа пользователя предоставляется отдельный уникальный интерфейс. В таблице 2 приведено описание ее полей.

Таблица 2 – Поля таблицы UsersTypes

Название поля	Описание	Тип поля
Type	Уникальный код	Идентификатор
NameType	Наименование типа	Текстовое

Таблица Rules предназначена для хранения информации о правилах распознавания текста в загружаемых документах. В таблице 3 приведено описание ее полей.

Таблица 3 – Поля таблицы Rules

Название поля	Описание	Тип поля
id	Уникальный код	Идентификатор
userType	Тип пользователя	Текстовое
ruleName	Наименование правила	Текстовое
Aafterword	Искать перед словом	Текстовое
beforeword	Искать перед словом	Текстовое
signcount	Количество символов	Текстовое

Таблица RuleTable предназначена для хранения информации о распознанных критериях документа в соответствии с заданными правилами по каждой сделке. В таблице 4 приведено описание ее полей.

Таблица 4 – Поля таблицы RulesRuleTable

Название поля	Описание	Тип поля
id	Уникальный код	Идентификатор
<u>userType</u> <u>mainid</u>	<u>Тип пользователя</u> <u>Номер сделки</u>	<u>Текстовое</u> <u>Целое</u>
<u>ruleName</u> <u>ruleid</u>	<u>Наименование</u> — <u>правила</u> <u>Номер</u> <u>правила</u>	<u>Целое</u> <u>Текстовое</u>
<u>Aafterword</u> <u>text</u>	<u>Искать</u> — <u>перед</u> — <u>словом</u> <u>Распознанное</u> <u>правило</u>	Текстовое

Таблица TradeinDockPack предназначена для хранения информации о загруженных документах по сделкам трейд-ин. В таблице 5 приведено описание ее полей.

Таблица 5 – Поля таблицы TradeinDockPack

<u>Название поля</u>	<u>Описание</u>	<u>Тип поля</u>
<u>id</u>	<u>Уникальный код</u>	<u>Идентификатор</u>
<u>date</u>	<u>Дата загрузки</u>	<u>Дата</u>
<u>userid</u>	<u>Код пользователя</u>	<u>Целое</u>
<u>filename</u>	<u>Загруженный файл(путь)</u>	<u>Текстовое</u>

Таблица StatusTable предназначена для хранения возможных статусов сделки, которые зависят от стадии формирования конечного файла.

1 – Загружен Трейд-ин.

2 – Чистая продажа.

3 – Утверждено.

4 – Возврат.

5 – Закрыт.

6 – Первичный возврат.

В таблице 6 приведено описание ее полей.

Таблица 6 – Поля таблицы StatusTable

<u>Название поля</u>	<u>Описание</u>	<u>Тип поля</u>
<u>status</u>	<u>Уникальный код</u>	<u>Идентификатор</u>
<u>name</u>	<u>Название статуса</u>	<u>Текстовое</u>

Таблица XmainTable предназначена для хранения информации о загруженных документах по сделкам и статусах сделки. В таблице 7 приведено описание ее полей.

Таблица 7 – Поля таблицы XmainTable

<u>Название поля</u>	<u>Описание</u>	<u>Тип поля</u>
<u>id</u>	<u>Уникальный код</u>	<u>Идентификатор</u>
<u>date</u>	<u>Дата сделки</u>	<u>Дата</u>
<u>isTradein</u>	<u>Признак обмена</u>	<u>Целый</u>
<u>tradeinId</u>	<u>Код сделки по обмену</u>	<u>Целый</u>
<u>finalFile</u>	<u>Конечный файл сделки(путь)</u>	<u>Текстовый</u>

<u>status</u>	<u>Статус сделки</u>	<u>Целое</u>
<u>oformId</u>	<u>Код сотрудника</u>	<u>Целое</u>

Таблица Dates сформирована в качестве календаря для формирования архива документов.

Таблица 8 – Поля таблицы Dates

<u>Название поля</u>	<u>Описание</u>	<u>Тип поля</u>
<u>id</u>	<u>Уникальный код</u>	<u>Идентификатор</u>
<u>date</u>	<u>Дата</u>	<u>Дата</u>
<u>year</u>	<u>Год</u>	<u>Целый</u>
<u>month</u>	<u>Месяц</u>	<u>Целый</u>
<u>daymonth</u>	<u>День месяца</u>	<u>Целый</u>
<u>dayweek</u>	<u>День недели</u>	<u>Целое</u>
<u>monthname</u>	<u>Наименование месяца</u>	<u>Текстовый</u>

В листинге 1 приведен текст запроса для формирования таблицы Dates

Листинг 1 – T-SQL запрос для формирования таблицы Dates

```

DECLARE @Cardinality datetime, @Seed datetime, @Increment INT;
SELECT
    @Cardinality = dbo.OnlyDate(getdate()),
    @Seed = dbo.OnlyDate(getdate()-600),
    @Increment = 1;
WITH T(ID,N) AS
(
    SELECT 1, @Seed WHERE @Cardinality >= getdate()-1
    UNION ALL
    SELECT ID + 1, N + @Increment FROM T WHERE ID < @Cardinality
)
SELECT
    ID,
    N,
    year(N) as year,
    month(N) as month,
    case when month(N) = 1 then 'Январь'
        when month(N) = 2 then 'Февраль'
        when month(N) = 3 then 'Март'
        when month(N) = 4 then 'Апрель'
        when month(N) = 5 then 'Май'

```

```

when month(N) = 6 then 'Июнь'
when month(N) = 7 then 'Июль'
when month(N) = 8 then 'Август'
when month(N) = 9 then 'Сентябрь'
when month(N) = 10 then 'Октябрь'
when month(N) = 11 then 'Ноябрь'
when month(N) = 12 then 'Декабрь'
end as monthname,
DAY(N) as daymonth,
DATEPART(dw,N) as dayweek
into Dates
FROM T ORDER BY ID
OPTION (MAXRECURSION 0);

```

Таблица ReportArchive сформирована для хранения архива утвержденных документов, предоставляемых в отчете для ООО Тойота Мотор.

Таблица 9 – Поля таблицы ReportArchive

<u>Название поля</u>	<u>Описание</u>	<u>Тип поля</u>
<u>id</u>	<u>Уникальный код</u>	<u>Идентификатор</u>
<u>snapdate</u>	<u>Дата записи</u>	<u>Дата</u>
<u>xmaindate</u>	<u>Дата сделки</u>	<u>Дата</u>
<u>year</u>	<u>Год</u>	<u>Целое</u>
<u>month</u>	<u>Месяц</u>	<u>Целое</u>
<u>istradein</u>	<u>Признак трейд-ин</u>	<u>Целое</u>
<u>rules</u>	<u>Содержание документа</u>	<u>Текстовый</u>
<u>finalfile</u>	<u>Путь к архивному файлу</u>	<u>Текстовый</u>
<u>oformid</u>	<u>Код оформителя</u>	<u>Целое</u>
<u>managerid</u>	<u>Код менеджера трейд-ин</u>	<u>Целое</u>

1.8 Разработка системы формирования отчетности

Проект содержит в себе 7 файлов форм, файлы ресурсов проекта в папке Resources, 2 файла исходного кода с расширением .cs, один из которых представляет собой класс, реализующий надежное хранение паролей.

Файлы, которые содержит в себе проект «MyTMR», представлены на рисунке 2.3.

Файл app.config содержит в себе настройки конфигурации системы, например, строки подключения баз данных, используемых в системе.

Файл TMRDataBDataSet.xsd – файл, содержащий в себе набор данных из подключенной базы.

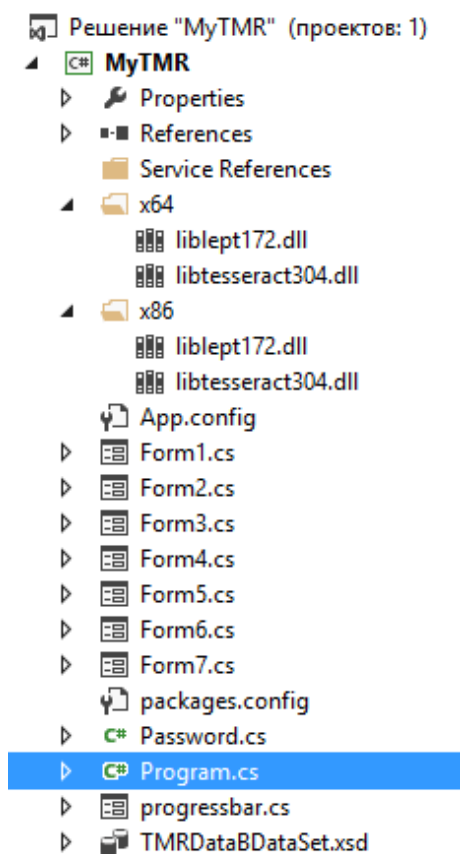


Рисунок 2.3 – Файлы разработанного проекта.

При запуске приложения открывается первая форма идентификации пользователя Form3.

The image shows a login form with a light gray background. At the top, the word 'Пользователь' is centered. Below it is a dropdown menu containing the text 'Широкова Ольга'. Underneath the dropdown is a password field with the label 'Пароль' and three black dots. At the bottom of the form is a gray button with the text 'Вход'.

Рисунок 2.4 – Форма авторизации пользователей.

Пользователь вводит пароль. При нажатии на кнопку «Вход» запускается процесс проверки его правильности. При успешном прохождении проверки определяется тип пользователя и в зависимости от определенного типа запускается следующая форма (Form1, Form4, Form7 или Form5). Каждая форма позволяет использовать инструменты, определяемые правами каждого типа пользователей.

Предопределены четыре типа пользователей приложения, для каждого свои функциональные разрешения.

Менеджер. Производит загрузку документов по принятому автомобилю с пробегом в систему и обрабатывает возвращенные, не принятые документы.

Оформитель. Производит прием документов по автомобилю с пробегом, и прикрепляет к ним дополнительный комплект по проданному новому автомобилю. При продаже автомобиля без обмена старого автомобиля на новый, оформитель самостоятельно загружает документы по сделке. Обрабатывает возвращенные, не принятые документы.

Логист. Отслеживает статистику за текущий месяц по текущим, утвержденным, возвращенным сделкам. Осуществляет конечную проверку загруженных документов, утверждает документы для дальнейшего предоставления в ООО Тойота Мотор. Имеет доступ к архиву ранее утвержденных сделок.

Администратор. Осуществляет управление пользователями. Управление правилами распознавания текста в документах

1.8.1 Интерфейс «Менеджер»

На рисунках 2.5 и 2.6 представлен интерфейс менеджера по приему автомобиля с пробегом (Form1).

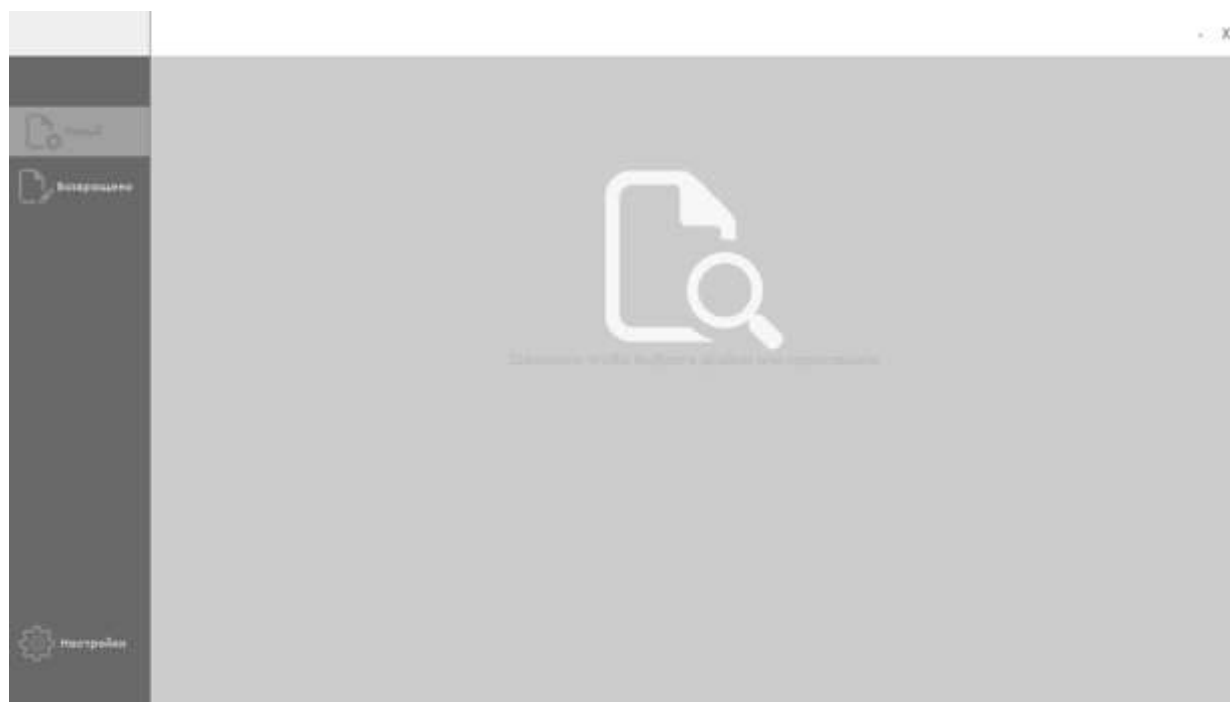


Рисунок 2.5 – Интерфейс пользователя «Менеджер»

На форме предусмотрена загрузка одного или нескольких файлов в формате .pdf путем перетаскивания на форму или выбором пути к файлу по нажатию на центральную кнопку (механизм openFileDialog). Осуществляется проверка формата выбранного файла с сообщением об ошибке.

На форме обеспечен функционал предварительного просмотра загруженного документа для исключения ошибок.

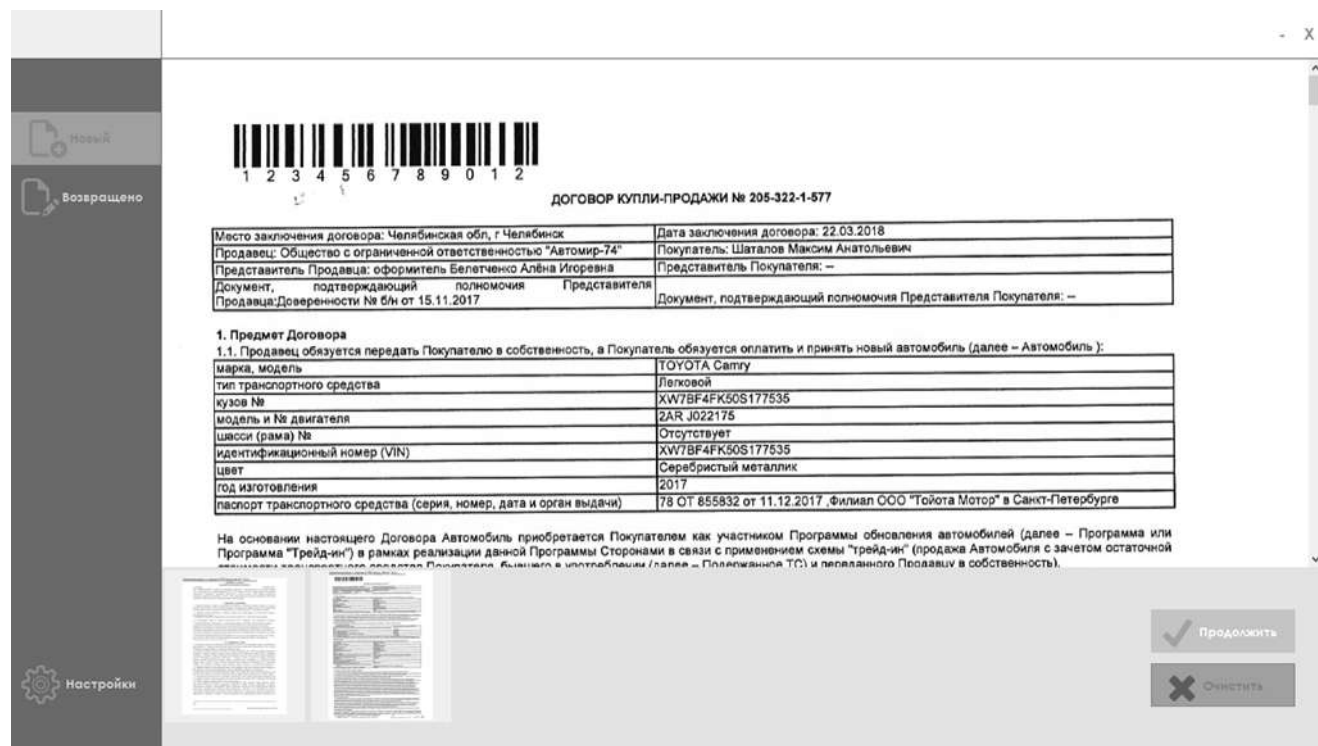


Рисунок 2.6 – Предварительный просмотр документа.

Кнопка «Очистка» предназначена для очистки загруженных по ошибке файлов. Нажатие на кнопку «Продолжить» запускает процесс получения необходимых критериев сделки. Сначала загруженные .pdf – файлы разделяются на отдельные файлы с расширением .jpg, затем на сервере запускается хранящая процедура для получения предустановленных правил распознавания текста в документе и формирование распознанных критериев. Вызывается динамическая форма [Form2](#), приведенная на рисунке 2.7.

VIN

XW7RDRF12P1231

Договор

№ 255-00-01

Отмена

Добавить

Рисунок 2.7 – Форма формирования критериев сделки.

Код распознавания текста приведен в листинге 1

Листинг1 – Код распознавания текста

```
// код главной формы (Form1)
private void button6 Click(object sender, EventArgs e)
// нажатие на кнопку «Продолжить»
{
    if (flowLayoutPanel1.Controls.Count == 0)
    {
        MessageBox.Show("Не выбран ни один файл");
    }
    else
    {
        Form2 f2 = new Form2(fileListjpg, 2); //список файлов
        f2.FormClosing += new FormClosingEventHandler(f2 FormClosing);
        f2.ShowDialog();
    }
}

// код вызываемой формы (Form2)
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using Tesseract;
using System.Text.RegularExpressions;

namespace MyTMR
{
    public partial class Form2 : Form
    {
        public List<TradeinList> tradeinlist = new List<TradeinList>();
        private List<string> filelist;
        private int type;
        public Form2(List<string> filelist, int type)
        {
            InitializeComponent();
            filelist = filelist;
            type = type;
        }
        string connectionString = @"Data Source=my-pc\sqlexpress;Initial Cata-
log=TMRDataB;Integrated Security=True"; // строка подключения к бд
        int flag = 0;
        class MyTable //Правила распознавания
        {
            public string id { get; set; }
        }
    }
}
```

```

        public string name { get; set; }
        public string after { get; set; }
        public string before { get; set; }
        public string length { get; set; }
    }
    public class TradeinList //Распознанные правила
    {
        public string id { get; set; }
        public string rule { get; set; }
    }
    public int GetFlag()
    {
        return flag;
    }
    public string[,] GetTradeinList()
    {
        string[,] s = new string[tradeinlist.Count,2];
        int i = 0;
        foreach (TradeinList t in tradeinlist)
        {
            s[i,0] = t.id;
            s[i,1] = t.rule;
            i++;
        }
        return s;
    }
    private void Form2_Load(object sender, EventArgs e)
    //распознавание текста
    {
        foreach (string jpg in filelist)
        // распознавание на русском языке
        {
            var img = new Bitmap(jpg);
            var ocr = new TesseractEngine("./tesdata", "rus",
            EngineMode.TesseractOnly);
            var page = ocr.Process(img);
            System.IO.File.AppendAllText(@"D:\files\temp.txt",
            page.GetText());
        }
        foreach (string jpg in filelist)
        //распознавание на английском языке
        {
            var img = new Bitmap(jpg);
            var ocr = new TesseractEngine("./tesdata", "eng",
            EngineMode.TesseractOnly);
            var page = ocr.Process(img);
            System.IO.File.AppendAllText(@"D:\files\tempE.txt",
            page.GetText());
        }
    }
    private void Form2_Shown(object sender, EventArgs e) //поиск по правилам
    {

```



```

List<MyTable> dat = new List<MyTable>();
string regular;
string input;
using (SqlConnection con = new SqlConnection(connectionString))
{
    using (SqlCommand command = new SqlCommand("SelectRulesOnType",
con))
    {
        command.CommandType = System.Data.CommandType.StoredProcedure;
        SqlParameter param = new SqlParameter();
        param.ParameterName = "@type";
        param.SqlDbType = SqlDbType.Int;
        param.Value = type;
        command.Parameters.Add(param);
        DataTable dt = new DataTable();
        con.Open();
        using (SqlDataReader reader = command.ExecuteReader())
        {
            while (reader.Read()) //получение списка правил
            {
                dat.Add(new MyTable()
                {
                    name = (reader["RuleName"].ToString()),
                    after = (reader["afterword"].ToString()),
                    before = (reader["beforeword"].ToString()),
                    length = (reader["signcount"].ToString()),
                    id = (reader["id"].ToString()),
                });
            }

            foreach (var d in dat) //формирование регулярных выражений
            {
                if (d.before.ToString() != "")
                {
                    regular = @"(?<=" + d.after + ")(.*)" +
d.before + ")";
                }
                else
                {
                    string str = new string('.', Con-
vert.ToInt32(d.length));
                    regular = @"((?<=" + d.after + ")("+str+"))";
                }
                Regex regex = new Regex(regular);
                input = Sys-
tem.IO.File.ReadAllText(@"D:\files\temp.txt");
                var match = regex.Match(input);
                if (match.Success)
                {
                    Label lab = new Label();

```

```

lab.Font = new Font("Century Gothic", 11,
FontStyle.Bold);
lab.ForeColor = Color.DarkGray;
lab.Margin = new Padding(5, 10, 5, 5);
lab.Text = d.name.ToString();
lab.Name = d.name.ToString();
TextBox txtb = new TextBox();
txtb.Margin = new Padding(5, 5, 5, 10);
txtb.BorderStyle = BorderStyle.None;
txtb.Font = new Font("Century Gothic", 11,
FontStyle.Regular);
txtb.ForeColor = SystemColors.ActiveCaption;
txtb.Width = 270;
txtb.Text = match.ToString();
txtb.Name = d.id.ToString();
this.flowLayoutPanel1.Controls.Add(lab);
this.flowLayoutPanel1.Controls.Add(txtb);
}
else
{
input = Sys-
tem.IO.File.ReadAllText(@"D:\files\tempE.txt");
match = regex.Match(input);
if (match.Success)
{
Label lab = new Label();
lab.Font = new Font("Century Gothic", 11,
FontStyle.Bold);
lab.ForeColor = Color.DarkGray;
lab.Margin = new Padding(5, 10, 5, 5);
lab.Text = d.name.ToString();
TextBox txtb = new TextBox();
txtb.Margin = new Padding(5, 5, 5, 10);
txtb.BorderStyle = BorderStyle.None;
txtb.Font = new Font("Century Gothic", 11,
FontStyle.Regular);
txtb.ForeColor = SystemColors.ActiveCaption;
txtb.Width = 270;
txtb.Text = match.ToString();
txtb.Name = d.id.ToString();
this.flowLayoutPanel1.Controls.Add(lab);
this.flowLayoutPanel1.Controls.Add(txtb);
}
else
{
Label lab = new Label();
lab.Font = new Font("Century Gothic", 11,
FontStyle.Bold);
lab.ForeColor = Color.DarkGray;
lab.Margin = new Padding(5, 10, 5, 5);
lab.Text = d.name.ToString();
TextBox txtb = new TextBox();
txtb.Margin = new Padding(5, 5, 5, 10);
txtb.BorderStyle = BorderStyle.None;

```

```

txtb.Font = new Font("Century Gothic", 11,
FontStyle.Regular);
txtb.ForeColor = SystemColors.ActiveCaption;
txtb.Name = d.id.ToString();
txtb.Width = 270;
this.flowLayoutPanel1.Controls.Add(lab);
this.flowLayoutPanel1.Controls.Add(txtb);
}
}
}
}
}
}
}
private void button3 Click(object sender, EventArgs e) //закрытие формы
{
foreach (TextBox txtb in
this.flowLayoutPanel1.Controls.OfType<TextBox>()) //считывание распознанных данных
{
tradeinlist.Add (new TradeinList ()
{
id = txtb.Name.ToString(),
rule = txtb.Text
});
}
flag = 1;
this.Close();
}
void f2 FormClosing(object sender, FormClosingEventArgs e)
{
const string dest = @"D:\files\";
int i = (sender as Form2).GetFlag();

switch (i)
{
case 1:
string tradeinpdfname = "";
string [,] tradein = (sender as Form2).GetTradeinList();
for (int k = 0; k < tradein.Rank;k++)
{
tradeinpdfname = tradeinpdfname + " " + tradein[k, 1].ToString();
}
try
{
PdfDocument outputDocument = new PdfDocument();
foreach (string f in fileListpdf)
{

```

```

// Импорт страниц из pdf
PdfDocument inputDocument = PdfReader.Open(f,
PdfDocumentOpenMode.Import);
//Перебор страниц.
int count = inputDocument.PageCount;
for (int idx = 0; idx < count; idx++)
{
PdfPage page = inputDocument.Pages[idx];

outputDocument.AddPage(page);
}
}
// Сохранение документа
outputDocument.Save( dest + tradeinpdfname + ".pdf");
this.queriesTableAdapter.TradeInIncert(userid,dest +
tradeinpdfname + ".pdf" );
for (int k = 0; k < tradein.Rank;k++)
{
this.queriesTableAdapter.XmainFiller(null,Convert.ToInt32(tradein[k, 0]),
tradein[k, 1]);
}
MessageBox.Show("Успешное добавление");
button4.Visible = true;
button4.BringToFront();
flowLayoutPanel1.Controls.Clear();
listClear();
}
catch
{
MessageBox.Show("Произошла непредвиденная ошиб-
ка.Попробуйте еще раз.");
}
break;
}
}
}

```

Форма позволяет откорректировать ошибки, возникшие в процессе распознавания, при которых текст с документа был распознан некорректно или вовсе не был распознан.

Форма является универсальной и вызывается из любого интерфейса с определенными входными параметрами (тип пользователя и список файлов для обработки).

Нажатие кнопки «Добавить» на форме запускает процесс формирования файла-слияния документов, запись его в файловую директорию, изменение стату-

са сделки и запись пути к файлу в базу данных. Данные действия реализуются с помощью хранимых процедур на сервере.

Хранимая процедура TradeInIncert предназначена для записи данных о сделке по трейд-ин. Код процедуры представлен в листинге 4.

Листинг 4 – Код хранимой процедуры TradeInIncert

```
CREATE PROCEDURE [dbo].[TradeInIncert]
    @userid int,
    @filename nvarchar(max)
AS
    SET NOCOUNT ON;
    declare @ret int

    insert into TradeinDocPack values (getdate(),@userid,@filename)
    declare @id int

    set @id = SCOPE IDENTITY ()
    insert into XmainTable values (GETDATE(),1,@id,null,1,null)
    select Max(id) from XmainTable
```

Хранимая процедура XmainFiller предназначена для записи данных о распознанных данных по сделке в соответствии с правилами. Код процедуры представлен в листинге 5

Листинг 5 – Код хранимой процедуры XmainFiller

```
CREATE PROCEDURE [dbo].[XmainFiller]
    @id int,
    @ruleid int,
    @text nvarchar(max)
AS
BEGIN
    SET NOCOUNT ON;
    declare @xmainid int
    if exists (select * from XmainTable where id = @id)
        begin
            set @xmainid = (@id)
        end
    else
        begin
            -- Insert statements for procedure here
            set @xmainid = (select max(id) from XmainTable)
        end
    insert into RuleTable values (@xmainid,@ruleid,@text)
END
```

1.7.11.8.2 Интерфейс «Оформитель»

Интерфейс позволяет осуществлять функционал по присоединению дополнительных документов к сделке. Интерфейс «Оформитель» представлен на рисунках 2.8 и 2.9.

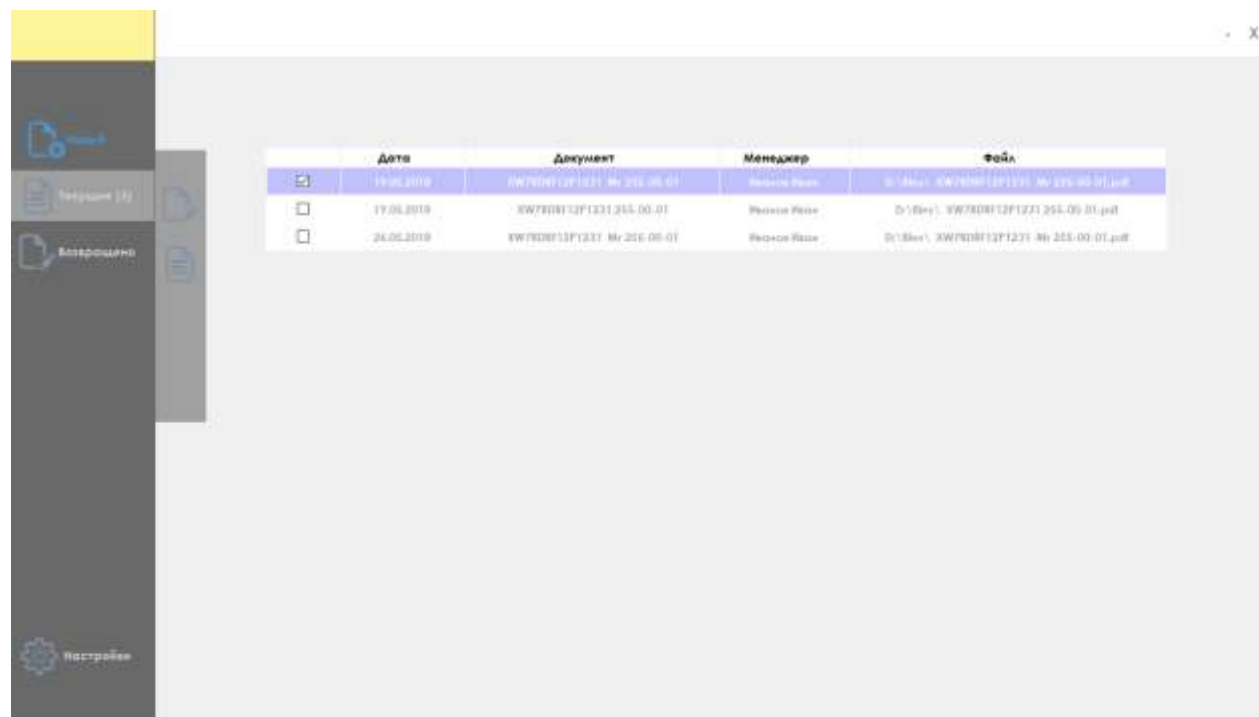


Рисунок 2.8 – Интерфейс «Оформитель».

На вкладке «Текущие» выводится список документов, загруженных при приеме автомобиля с пробегом. Включена функция предварительного просмотра документа для его визуальной проверки. При формировании файла слияния статус сделки меняется на 2 и в базу данных записывается путь к конечному, основному файлу сделки.

Интерфейсы загрузки первичного файла по продаже нового автомобиля, присоединения дополнительного файла и их механизмы идентичны описанным выше.



Рисунок 2.9 – Предварительный просмотр трейд-ин документа.

Добавлена функция возврата файла на доработку, которая реализуется с помощью хранимой процедуры `Back_Status`, код которой приведен в листинге 6.

Листинг 6 – Код хранимой процедуры `Back_Status`

```
CREATE PROCEDURE [dbo].[Back_Status]
____ @mainid int,
____ @status int
AS
BEGIN
____ SET NOCOUNT ON;
____ update xmaintable set status = @status where id = @mainid
____ END
```

1.7.21.8.3 Интерфейс «Логист»

Логист является ответственным за правильность данных в предоставленной отчетности. Функционал пользователя расширен по сравнению с другими пользователями. На рисунках 2.10, 2.11 и 2.12 представлен интерфейс и основной функционал пользователя.

Статистика	Количество
Загружено Документов	0
Загружено Документов на доработку(возврат)	0
Утверждено	0
Загружено Документов для отчетности	0
Утверждено для отчетности	0

Рисунок 2.10 – Интерфейс «Логист», вкладка статистика

При запуске формы формируется статистика по текущему месяцу, где логист может отслеживать количество загруженных документов по каждой сделке, количество документов, попавших на доработку(возврат) и документы, утвержденные для отчетности.

На вкладке «Текущие» происходит управление сделками. Предусмотрен просмотр документов, отправка на доработку, и утверждение к отчету.

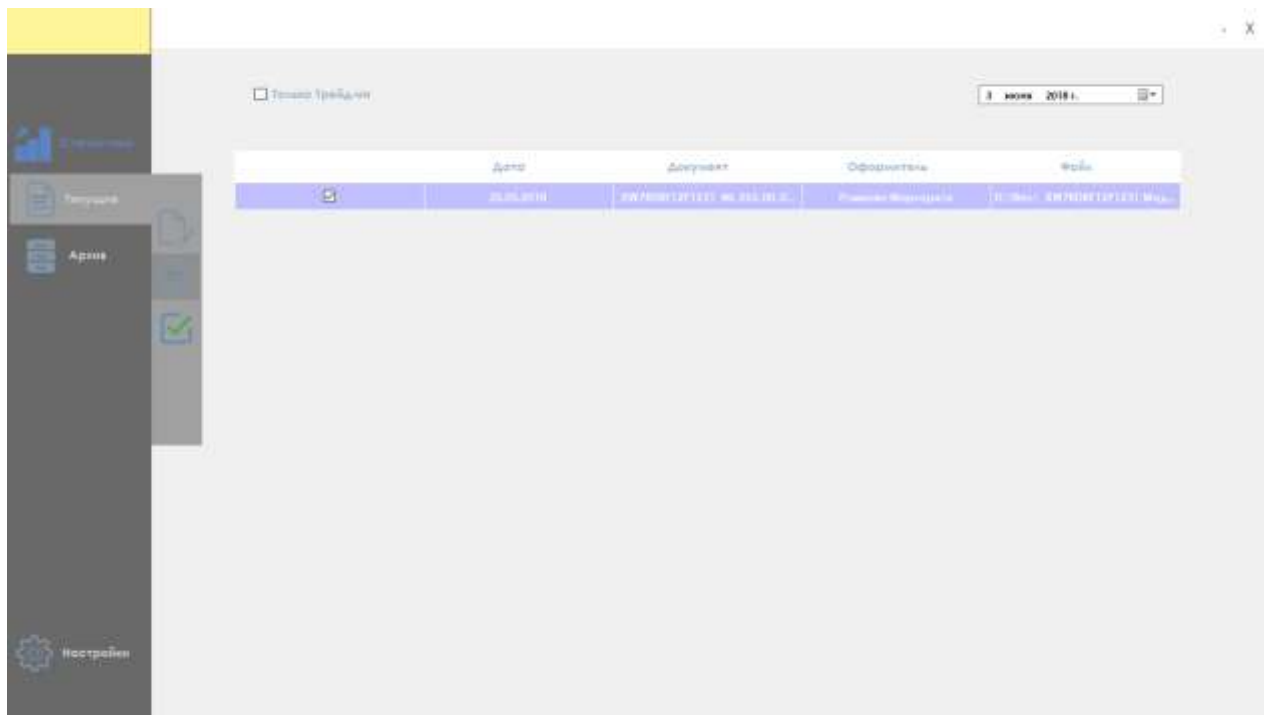


Рисунок 2.11 – Интерфейс «Логист», текущие документы

Пользователь с типом «Логист» имеет доступ к архиву ранее загруженных документов.

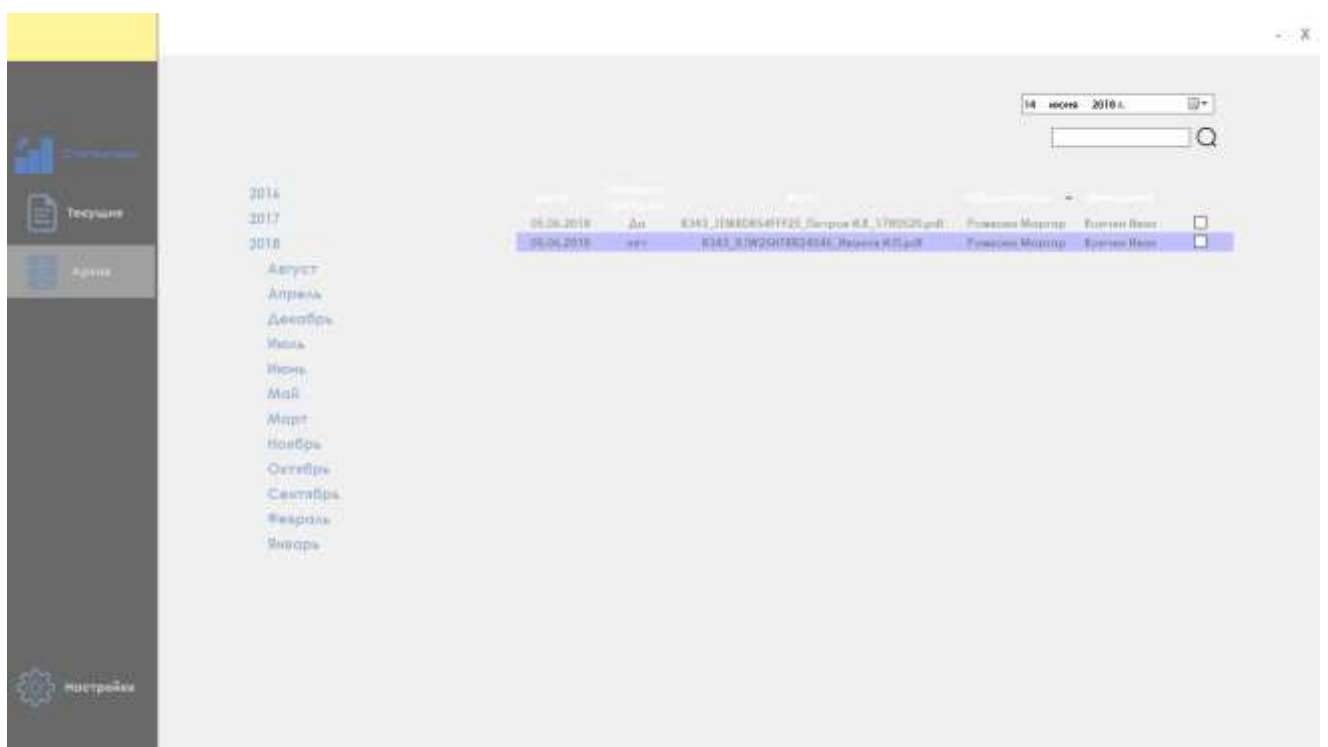


Рисунок 2.12 – Интерфейс «Логист», архив сформированной отчетности

Архив документов (таблица ReportArcive) формируется с помощью задания GoArcive, которое запускается по расписанию каждое 5 число месяца. Таблица заполняется данными за предыдущий месяц, являющийся отчетным. Таблицы Xmaintable, Ruletable TradeinDockPack очищаются. Листинг задания приведен в листинге 7.

Листинг 7 – Задание SQL-Агента для формирования таблицы ReportArcive

```
set nocount on  
  
declare @month int  
set @month = month(getdate())-1  
  
select  
*  
into #xmaintable  
from XmainTable  
where status = 3 and date  
  
select  
*  
into #tradein  
from TradeinDocPack  
where id in (select tradeinid from #xmaintable)  
  
select  
*  
into #rules  
from RuleTable  
where mainid in (select id from #xmaintable)  
  
insert into
```

По итогам выполнения задания и формирования отчетности на почту приходит уведомление о том, что файлы доступны. Пример письма представлен на рисунке 2.13



Рисунок 2.13 – Образец присылаемого письма о доступности отчета

1.8.4Интерфейс Администратора

На рисунке 2.14 приведена форма администратора. На форме администратора обеспечен функционал работы с пользователями (добавление, удаление, сброс пароля).

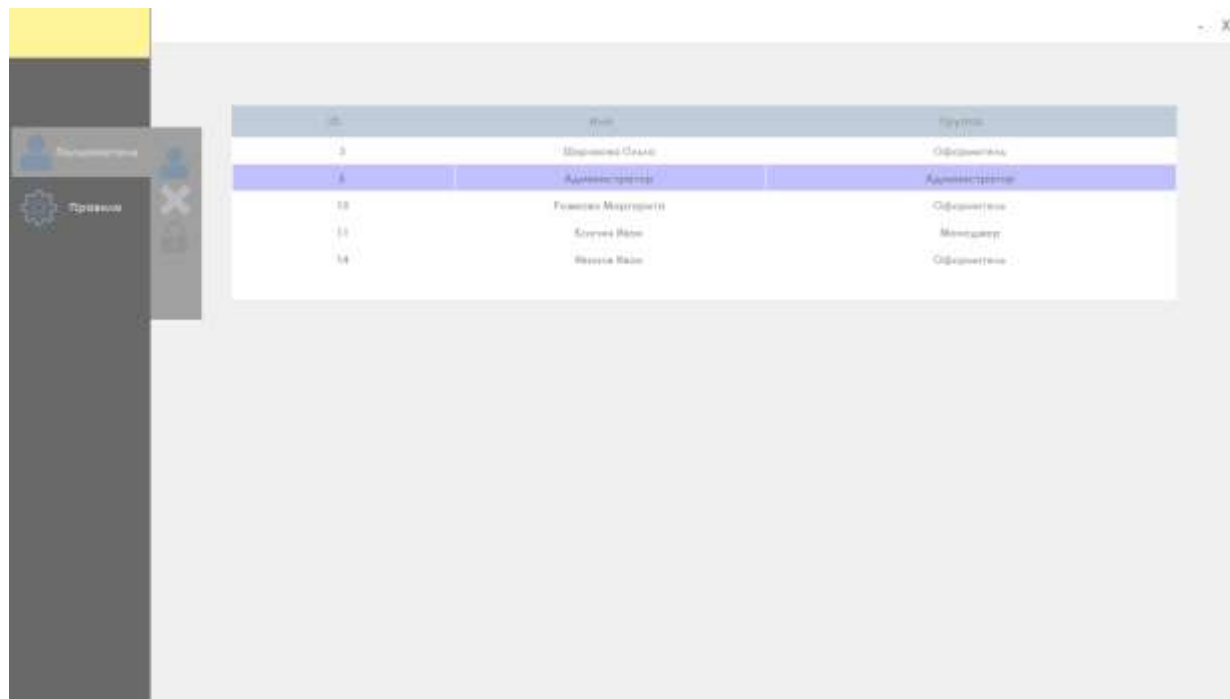


Рисунок 2.14 – Интерфейс управления пользователями системы

Инструменты на вкладке «Правила» позволяют задавать правила именования загружаемых документов и правила, по которым происходит распознавание текста в документах.

Настроена функция задания параметров распознавания текста, приведен пример на рисунке 2.15.



Рисунок 2.15 – Интерфейс задания правил распознавания

Форма задания правил распознавания представлена на рисунках 2.16 и 2.17. При неизвестном количестве символов можно задать слово, до которого будет распознан текст.

Группа пользователей:

Наименование правила:

Искать после:

Искать перед:

Количество символов:

Рисунок 2.16 – Способ задания правил при неизвестном количестве символов

При таком способе задания правил распознавания текста, будет найдена фраза, которая заключена между «Покупатель» и «заключили договор».

При известном количестве символов, которое необходимо распознать, нужно указать это количество и слово, после которого необходимо начать распознавание.

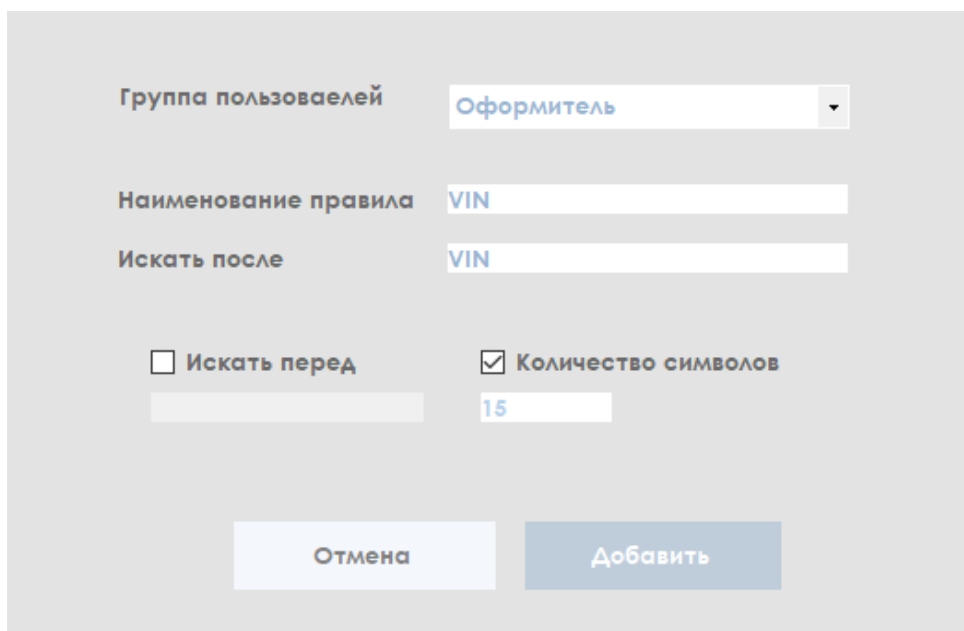


Рисунок 2.17 –Способ задания правил при известном количестве символов

При данном способе задания правил распознавания текста, будет найдена фраза, которая содержит 15 символов после фразы «VIN».

Код процедуры формирования списка правил представлен в листинге 8.

Листинг 8 – Код процедуры AddNewRule

```
CREATE PROCEDURE [dbo].[AddNewRule]  
_____ @type int,  
_____ @rulename varchar(50),  
_____ @aferword varchar(50),  
_____ @beforeword varchar(50),  
_____ @signcount varchar(4)  
  
AS  
BEGIN  
  
_____ SET NOCOUNT ON;  
  
_____ insert into rules values (@type,@rulename,@aferword, case when @beforeword = ''  
then null else @beforeword end,  
_____ case when @signcount = '' then null else @signcount end)
```

1.9 Правовые аспекты и меры безопасности

К защите конфиденциальных сведений предъявляют все большие требования. Поскольку персональные данные клиентов компании содержат данные личного характера, к обеспечению гарантий по их сохранности и неразглашению относятся очень серьезно. Персональные данные отнесены к категории конфиденциальных сведений (в соответствии с федеральным законом «О персональных данных» от 27.07.2006 N 152-ФЗ).

В целях настоящего Федерального закона относительно разработанного проекта используются следующие основные понятия.

– Обработка персональных данных – любое действие (операция) или совокупность действий (операций), совершаемых с использованием средств автоматизации или без использования таких средств с персональными данными, включая сбор, запись, систематизацию, накопление, хранение, уточнение (обновление, изменение), извлечение, использование, передачу (распространение, предоставление, доступ), обезличивание, блокирование, удаление, уничтожение персональных данных.

– Автоматизированная обработка персональных данных – обработка персональных данных с помощью средств вычислительной техники.

Обработка персональных данных должна осуществляться с соблюдением принципов и правил, предусмотренных настоящим Федеральным законом. Обработка персональных данных допускается с согласия субъекта персональных данных на обработку его персональных данных.

При формировании сделки по продаже нового автомобиля с каждым клиентом подписывается согласие на обработку персональных данных, в соответствии с которым организация имеет право хранить и использовать данные клиентов

для анализа, формирования статистики и отчетности без права передачи информации третьим лицам. Бланк согласия приведен в приложении А.

Разработанный проект содержит персональные данные клиентов компании (их паспортные данные, сведения об уплаченной сумме за автомобиль), следовательно, должны быть обеспечены меры по безопасному хранению загружаемых документов.

Все необходимые меры безопасности на данный момент приняты в организации ООО Автомир-74. Ограничен доступ к серверной, имеются средства шифрования сообщений электронной почты и данных на локальных жестких дисках, на персональных компьютерах установлены современные антивирусные программы и регулярно обновляемое лицензионное программное обеспечение.

В разработанном проекте для обеспечения безопасного хранения паролей учетных записей пользователей в базе данных использован алгоритм хеширования SHA-256 с дополнительным «подсаливанием» хеша.

SHA-256 (сокращение от Secure Hashing Algorithm) – популярный криптографический алгоритм хеширования, разработанный National Security Agency (Агентством национальной безопасности США). Задача SHA-256 состоит в том, чтобы сделать из случайного набора данных определённые значения с фиксированной длиной, которое послужит идентификатором этих данных.

Полученное значение сравнивается с дубликатами исходных данных, извлечь которые невозможно. Основная сфера применения алгоритма, это использование в различных приложениях или сервисах, связанных с защитой информации, где функция и получила широкое распространение. Этот алгоритм относится к группе шифровальных алгоритмов SHA-2, которые в свою очередь разработаны на базе алгоритма SHA-1, впервые созданного в 1995 году для использования в гражданских целях. Сам SHA-2 разработан Агентством национальной безопасности США весной 2002 года.

У такого метода есть два значительных достоинства:

– быстрая скорость шифрования и практически невозможная расшифровка без ключей;

– минимальный риск появления коллизий (одинаковых образов).

Хеш – строка пароля и сгенерированная «соль» хранятся в базе данных проекта в таблице Users.

В листинге 5 приведено описание класса Password

Листинг 5 – Описание класса Password

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Security.Cryptography;

namespace TMR
{
    class Password
    {
        private byte[] _salt;          // синхропосылка
        private byte[] _hash;         // хэш пароля
        /*все четыре свойства просто возвращают значения
        * соответствующих полей объекта
        */
        internal string Salt
        {
            get { return Convert.ToBase64String(_salt); }
        }

        internal string Hash
        {
            get { return Convert.ToBase64String(_hash); }
        }

        internal byte[] RawSalt
        {
            get { return (byte[])_salt.Clone(); }
        }

        internal byte[] RawHash
        {
            get { return (byte[])_hash.Clone(); }
        }
        internal Password(string salt, string hash)
        {
            /*Конструктор с двумя строковыми аргументами
            * просто преобразует синхропосылку и хэш из base64
            * в двоичное представление и сохраняет в
```



```

        * соответствующих полях класса
        * В конструктор передаем полученную соль и хэш пароля
        */
        _salt = Convert.FromBase64String(salt);
        _hash = Convert.FromBase64String(hash);
    }
    internal Password(byte[] salt, byte[] hash)
    {
        /*Конструктор, принимающий массивы байт,
        * создает копии входных массивов
        * В конструктор передаем полученную соль и хэш пароля
        */
        _salt = (byte[])salt.Clone();
        _hash = (byte[])hash.Clone();
    }
    internal Password(char[] clearText)
    {
        /*Генерация соли и пароля
        * Конструктор с одним аргументом генерирует
        * случайную синхрпосылку и вычисляет хэш
        * предоставленного пароля, используя внутренний
        * метод HashPassword()
        */
        _salt = GenerateRandom(6);
        _hash = HashPassword(clearText);
    }
    /*Метод Verify сначала вычисляет хэш предоставленного
    * пароля, используя все тот же внутренний метод
    * HashPassword, после чего сравнивает байты сохраненного
    * хэша с байтами только что вычисленного хэша. Совпадение
    * всех до одного байтов хэша означает, что пароль верный.
    */

    ///
<summary>
    ///
    /// </summary>

    /// <param name="clearText">Передаем введенный пароля для провер-
ки.</param>
    /// <returns>Возвращаем если все ок. или не ок</returns>
    internal bool Verify(char[] clearText)
    {
        byte[] hash = HashPassword(clearText);

        if (hash.Length == _hash.Length)
        {
            for (int i = 0; i < hash.Length; i++)
            {
                if (hash[i] != _hash[i])
                    return false;
            }
        }
    }

```

```

        return true;
    }

    return false;
}
/*Статический метод Generate просто генерирует
 * массив случайных байтов и преобразует его
 * в base64-строку.
 */
private static char[] Generate()
{
    char[] random = new char[12];

    // генерируем 9 случайных байтов; этого достаточно, чтобы
    // получить 12 случайных символов из набора base64
    byte[] rnd = GenerateRandom(9);

    // конвертируем случайные байты в base64
    Convert.ToBase64CharArray(rnd, 0, rnd.Length, random, 0);

    // очищаем рабочий массив
    Array.Clear(rnd, 0, rnd.Length);

    return random;
}
/* Метод записывает синхрпосылку и пароль в поток на
 * основе массива байтов, а затем вычисляет хэш
 * содержимого потока. В качестве хэш-функции используется
 * алгоритм SHA-256.
 */
private byte[] HashPassword(char[] clearText)
{
    Encoding utf8 = Encoding.UTF8;
    byte[] hash;

    // создаем рабочий массив достаточного размера, чтобы вместить
    byte[] data = new byte[_salt.Length
        + utf8.GetMaxByteCount(clearText.Length)];

    try
    {
        // копируем синхрпосылку в рабочий массив
        Array.Copy(_salt, 0, data, 0, _salt.Length);

        // копируем пароль в рабочий массив, преобразуя его в UTF-8
        int byteCount = utf8.GetBytes(clearText, 0, clearText.Length,
            data, _salt.Length);
        // хэшируем данные массива
        using (HashAlgorithm alg = new SHA256Managed())
            hash = alg.ComputeHash(data, 0, _salt.Length + byteCount);
    }
    finally
    {

```

```

        // очищаем рабочий массив в конце работы, чтобы избежать
        // утечки открытого пароля
        Array.Clear(data, 0, data.Length);
    }
    return hash;
}
/*Метод GenerateRandom генерирует массив указанной длины,
 * состоящий из случайных байтов.
 */
private static byte[] GenerateRandom(int size)
{
    byte[] random = new byte[size];
    RandomNumberGenerator.Create().GetBytes(random);
    return random;
}
}
}
}

```

Выводы по разделу

Приведены выбранные программные средства разработки и обоснована их актуальность. Описаны таблицы базы данных SQL Server, используемые при разработке клиентского приложения. Описана структура разработанного приложения и интерфейс программы, а именно, все справочники и основные формы. Приведены примеры их использования.

3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

При принятии решения о создании и внедрении автоматизированной системы формирования отчетности необходимо рассчитать экономическую эффективность и целесообразность внедрения разработки. Для определения экономической эффективности необходимо рассчитать и проанализировать эффект окупаемости. Для определения эффекта окупаемости используется методика срока окупаемости, для этого рассчитываются затраты до внедрения и после внедрения проекта.

1.10 Расчет затрат

Затраты на приобретение ресурсов, необходимых для разработки программного продукта отражены в таблице 8.

Таблица 10 – Затраты на ПО и ПК

<u>Наименование ПО</u>	<u>Стоимость, руб</u>
<u>Microsoft Visual Studio 2013</u>	<u>25 870</u>
<u>ПК</u>	<u>22 000</u>
<u>Итого:</u>	<u>47870,00</u>

Расчет заработной платы работников, разрабатывающих и внедряющих проект, приведен в таблице 9.

Таблица 11 – Расчет заработной платы

<u>Должность</u>	<u>Оклад руб/мес.</u>	<u>Оплата руб/день</u>	<u>Продолжительность работ, дни</u>	<u>Итого, руб.</u>
<u>Программист</u>	<u>30_000</u>	<u>1000</u>	<u>20</u>	<u>22000</u>
<u>Дополнительная заработная плата</u>		<u>150</u>	<u>2</u>	<u>4500</u>
<u>Социальные платежи (26% основной и дополнительной заработной платы)</u>			<u>20</u>	<u>6890,88</u>

Затраты на разработку и внедрение программного продукта составляют 81 260,48 руб.

1.81.11 Анализ убытков

По итогам предоставления отчетности за полный месяц компания ООО Автомир-74 получает возмещение (компенсацию) предоставленной скидки на каждый проданный автомобиль. Компенсация варьируется от 30 до 200 тысяч рублей, в зависимости от схемы продажи, модели нового автомобиля и его конечной стоимости. В таблице 10 приведены пороги возмещений по моделям автомобилей.

Таблица 12 – Пороги возмещения денежных средств

Модель автомобиля	Пороги возмещения (руб.)
TOYOTA COROLLA	30 000 - 50 000
TOYOTA CAMRY	50 000 - 80 000
TOYOTA RAV4	70 000 - 120 000
TOYOTA LC150,LC200	80 000 - 200 000

При ошибках, недостающих документах в предоставленной отчетности, компания не получает возмещение и не имеет возможности подать документы повторно, что несет крупные убытки.

Анализируя данные, приведенные в таблице 10 в сопоставлении со средствами, которые необходимо затратить на разработку программного обеспечения, можно сделать вывод, что данное решение может окупить себя в первый месяц эксплуатации.

Выводы по разделу

Проанализированы и рассчитаны статьи затрат на разработку и внедрение, вычислена цена продукта. Стоимость разработки программного продукта сопоставима с возможными убытками компании.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы поставлены цели и задачи, обоснована актуальность и необходимость выбранной темы. Проанализированы существующие информационные системы, выявлены их преимущества и недостатки.

Информационная системы разработана на языке программирования C# на платформе .NET с использованием Microsoft SQL Server.

Приведены схемы базы данных и описаны используемые в программе таблицы базы данных. Разработана структура, интерфейс приложения, описана работа отдельных частей программы, составляющих основной функционал.

В экономической части работы подсчитаны затраты на разработку и внедрение программного продукта, проанализирован срок окупаемости внедрения программы, т.е. вычислен экономический эффект.

Цель работы – разработка системы электронного документооборота – достигнута. В полной мере решены поставленные задачи.

- Настроено организованное взаимодействие между отделами организации для сбора информации по предоставлению отчетности.
- Структурировано хранение электронных документов и автоматизирована проверка предоставляемых документов на правильность.
- Сокращены финансовые риски организации.
- Исключены сложность и монотонность выполняемых операций, сокращена доля человеческих манипуляций с данными и файлами.

ПРИЛОЖЕНИЕ А

Бланк согласия на обработку персональных данных

Приложение 3. Образец согласия на обработку персональных данных

СОГЛАСИЕ НА ОБРАБОТКУ ПЕРСОНАЛЬНЫХ ДАННЫХ

Я, _____,
(ФИО)

паспорт _____ выдан _____,
(серия, номер) (квартал и кем выдан)

адрес регистрации: _____

даю свое согласие на обработку в _____,
(наименование организации)

моих персональных данных, относящихся исключительно к перечисленным ниже категориям персональных данных: фамилия, имя, отчество; пол; дата рождения; тип документа, удостоверяющего личность; данные документа, удостоверяющего личность; гражданство; информация о выбранных экзаменах; информация о результатах итогового сочинения (изложения), информация об отнесении участника единого государственного экзамена к категории лиц с ограниченными возможностями здоровья, инвалидов; информация о результатах экзаменов.

Я даю согласие на использование персональных данных исключительно в целях формирования федеральной информационной системы обеспечения проведения государственной итоговой аттестации обучающихся, освоивших основные образовательные программы основного общего и среднего общего образования, и приема граждан в образовательные организации для получения среднего профессионального и высшего образования и региональной информационной системы обеспечения проведения государственной итоговой аттестации, а также хранение данных об этих результатах на электронных носителях.

Настоящее согласие предоставляется мной на осуществление действий в отношении моих персональных данных, которые необходимы для достижения указанных выше целей, включая (без ограничения) сбор, систематизацию, накопление, хранение, уточнение (обновление, изменение), использование, передачу третьим лицам для осуществления действий по обмену информацией (органу исполнительной власти, осуществляющему полномочия в сфере образования, Федеральному бюджетному государственному учреждению «Федеральный центр тестирования», Федеральной службе по надзору в сфере образования и науки), обезличивание, блокирование персональных данных, а также осуществление любых иных действий, предусмотренных действующим законодательством РФ.

Я проинформирован, что _____ гарантирует.
(наименование организации)

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Шандров, Б.В. Технические средства автоматизации. / Б.В. Шандров, А.Д. Чудаков. – М.: Академия, 2007. – 368 с.
2. Бобылева, М.П. Управленческий документооборот: от бумажного к электронному. Вопросы теории и практики. / М.П. Бобылева - М.: Изд-во «ТЕРМИКА», 2016. – 360 с.
3. Вишневский, А. В. Microsoft SQL Server. Эффективная работа./ А.В.Вишневский. – Санкт-Петербург: Изд-во Питер, 2009. – 540 с.
4. Справочник языка T-SQL [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/sql/t-sql/>.
5. Справочник языка программирования с# [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/>
- 5.6. Сартасов, Е.М. Объектно-ориентированное программирование. Учебное пособие 004.4(07) С207 – Челябинск: Издательский центр ЮУрГУ, 2014 – 55 с.
- 6.7. ProfessorWeb - C# 5.0 и платформа .Net 4.5 [Электронный ресурс]. – Режим доступа: <https://professorweb.ru/>
- 7.8. Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ