

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

вычислительной техники

Кафедра

МОиПЭВМ

Направление подготовки 09.03.02 «Информационные системы и технологии»

БАКАЛАВРСКАЯ РАБОТА

на тему

Информационная система управления сервисным центром на платформе 1С:Предприятие с веб-интерфейсом

Студент

(подпись, дата)

Белякова К.Д.

(ФИО полностью)

Руководитель

(подпись, дата)

Попова Н.А.

(фамилия, инициалы)

Нормоконтролёр

(подпись, дата)

Такташкин Д.В.

(фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой

(подпись)

Козлов А.Ю.

(фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК

(подпись)

Попова Н.А.

(фамилия, инициалы)

Пенза, 2023

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
Вычислительной техники

Кафедра
МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
_____ А.Ю. Козлов
«__» _____ 2023 г

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА**

1. Студент Белякова Кристина Дмитриевна гр. 19ВИ1 факультета ВТ
<Фамилия, имя, отчество полностью>

2. Тема работы Информационная система управления сервисным центром на платформе 1С:Предприятие с веб-интерфейсом

Тема утверждена приказом ПГУ № _____ от " _____ " _____ 20 _____

3. Руководитель работы к.т.н., доцент кафедры «МО и ПЭВМ» Попова Н.А.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Назначение: автоматизация деятельности предприятия, оказывающее услуги ремонта бытовой техники.

Основные функции системы:

1. Ведение нормативно-справочной информации о номенклатуре, контрагентах, сотрудниках;
2. Ведение учета о поступлении техники от контрагента;
3. Ведение учета о возврате техники контрагенту;
4. Ведение учета закупок комплектующих для бытовой техники у поставщиков;
5. Ведение учета реализации услуг по ремонту бытовой техники;
6. Формирование печатных форм заявок;
7. Формирование аналитических отчетов.

Технология разработки: предметно-ориентированное программирование.

Языки программирования: встроенный язык программирования 1С: Предприятие, JavaScript.

Среда программирования: технологическая платформа 1С: Предприятие 8.3, JetBrains WebStorm.

5. Перечень подлежащих разработке вопросов

1. Анализ предметной области и постановка задачи; _____

2. Проектирование программных средств; _____

3. Реализация программных средств; _____

4. Тестирование программных средств; _____

5. Оформление ВКР. _____

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Анализ предметной области и требований к системе	10%	12.04.2023 – 16.04.2023	
2	Проектирование программных средств	20%	17.04.2023 – 26.04.2023	
3	Реализация программных средств	30%	27.04.2023 – 11.05.2023	
4	Тестирование программных средств	20%	12.05.2023 – 21.05.2023	
5	Оформление пояснительной записки	20%	22.05.2023 – 01.06.2023	

Дата выдачи задания «22» декабря 2022г.

Руководитель бакалаврской работы _____
(подпись, дата)

Задание к исполнению принял студент _____
(подпись, дата)

Работу к защите допустить
Декан ФВТ _____ Л.Р. Фионова
(подпись, дата)

Реферат

Пояснительная записка содержит 132 листа, 68 рисунков, 12 таблиц, 1 листинг, 16 использованных источников и 5 приложений.

1С:ПРЕДПРИЯТИЕ 8.3, КОНФИГУРАЦИЯ, АВТОМАТИЗАЦИЯ, ИНФОРМАЦИОННАЯ СИСТЕМА, УЧЕТ ДЕЯТЕЛЬНОСТИ, СЕРВИСНЫЙ ЦЕНТР, МОДУЛЬ МЕНЕДЖЕРА, МОДУЛЬ ФОРМЫ

Предмет работы: разработка информационной системы управления сервисным центром.

Объект работы: автоматизация деятельности сервисного центра, оказывающих услуги ремонта бытовой техники.

Цель работы: разработка информационной системы управления сервисным центром на платформе 1С:Предприятие с веб-интерфейсом.

Технология разработки: платформа 1С: Предприятие 8.3, React, JavaScript, CSS.

Результаты работы: спроектирована и разработана конфигурация на платформе 1С:Предприятие 8.3 для ведения учета деятельности сервисного центра с веб-интерфейсом.

Область применения: предприятия, предоставляющие услуги по ремонту бытовой техники.

					ПГУ 09.03.02 – 08БР191.04 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Информационная система управления сервисным центром на платформе 1С:Предприятие с веб-интерфейсом Пояснительная записка	Лит.	Лист	Листов
Разраб.		Белякова К.Д.					4	132
Провер.		Попова Н.А.						
Реценз.								
Н. Контр.		Такташкин Д.В.						
Утв.		Козлов А.Ю.				группа 19ВИ1		

Содержание

Введение.....	7
1 Постановка задачи и анализ аналогов разрабатываемой системы	9
1.1 Постановка задачи	9
1.2 Анализ информационных систем аналогичного назначения.....	10
1.3 Выбор технологий проектирования и разработки	15
2 Планирование и расчет бюджета разрабатываемой системы	18
3 Проектирование информационной системы.....	23
3.1 Анализ бизнес-процессов предприятия	23
3.2 Проектирование логической модели базы данных	26
3.3 Проектирование программного обеспечения	31
3.3.1 Анализ прецедентов	31
3.3.2 Диаграмма состояний	34
3.3.3 Диаграмма деятельности	35
3.3.4 Диаграмма последовательности	37
3.3.5 Диаграмма кооперации	39
4 Реализация информационной системы.....	41
4.1 Реализация структур программного средства	41
4.2 Реализация конфигурации информационной системы.....	60
4.3 Реализация веб-интерфейса.....	63
4.4 Реализация пользовательского интерфейса.....	65
5 Тестирование информационной системы	72
5.1 Функциональное тестирование	72
5.2 Метрика Холстеда	75
Заключение	78
Список использованных источников	80
Приложение А. Глоссарий	82
Приложение Б. Диаграмма компонентов.....	84
Приложение В. Листинг конфигурации	86

Приложение Г. Листинг веб-интерфейса.....	103
Приложение Д. Результаты тестирования	124

Введение

На сегодняшний день автоматизация бизнес–процессов предприятий оказывает существенное влияние на конкурентоспособность предоставляемых услуг. Цель автоматизации – разработка и внедрение на предприятие информационной системы, которая обеспечивает оптимизацию учета и анализа деятельности организации.

Актуальность работы заключается в том, что создание информационной системы (ИС) с целью автоматизации деятельности сервисного центра по ремонту бытовой техники рассматривается как элемент развития предприятия, что способствует упрощению работы сотрудников и ускорению процесса обслуживания клиентов и ремонта техники. Информационная система обеспечивает сотрудников сервисного центра оперативной информацией о наличии комплектующих на складе, о занятости мастеров. Эта информация необходима для правильного принятия решений о приеме техники в ремонт.

Объектом выпускной квалификационной работы является автоматизация деятельности сервисного центра.

Предметом является информационная система автоматизации деятельности сервисного центра.

Целью выпускной квалификационной работы является разработка информационной системы управления сервисным центром.

Для достижения поставленной цели были определены следующие задачи:

- определить задачи на разработку информационной системы управления сервисным центром;
- провести анализ существующих информационных систем аналогичного назначения;
- определить требования к разрабатываемой информационной системе;
- выполнить функциональное моделирование системы;
- спроектировать информационную систему управления сервисным центром;

- реализовать информационную систему с помощью выбранного средства разработки;
- протестировать разработанную информационную систему.

В результате была спроектирована и разработана конфигурация с веб-интерфейсом, которая позволяет автоматизировать наиболее важные аспекты в работе сервисного центра, вести учет движения материалов и техники в оперативном учете, отслеживать поступление материалов и техники на склад.

1 Постановка задачи и анализ аналогов разрабатываемой системы

1.1 Постановка задачи

Сервисный центр по ремонту бытовой техники является организацией, предоставляющей услуги по ремонту и обслуживанию бытовой техники различных производителей. Клиенты могут обратиться в сервисный центр с проблемами и поломками своей техники, после чего им будет предоставлено качественное обслуживание и ремонт.

В данной предметной области необходимо учитывать следующие аспекты:

- клиенты: физические лица и организации, которые обращаются в сервисный центр для ремонта и обслуживания бытовой техники;
- бытовая техника: различные виды бытовой техники, которые могут требовать ремонта и обслуживания (холодильники, стиральные машины, пылесосы и т.д.);
- специалисты: квалифицированные специалисты, которые проводят ремонт и обслуживание бытовой техники;
- запчасти: различные запчасти, которые могут понадобиться для проведения ремонта.

Для учета клиентов, заказов на ремонт, бытовой техники, специалистов и запчастей необходимо разработать информационную систему управления сервисным центром по ремонту бытовой техники в 1С:Предприятие с веб-интерфейсом. Задачи на разработку информационной системы управления сервисным центром перечислены ниже.

1. Учет клиентов и заказов на ремонт. Необходимо разработать функционал для учета клиентов и заказов на ремонт. Система должна позволять создавать новые заказы на ремонт, просматривать и редактировать существующие заказы, а также фиксировать стоимость ремонта.

2. Учет бытовой техники. Система должна позволять вести учет бытовой техники, которая находится на ремонте. Для каждого заказа должна быть возможность указать тип техники и ее модель.

3. Учет специалистов и их задач. Система должна обеспечивать ведение учета специалистов, которые работают в сервисном центре. Система должна позволять назначать задачи для специалистов и контролировать выполнение этих задач.

4. Учет запчастей. Система должна осуществлять учет запчастей, которые используются при проведении ремонта. Для каждой запчасти должны быть указаны ее наименование, стоимость и количество в наличии.

5. Отчетность. Система должна позволять формировать различные отчеты по работе сервисного центра. Например, отчеты о выполненных работах за определенный период, отчеты о доходах и расходах, отчеты об остатках номенклатуры.

6. Веб-интерфейс. Для удобства система должна иметь веб-интерфейс, который позволит пользователям получать доступ к системе с компьютера, который имеет подключение к интернету. Клиенты смогут следить за статусом своих заказов на ремонт. Веб-интерфейс может быть использован для удобного управления системой со стороны сотрудников сервисного центра. Веб-интерфейс должен обладать простым и понятным интерфейсом, что позволит быстро освоить систему и сэкономить время на обучение сотрудников.

Разработка информационной системы управления сервисным центром по ремонту бытовой техники в 1С:Предприятие с веб-интерфейсом позволит оптимизировать и автоматизировать работу сервисного центра, ускорить процесс ремонта техники и повысить уровень обслуживания клиентов.

1.2 Анализ информационных систем аналогичного назначения

Перед разработкой информационной системы управления сервисным центром необходимо проанализировать системы-аналоги, что позволит учесть достоинства и недостатки уже существующих систем.

В настоящее время существует достаточно большое количество автоматизированных систем управления сервисным центром, различающихся функциональными возможностями, технической реализацией, стоимостью и т.д.

Рассмотрим несколько существующих аналогов разрабатываемой информационной системы: Gincore, Remonline, 1С:Предприятие 8. Управление сервисным центром.

Gincore – облачная программа для учета в сервисном центре или ремонтной мастерской. Включает в себя складскую программу, полноценную бухгалтерию, внутреннюю аналитику, интеграцию с IP-телефонией и Google Analytics [1].

Функциональные возможности сервиса Gincore.

- прием и выдача устройства в ремонт;
- принимая устройство в ремонт, прямо из карточки ремонта можно сделать фото устройства через внешнюю камеру и сохранить фото в заказе;
- вложения снимков сопроводительных документов, если устройство поступило на ремонт по почте;
- бухгалтерия группирует расходы по бухгалтерским статьям и легко можно получить отчёт по любой статье расходов;
- автоматизированный модуль закупок, расчёт количества товара для закупки на склад на основе данных по продажам прошлых периодов и тенденций к увеличению или уменьшению спроса;
- интегрированная бухгалтерская и складская программа для учёта оприходований, списаний товаров, выдачи денежных средств и точной истории транзакций по каждому контрагенту с финальным балансом;
- удобные модули для ведения заказов клиентов, отдела закупок и контроля сроков выполнения.

Если запчасти нет на складе, инженер прямо в карточке ремонта делает запрос на закупку запчасти, менеджер получает запрос, он отображается у него как необработанный. И, если в течении 2х дней не закажет запчасть, то ремонт начнёт светиться красным цветом сигнализируя о нарушении сроков.

Стоимость программы Gincore: от 1500 Р в месяц.

Интерфейс Gincore представлен на рисунке 1.

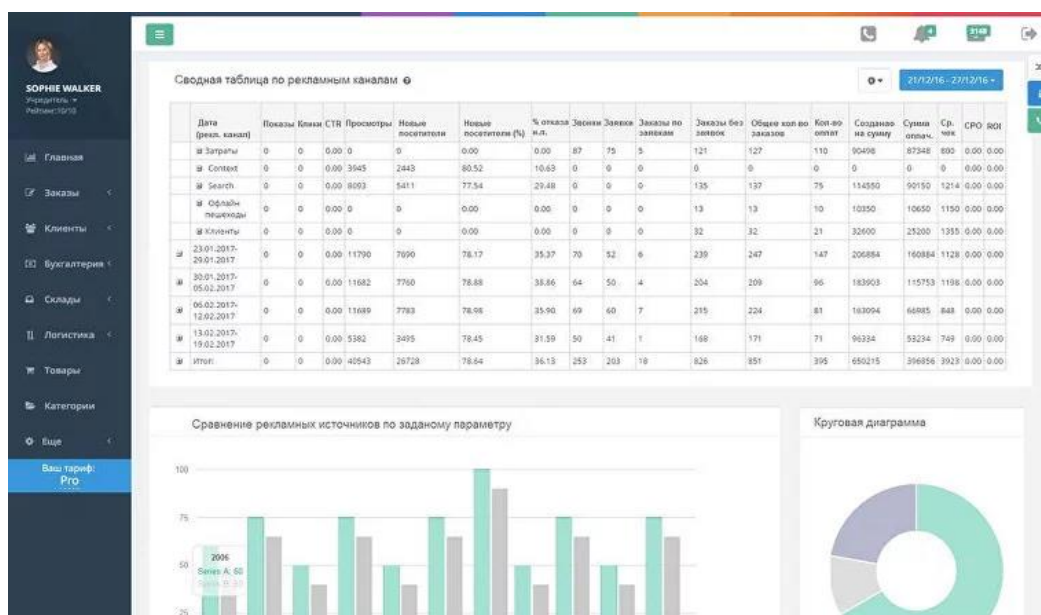


Рисунок 1 – Интерфейс сервиса Gincore

Следующий программный продукт для рассмотрения – облачный сервис РемОнлайн, который позиционирует себя как платформа для автоматизации сферы услуг. Подходит сервисным центрам и ремонтным мастерским.

РемОнлайн позволяет обрабатывать заказы, контролировать сотрудников, вести складской и финансовый учет, печатать документы и отчеты, быстро анализировать показатели компании и принимать эффективные управленческие решения. Поддерживает более 200 операторов IP-телефонии, автоматически отправляет SMS клиенту о готовности заказа, а менеджера уведомляет по email о ходе работы над заказом [2].

Кроме того, система РемОнлайн имеет такие специфических функции, как: учет по серийным номерам и адресное хранение. Этот функционал может быть полезен для сервисных центров с большим складским оборотом, однако это будет стоить дороже.

Стоимость программы зависит от числа мастерских и количества сотрудников: от 33\$ до 55\$ в месяц.

Интерфейс РемОнлайн представлен на рисунке 2.

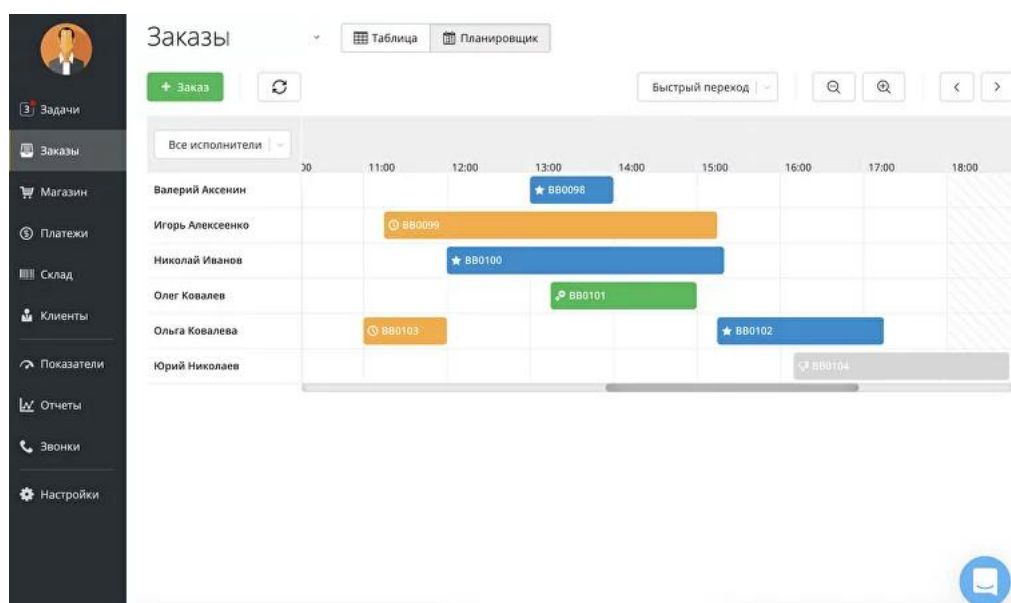


Рисунок 2 – Интерфейс системы РемОнлайн

Последнее из рассматриваемых программных решений – 1С:Предприятие 8. Управление сервисным центром, которое предназначено для автоматизации деятельности предприятий, оказывающих услуги гарантийного и послегарантийного ремонта, обслуживания бытовой и компьютерной техники и различного оборудования [3].

Решение 1С:Предприятие 8. Управление сервисным центром позволяет:

- оптимизировать работу персонала и сократить простои за счет использования механизма по распределению загрузки мастеров, в том числе и выездных;
- оперативно отслеживать заявки на ремонт и техобслуживание, стадию их исполнения, стоимость работ;
- оптимизировать заказ запчастей производителям и поставщикам с учетом потребностей сервисного центра и имеющихся заявок клиентов;
- контролировать задолженность по взаиморасчетам;
- повысить эффективность работы компании, предоставляя владельцам и руководителям широкий спектр инструментов для управления, а сотрудникам – новые возможности для продуктивной ежедневной работы.

Стоимость программного решения 1С:Предприятие 8. Управление

сервисным центром: 26 400 руб.

Интерфейс 1С:Предприятие 8. Управление сервисным центром представлен на рисунке 3.

Дата	Тип документа	Текущий этап	Номенклатура	Принадлежность товара	Серийный номер	Дата окончания
04.11.2022 17:11:58	Выезд мастера	Выезд мастера	Canon iR 3225N	ЦТО "ТехноСервис"		01.11.2022
04.11.2022 17:12:27	Выезд мастера	Выезд мастера	Canon iR 3225N	ЦТО "ТехноСервис"		01.11.2022
04.11.2022 17:15:02	Выезд мастера	Выезд мастера	Canon iR 3225N	ЦТО "ТехноСервис"		01.11.2022
24.11.2022 14:52:53	Принем в ремонт	Возврат из ремонта	Стиральная машина Samsung W...	Товар клиента	03	

Рисунок 3 – Интерфейс 1С:Предприятие 8. Управление сервисным центром

Рассмотрев функциональные характеристики каждого аналога, был проведен сравнительный анализ, результаты которого приведены в таблице 1.

Таблица 1 – Сравнительный анализ систем-аналогов

Функциональные возможности	Gincore	РемОнлайн	1С:Предприятие 8. Управление сервисным центром	Разрабатываемая система
Работа с документами	Да	Да	Да	Да
Email-уведомления клиентов	Нет	Да	Да	Да
Личный кабинет клиента	Нет	Нет	Нет	Да
Платформы	Веб-приложение	Веб-приложение	1С:Предприятие	1С:Предприятие и веб-приложение
Учёт расходных материалов	Нет	Да	Да	Да
Учет обращений (заявок на ремонт)	Да	Да	Да	Да

Перечисленные системы аналогичного действия используются только на одной платформе. В ходе анализа было принято решение разработать систему на платформе 1С:Предприятие с поддержкой веб-интерфейса, чтобы у пользователя был доступ в систему с любого компьютера, имеющий доступ в интернет. В системах аналогичного назначения нет возможности создания личного кабинета

для клиентов, поэтому в разрабатываемой системе было принято решение добавить доступ к системе для клиентов организации, которые смогут следить за статусом своих заказов на ремонт бытовой техники.

В соответствии с характеристиками, приведенными в описании аналогичных систем, проведено сравнение по следующим критериям: функциональная сложность системы, качество пользовательского интерфейса, производительность, стоимость системы. Результаты представлены в таблице 2.

Таблица 2 – Оценка систем, основанная на критериях аналогов

Системы	Функциональная сложность	Качество пользовательского интерфейса	Производительность	Стоимость
Gincore	Высокая	Среднее	Высокая	Средняя
РемОнлайн	Средняя	Среднее	Высокая	Высокая
1С:Предприятие 8. Управление сервисным центром	Высокая	Низкое	Средняя	Высокая

Из анализа каждого аналога с точки зрения выделенных критериев можно сделать вывод, что существующие информационные системы управления сервисным центром являются дорогими и имеют избыточный функционал, что не подходит для использования в небольших компаниях. Из-за этого возникает необходимость в разработке системы, которая будет включать в себя только наиболее важные функции для малых сервисных центров. Такое решение будет оптимальным, так как оно сократит затраты на приобретение программного решения и время обучения сотрудников компании, благодаря отсутствию избыточного функционала.

1.3 Выбор технологий проектирования и разработки

Для проектирования информационной системы управления сервисным центром требуется построение UML-диаграмм, функциональной диаграммы в нотации IDEF0, а также модели данных в нотации IDEF1X. Для построения перечисленных диаграмм были выбраны такие CASE-средства, как MS Visio 2021, Erwin Data Modeler.

Microsoft Visio 2021 – программа для создания графических схем и диаграмм с широким набором инструментов визуализации для наглядного представления сложных данных, таких как принципиальные схемы, комплексные системы многоуровневых процессов, блок-схемы или диаграммы. MS Visio может использоваться для создания различных типов диаграмм, таких как блок-схемы, организационные диаграммы, диаграммы потоков работы, диаграммы Ганта, UML-диаграммы [4].

Erwin Data Modeler – программа для моделирования данных, которая предоставляет возможность проектирования БД, включая логические, физические, архитектурные и сложные сценарии интеграции данных, необходимые для эффективной работы в области управления данными [5].

При выборе CASE-средств для проектирования информационной системы мы учитываем следующие факторы: наличие необходимого функционала, удобный пользовательский интерфейс и легкость использования.

Для разработки информационной системы управления сервисным центром были выбраны следующие технологии и инструменты: платформа 1С:Предприятие 8.3, язык программирования JavaScript, библиотека React, интегрированная среда разработки JetBrains WebStorm, веб-сервер Apache.

Платформа 1С:Предприятие 8.3 является мощным инструментом для автоматизации бизнес-процессов и создания прикладных решений любой сложности. Платформа обладает высокой степенью защиты данных и соответствует современным требованиям безопасности. Она поддерживает различные режимы доступа к данным и функционалу системы, а также имеет механизмы защиты от внешних угроз [6].

Для разработки веб-интерфейса было принято решение использовать библиотеку React, которая работает на языке программирования JavaScript. React работает на основе компонентного подхода, что делает разработку более модульной и удобной. Она основывается на принципах обновления данных в режиме реального времени и взаимодействия с пользователем через события и обратные вызовы. React также помогает ускорить время разработки и улучшить

производительность приложений, т.к. использует виртуальный DOM (Document Object Model), который позволяет обновлять только те части страницы, которые действительно изменились. Это увеличивает производительность приложения и уменьшает количество запросов к серверу [7].

В качестве инструмента для установки и настройки приложения на сервере принято решение использовать Apache. Назначение Apache веб-сервера заключается в обслуживании веб-страниц и веб-приложений, обработке запросов и доставке содержимого по протоколу HTTP. Благодаря своей надёжности, масштабируемости и модульной архитектуре, Apache является одним из наиболее популярных и распространенных веб-серверов в мире [8].

Выбранные технологии разработки и проектирования информационных систем позволяют с успехом выполнить необходимую работу, одновременно являясь при этом удобными и эффективными в использовании.

2 Планирование и расчет бюджета разрабатываемой системы

Для успешного управления проектами требуется определить конечные цели и обеспечить их выполнение, путем планирования необходимых ресурсов, времени, качества и бюджета проекта, а также проведения оценки возможных рисков и трудозатрат. Наличие подробного плана заранее также играет ключевую роль в обеспечении эффективного проектного управления.

Для планирования и расчета бюджета разработки информационной системы управления сервисным центром на платформе 1С:Предприятие с веб-интерфейсом необходимо пройти этапы, перечисленные ниже.

1. Определить объем и спецификации функциональных требований. Это позволит определить, какие модули и функции системы будут необходимы.

2. Оценить трудозатраты и затраты на разработку системы. Нужно учитывать затраты на анализ требований, проектирование, реализацию, тестирование и документацию. Также нужно учитывать затраты на оборудование.

3. Составить план работ, расписание и бюджет. На основе оценки трудозатрат и затрат можно определить сроки выполнения работ и необходимый бюджет. Расписание позволит управлять проектом и отслеживать выполнение работ в соответствии со сроками.

4. Оценить экономическую эффективность системы.

Существует несколько формализованных методов для планирования и управления выполнением проектных работ. Один из основных методов – это построение линейного графика Ганта.

Диаграмма Ганта – это инструмент визуализации временных интервалов выполнения задач и проектов. Она позволяет представить графически все этапы проекта, от начала до конца, и определить продолжительность каждой задачи [9].

Назначение диаграммы Ганта – помочь определить общую продолжительность проекта, сроки выполнения каждой задачи, количество ресурсов, необходимых для выполнения проекта, а также выявить возможные

риски и проблемы, которые могут возникнуть в процессе выполнения задач.

Построение диаграммы Ганта для осуществления планирования и расчета бюджета разрабатываемой системы выполнялось в программе для управления проектами Microsoft Office Project.

В ходе планирования проекта было выделено 25 задач, описание которых представлено в таблице 3.

Таблица 3 – Задачи проекта

№ п/п	Название задачи	Длительность	Начало	Окончание	Предшественники
1	Начало проекта	0 дней	12.04.2023	12.04.2023	
2	Анализ предметной области и требований к системе	4 дня	12.04.2023	16.04.2023	
3	Анализ бизнес-процессов предприятия	1 день	12.04.2023	12.04.2023	
4	Поиск и анализ информационных систем аналогичного назначения	1 день	13.04.2023	13.04.2023	3
5	Анализ требований и постановка задач	1 день	14.04.2023	14.04.2023	4
6	Выбор технологий проектирования и разработки	1 день	15.04.2023	15.04.2023	5
7	Завершение анализа предметной области и требований к системе	0 дней	16.04.2023	16.04.2023	2
8	Проектирование информационной системы	9 дней	17.04.2023	26.04.2023	7
9	Проектирование логической модели базы данных	4 дня	17.04.2023	20.04.2023	
10	Проектирование структуры данных	2 дня	21.04.2023	22.04.2023	9
11	Проектирование пользовательского интерфейса	3 дня	23.04.2023	26.04.2023	10
12	Завершение проектирования информационной системы	0 дней	26.04.2023	26.04.2023	8
13	Реализация информационной системы	11 дней	27.04.2023	11.05.2023	12
14	Реализация конфигурации	4 дня	27.04.2023	01.05.2023	
15	Реализация веб-интерфейса системы	7 дней	03.05.2023	11.05.2023	14
16	Завершение реализации информационной системы	0 дней	11.05.2023	11.05.2023	13
17	Тестирование программных средств	9 дней	12.05.2023	21.05.2023	16
18	Тестирование метрикой Холстеда	5 дней	12.05.2023	17.05.2023	
19	Функциональное тестирование	4 дня	18.05.2023	21.05.2023	
20	Завершение тестирования программных средств	0 дней	21.05.2023	21.05.2023	17
21	Оформление документации	10 дней	22.05.2023	01.06.2023	20

Продолжение таблицы 3

22	Написание документации	7 дней	22.05.2023	28.05.2023	
23	Оформление и печать документации	3 дня	29.05.2023	31.06.2023	22
24	Завершение оформления документации	0 дней	01.06.2023	01.06.2023	21
25	Конец проекта	0 дней	01.06.2023	01.06.2023	24

Также были определены необходимые ресурсы. Список ресурсов представлен в таблице 4.

Таблица 4 – Ресурсы и затраты проекта

№ п/п	Название ресурса	Тип ресурса	Стандартная ставка	Начисление
1	Аналитик	Трудовой	200,00 Р/ч	Пропорциональное
2	Проектировщик	Трудовой	200,00 Р/ч	Пропорциональное
3	Программист	Трудовой	250,00 Р/ч	Пропорциональное
4	Тестировщик	Трудовой	200,00 Р/ч	Пропорциональное
5	Компьютер	Затраты		В начале
6	Принтер	Затраты		В начале
7	Бумага	Материальный	300 Р	В начале
8	Интернет	Затраты		В начале
9	Электроэнергия	Затраты		По окончании

Выделенные задачи и ресурсы были занесены в проект в Microsoft Office Project. Результаты представлены на рисунке 4.

	Режим задачи	Название задачи	Длительность	Начало	Окончание	Предшественник	Названия ресурсов
1	✓	Начало проекта	1 день	Ср 12.04.23	Ср 12.04.23		Бумага[1];Интерне
2	✓	Анализ предметной области и требований к системе	5 дней	Ср 12.04.23	Вс 16.04.23		
3	✓	Анализ бизнес-процессов предприятия	1 день	Ср 12.04.23	Ср 12.04.23		Аналитик
4	✓	Поиск и анализ информационных систем аналогичного назначения	1 день	Чт 13.04.23	Чт 13.04.23	3	Аналитик
5	✓	Анализ требований и постановка задач	1 день	Пт 14.04.23	Пт 14.04.23	4	Аналитик
6	✓	Выбор технологий проектирования и разработки	1 день	Сб 15.04.23	Сб 15.04.23	5	Аналитик
7	✓	Завершение анализа предметной области и требований к системе	0 дней	Пн 17.04.23	Пн 17.04.23	2	
8	✓	Проектирование информационной системы	9 дней	Пн 17.04.23	Ср 26.04.23	7	
9	✓	Проектирование логической модели базы данных	4 дней	Пн 17.04.23	Чт 20.04.23	7	Проектировщик
10	✓	Проектирование структуры данных	2 дней	Пт 21.04.23	Сб 22.04.23	9	Проектировщик
11	✓	Проектирование пользовательского интерфейса	3 дней	Пн 24.04.23	Ср 26.04.23	10	Проектировщик
12	✓	Завершение проектирования информационной системы	0 дней	Чт 27.04.23	Чт 27.04.23	8	
13	✓	Реализация информационной системы	11 дней	Чт 27.04.23	Чт 11.05.23	12	
14	✓	Реализация конфигурации	4 дней	Чт 27.04.23	Пн 01.05.23		Программист
15	✓	Реализация веб-интерфейса системы	8 дней	Вт 02.05.23	Чт 11.05.23	14	Программист
16	✓	Завершение реализации информационной системы	0 дней	Пт 12.05.23	Пт 12.05.23	13	
17	✓	Тестирование программных средств	8 дней	Пт 12.05.23	Вс 21.05.23	16	
18	✓	Тестирование метрикой Холстеда	5 дней	Пт 12.05.23	Ср 17.05.23		Тестировщик
19	✓	Функциональное тестирование	4 дней	Чт 18.05.23	Вс 21.05.23		Тестировщик
20	✓	Завершение тестирования программных средств	0 дней	Пн 22.05.23	Пн 22.05.23	17	
21	✓	Оформление документации	9 дней	Пн 22.05.23	Ср 31.05.23	20	
22	✓	Написание документации	7 дней	Пн 22.05.23	Вс 28.05.23		Аналитик
23	✓	Оформление и печать документации	3 дня	Пн 29.05.23	Ср 31.05.23	22	Аналитик
24	✓	Завершение оформления документации	0 дней	Чт 01.06.23	Чт 01.06.23	21	
25	✓	Конец проекта	0 дней	Чт 01.06.23	Чт 01.06.23	24	Электроэнергия[1]

Рисунок 4 – Задачи проекта с распределенными по ним ресурсами в Microsoft Office Project

В соответствии с занесенными задачами и ресурсами была построена

диаграмма Ганта, представленная на рисунке 5.

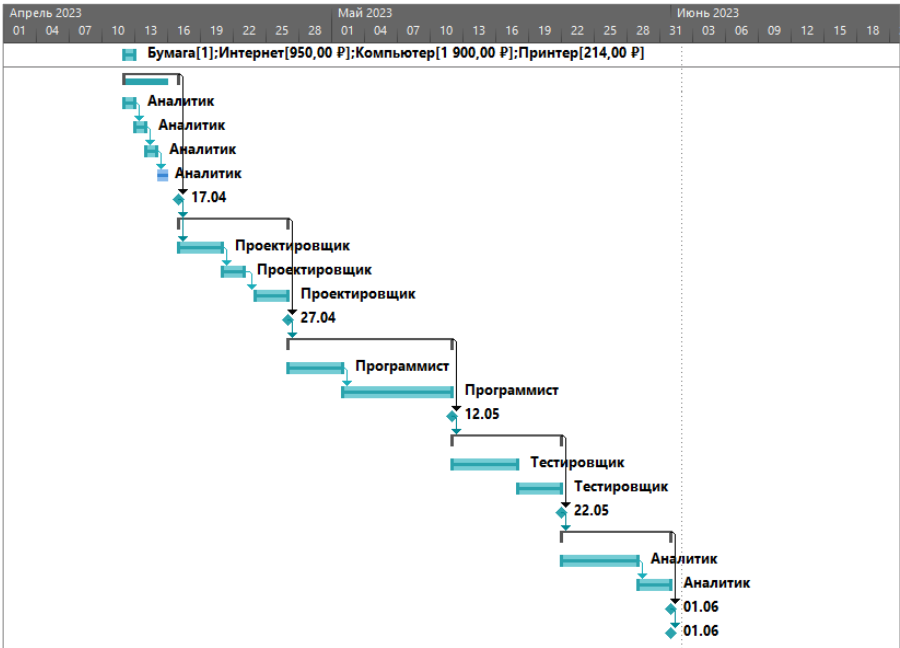


Рисунок 5 – Диаграмма Ганта

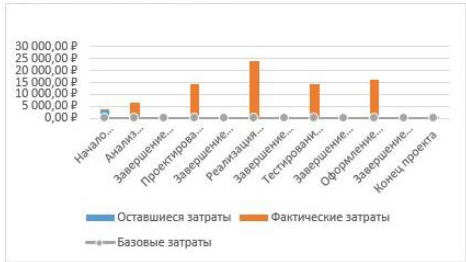
В соответствии с построенной диаграммой был сформирован отчет по затратам на разработку, который представлен на рисунке 6.

ОБЗОР ЗАТРАТ

CP 12.04.23 - ЧТ 01.06.23



СОСТОЯНИЕ ЗАТРАТ
Состояние затрат для всех задач верхнего уровня. Равны ли ваши базовые затраты нулю?
[Попытка задания в качестве базового плана](#)



СОСТОЯНИЕ ЗАТРАТ					
Состояние затрат для задач верхнего уровня.					
Название	Фактически ие затраты	Оставшиеся затраты	Базовые затраты	Затраты	Отклонение по стоимости
Начало проекта	300,00 P	3 064,00 P	0,00 P	3 364,00 P	3 364,00 P
Анализ предметной области и требований к системе	6 400,00 P	0,00 P	0,00 P	6 400,00 P	6 400,00 P
Завершение анализа предметной области и требований к системе	0,00 P	0,00 P	0,00 P	0,00 P	0,00 P
Проектирование информационной системы	14 400,00 P	0,00 P	0,00 P	14 400,00 P	14 400,00 P
Завершение проектирования информационной системы	0,00 P	0,00 P	0,00 P	0,00 P	0,00 P
Реализация информационной системы	24 000,00 P	0,00 P	0,00 P	24 000,00 P	24 000,00 P
Завершение реализации информационной системы	0,00 P	0,00 P	0,00 P	0,00 P	0,00 P
Тестирование программных средств	14 400,00 P	0,00 P	0,00 P	14 400,00 P	14 400,00 P
Завершение тестирования программных средств	0,00 P	0,00 P	0,00 P	0,00 P	0,00 P
Оформление документации	16 000,00 P	0,00 P	0,00 P	16 000,00 P	16 000,00 P
Завершение оформления документации	0,00 P	0,00 P	0,00 P	0,00 P	0,00 P
Конец проекта	106,00 P	0,00 P	0,00 P	106,00 P	106,00 P

Рисунок 6 – Отчет по затратам проекта

Таким образом, в результате функционально-стоимостного анализа, временные затраты для выполнения проекта составят 51 день, включая выходные и праздничные дни, а стоимость его выполнения, с учетом прочих затрат на ресурсы, составит 78670 рублей. Оценка стоимости является неотъемлемой частью планирования любого проекта. Она позволяет установить возможный бюджет на разработку, определить оптимальный способ расходования средств и контролировать затраты на всех этапах работы над проектом. Это дает возможность руководству и исполнителям проекта понимать, какие ресурсы нужны, чтобы выполнить задачи в соответствии с ожиданиями, анализировать стоимостную эффективность различных решений и планировать нужные корректировки. Более того, правильная оценка стоимости создает базу для принятия качественных решений и снижения рисков в процессе выполнения проекта.

3 Проектирование информационной системы

3.1 Анализ бизнес-процессов предприятия

Перед разработкой информационной системы необходимо провести анализ бизнес-процессов сервисного центра. Для этого необходимо построить функциональную модель в нотации IDEF0, которая предназначена для формализации и описания бизнес-процессов.

Для построения диаграммы бизнес-процессов было выбрано CASE-средство Erwin Data Modeler.

Контекстная диаграмма бизнес-процессов в нотации IDEF0 для предметной области сервисного центра представлена на рисунке 7.

Контекстная диаграмма состоит из единственного функционального блока, который служит общим описанием системы и ее взаимодействие с внешней средой.

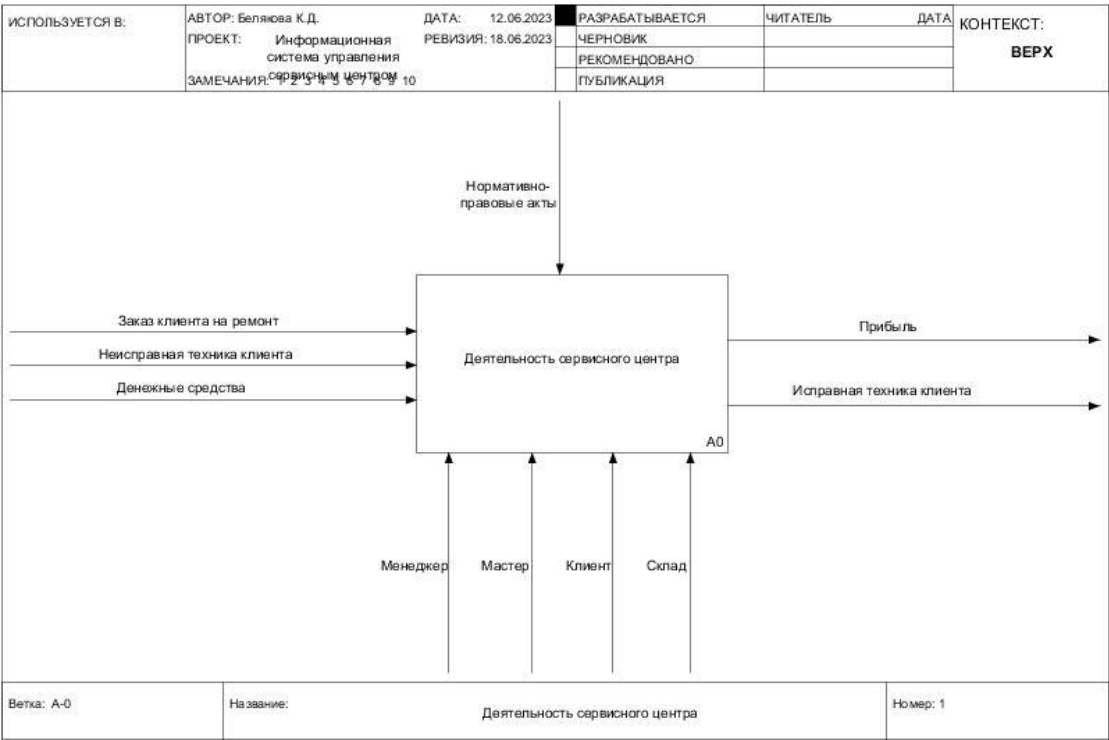


Рисунок 7 – Контекстная диаграмма бизнес-процессов сервисного центра в нотации IDEF0

Основная функция «Деятельность сервисного центра», представленная на

контекстной диаграмме, была декомпозирована на составные части путем создания низкоуровневой дочерней диаграммы. В результате декомпозиции основного процесса «Деятельность сервисного центра» было выделено 4 дочерних процесса: «Выполнение диагностики техники», «Закупка запчастей», «Выполнение ремонта техники» и «Выставление счета за услуги и возврат техники клиенту». Диаграмма декомпозиции представлена на рисунке 8.

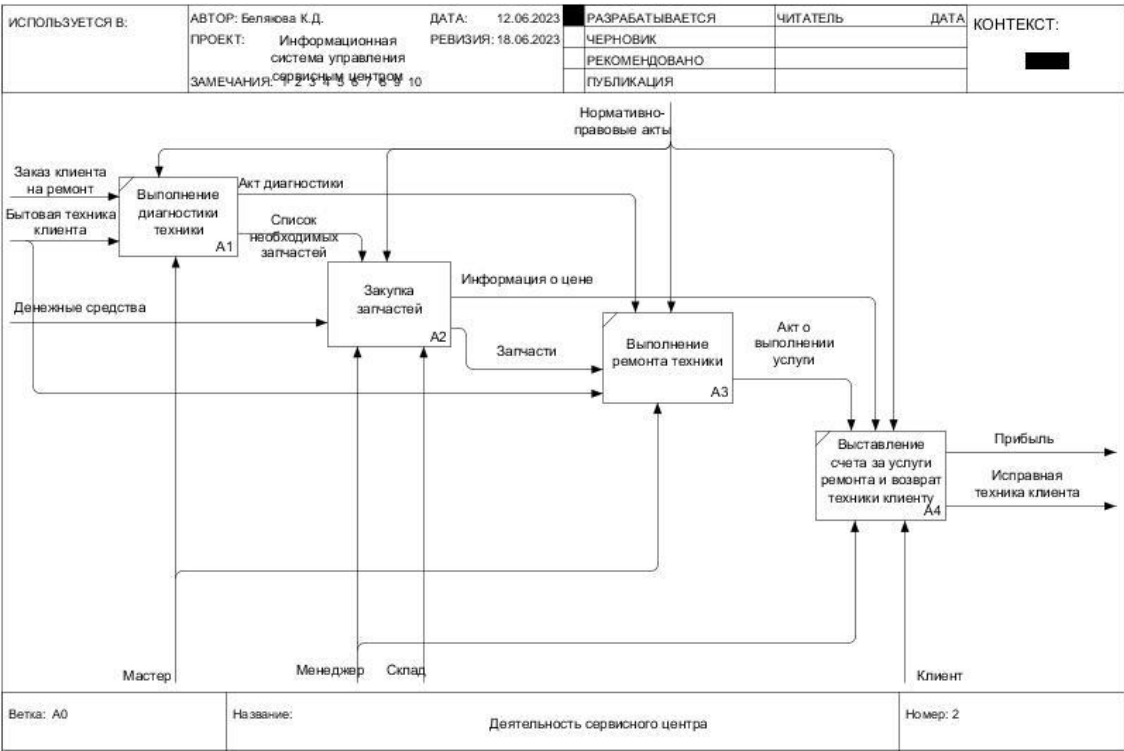


Рисунок 8 – Детализирующая диаграмма процесса «Деятельность сервисного центра» в нотации IDEF0

Процесс «Выполнение диагностики техники» предполагает выполнение технической диагностики бытовой техники, определение неисправностей, составление акта диагностики, а также согласование с клиентом возможности и стоимость ремонта.

Процесс «Закупка запчастей» предполагает оформление и оплату заказа, получение необходимых запчастей от поставщика. На данном этапе пополняются запасы запчастей сервисного центра и устанавливаются цены на поступившие детали техники для расчета стоимости услуги по ремонту бытовой техники.

Процесс «Выполнение ремонта техники» предполагает выполнение ремонтных работ, замену деталей, а также проведение повторной диагностики с целью проверки исправности техники.

Процесс «Выставление счета за услуги и возврат техники клиенту» предполагает получение оплаты от клиента за предоставленные услуги по ремонту и передачу техники клиенту.

Декомпозиция процесса «Закупка запчастей» в нотации IDEF0 представлена на рисунке 9.

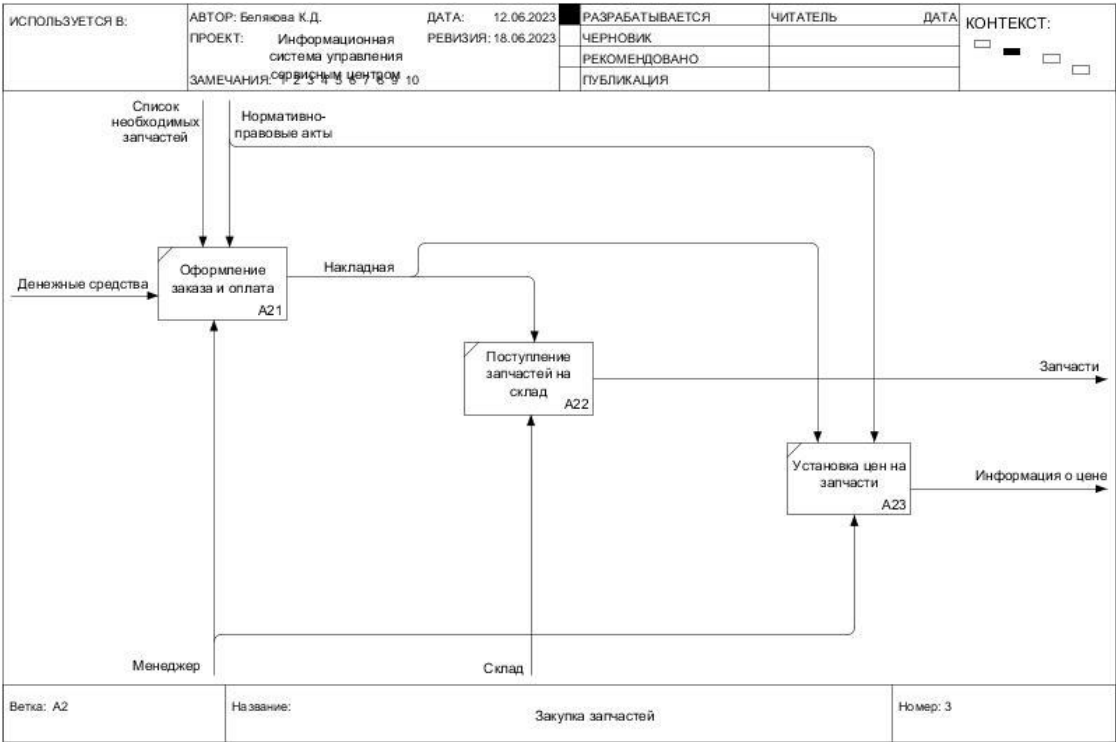


Рисунок 9 – Детализирующая диаграмма процесса «Закупка запчастей»

Закупка запчастей начинается с оформления заказа в соответствии с информацией о необходимости запчастей. После оплаты заказа выдается накладная, в соответствии с которыми поставщик выполняет заказ. После получения заказанных запчастей они размещаются на складе. Затем производится фиксация прихода новых запчастей и установка цен для последующего расчета стоимости услуг.

Описание сущностей и их атрибутов логической модели представлены в таблице 5.

Таблица 5 – Описание сущностей и атрибутов логической модели

Сущность	Атрибут	Роль атрибута	Описание атрибута
Номенклатура	Код номенклатуры	PK	Уникальный код номенклатуры
	Наименование		Наименование номенклатуры
	Код вида номенклатуры	FK	Уникальный код вида номенклатуры
	Код ставки НДС	FK	Уникальный код ставки НДС
Вид номенклатуры	Код вида номенклатуры	PK	Уникальный код вида номенклатуры
	Наименование		Вид номенклатуры
Ставка НДС	Код ставки НДС	PK	Уникальный код ставки НДС
	Наименование		Наименование ставки НДС
	Ставка		Процент ставки НДС
Значения свойства номенклатуры	Код значения свойства номенклатуры	PK	Уникальный код значения свойства номенклатуры
	Значение		Значение свойства
	Код номенклатуры	FK	Уникальный код номенклатуры
	Код вида свойства номенклатуры	FK	Уникальный код вида свойства номенклатуры
Вид свойства номенклатуры	Код вида свойства номенклатуры	PK	Уникальный код вида свойства номенклатуры
	Наименование		Вид свойства номенклатуры
Контрагент	Код контрагента	PK	Уникальный код контрагента
	Наименование		Имя контрагента
	Код вида номенклатуры	FK	Уникальный код вида контрагента
Значения свойства контрагента	Код значения свойства контрагента	PK	Уникальный код значения свойства контрагента
	Значение		Значение свойства
	Код вида свойства контрагента	FK	Уникальный код вида свойства контрагента
	Код контрагента	FK	Уникальный код контрагента
Вид свойства контрагента	Код вида свойства номенклатуры	PK	Уникальный код вида свойства номенклатуры
	Наименование		Наименование свойства
Вид контрагента	Код вида контрагента	PK	Уникальный код вида контрагента
	Наименование		Вид контрагента
Контактное лицо контрагента	Код контактного лица контрагента	PK	Уникальный код контактного лица контрагента
	ФИО		Фамилия, имя и отчество
	Телефон		Телефон
	Email		Электронная почта
	Код роли контактного лица	FK	Уникальный код роли контактного лица

Продолжение таблицы 5

Сущность	Атрибут	Роль атрибута	Описание атрибута
Роль контактного лица	Код роли контактного лица	РК	Уникальный код роли контактного лица
	Наименование		Имя контактного лица
Статус заказа	Код статуса заказа	РК	Уникальный код статуса заказа
	Наименование		Название статуса заказа
Тип склада	Код типа склада	РК	Уникальный код типа склада
	Наименование		Название типа склада
Склад	Код склада	РК	Уникальный код склада
	Наименование		Название склада
	Код сотрудника	FK	Уникальный код сотрудника
	Код типа склада	FK	Уникальный код типа склада
Сотрудник	Код сотрудника	РК	Уникальный код сотрудника
	ФИО		Фамилия, имя и отчество сотрудника
	Дата рождения		Дата рождения сотрудника
	Телефон		Номер телефона сотрудника
	Адрес регистрации		Адрес регистрации сотрудника
	Адрес проживания		Адрес фактического проживания сотрудника
	ИНН		Идентификационный номер налогоплательщика
	СНИЛС		Страховой номер индивидуального лицевого счёта сотрудника
	Паспорт		Серия и номер паспорта сотрудника
	Кем выдан паспорт		Кем выдан паспорт сотрудника
	Дата выдачи паспорта		Дата выдачи паспорта сотрудника
	Email		Электронная почта сотрудника
	Код должности	FK	Уникальный код должности
Должность	Код должности	РК	Уникальный код должности
	Наименование		Название должности
Организация	Код организации	РК	Уникальный код организации
	Наименование		Название организации
	КПП		Код причины постановки на учет организации
	ОГРН		Основной государственный регистрационный номер
	ИНН		Идентификационный номер налогоплательщика
	Дата регистрации		Дата государственной регистрации
	Адрес		Адрес организации
	Телефон		Телефон организации
	Email		Электронный адрес почты организации

Продолжение таблицы 5

Сущность	Атрибут	Роль атрибута	Описание атрибута
Документ Поступление материалов	Код документа Поступление материалов	РК	Уникальный код документа Поступление материалов
	Дата		Дата создания документа
	Код организации	FK	Уникальный код организации
	Код склада	FK	Уникальный код склада
	Код контактного лица контрагента	FK	Уникальный код контактного лица контрагента
	Код контрагента	FK	Уникальный код контрагента
Перечень материалов	Код документа Поступление материалов	FK	Уникальный код документа Поступление материалов
	Код номенклатуры	FK	Уникальный код номенклатуры
	Количество		Количество материала
	Цена		Цена за единицу материала
	Сумма		Сумма
Документ Реализация материалов	Код документа Реализация материалов	РК	Уникальный код документа Реализация материалов
	Дата		Дата создания документа
	Код организации	FK	Уникальный код организации
	Код сотрудника	FK	Уникальный код сотрудника
	Код контрагента	FK	Уникальный код контрагента
	Код склада	FK	Уникальный код склада
Список материалов	Код документа Реализация материалов	FK	Уникальный код документа Реализация материалов
	Код номенклатуры	FK	Уникальный код номенклатуры
	Количество		Количество материала
	Цена		Цена за единицу материала
	Сумма		Сумма
	Код ставки НДС	FK	Уникальный код ставки НДС
	Сумма НДС		Сумма НДС
	Всего		Сумма с НДС
Документ Установка цен	Код документа Установка цен	РК	Уникальный код документа Установка цен
	Дата		Дата создания документа
	Код организации	FK	Уникальный код организации
Перечень номенклатуры	Код документа Установка цен	FK	Уникальный код документа Установка цен
	Код номенклатуры	FK	Уникальный код номенклатуры
	Цена		Цена за единицу
Документ Поступление техники	Код документа Поступление техники	РК	Уникальный код документа Поступление техники
	Дата		Дата создания документа
	Код организации	FK	Уникальный код организации
	Код контрагента	FK	Уникальный код контрагента
	Код склада	FK	Уникальный код склада
	Код сотрудника	FK	Уникальный код сотрудника

Окончание таблицы 5

Сущность	Атрибут	Роль атрибута	Описание атрибута
Перечень техники	Код документа Поступление техники	FK	Уникальный код документа Поступление техники
	Заявленная неисправность		Заявленная неисправность техники со слов клиента
	Комментарии		Дополнительные комментарии к заказу
	Код номенклатуры	FK	Уникальный код номенклатуры
	Код статуса заказа	FK	Уникальный код статуса заказа
Документ Передача техники мастеру	Код документа Передача техники	PK	Уникальный код документа Передача техники
	Дата		Дата создания заказа
	Код документа Поступление техники	FK	Уникальный код документа Поступление техники
	Код сотрудника	FK	Уникальный код сотрудника
Список техники на ремонт мастеру	Код документа Передача техники	FK	Уникальный код документа Передача техники
	Код номенклатуры	FK	Уникальный код номенклатуры
	Неисправность		Заявленная неисправность техники со слов клиента
	Код статуса заказа	FK	Уникальный код статуса заказа
	Комментарии		Дополнительные комментарии к заказу
	Результат		Промежуточный результат ремонта техники
Документ Возврат техники контрагенту	Код документа Возврат техники контрагенту	PK	Уникальный код документа Возврат техники контрагенту
	Дата		Дата создания заказа
	Код документа Передача техники мастеру	FK	Уникальный код документа Передача техники мастеру
Перечень техники на возврат	Код документа Возврат техники контрагенту	FK	Уникальный код документа Возврат техники контрагенту
	Код номенклатуры	FK	Уникальный код номенклатуры
	Код статуса заказа	FK	Уникальный код статуса заказа
	Неисправность		Заявленная неисправность техники со слов клиента
	Комментарии		Дополнительные комментарии к заказу
	Результат		Конечный результат ремонта техники
	Код документа Реализация материалов	FK	Уникальный код документа Реализация материалов
	Стоимость		Стоимость ремонта

После построения логической модели данных необходимо провести ее нормализацию, то есть осуществить организацию данных таким образом, чтобы обеспечить защиту данных и сделать базу данных более гибкой, без

избыточности и несогласованности. Цели нормализации заключаются в исключении дублирования информации в таблицах, обеспечении возможности изменений в структуре таблиц и уменьшении влияния структурных изменений базы данных на работу программы.

Так как сущности спроектированной логической модели базы данных содержат простые атрибуты, а также каждый не ключевой атрибут зависит от первичного ключа и не зависит от всех остальных атрибутов, то следует вывод, что данная модель является нормализованной и находится в третьей нормальной форме.

3.3 Проектирование программного обеспечения

3.3.1 Анализ прецедентов

Для определения функциональных требований к информационной системе управления сервисным центром была разработана диаграмма прецедентов.

Диаграмма прецедентов или диаграмма вариантов использования – это графическое представление функциональности системы, которое описывает взаимодействие пользователей с системой и ее компонентами. Диаграмма прецедентов состоит из актеров (пользователи системы) и прецедентов (действия, которые могут быть выполнены в системе). Актеры представляют роли, которые пользователи могут играть в системе, а прецеденты описывают, как эти роли взаимодействуют с системой.

Каждый прецедент имеет свое название и описание, которые описывают его функциональность. Кроме того, для каждого прецедента указываются возможные варианты использования, которые могут быть выполнены пользователем.

На диаграмме прецедентов также могут быть указаны связи между прецедентами, которые показывают, как один прецедент может вызывать другой. Эти связи могут быть представлены в виде стрелок или линий [11].

Таким образом, диаграмма прецедентов помогает лучше понимать, как система будет использоваться и как должна быть разработана для

удовлетворения потребностей пользователей. Диаграмма прецедентов представлена на рисунке 11.

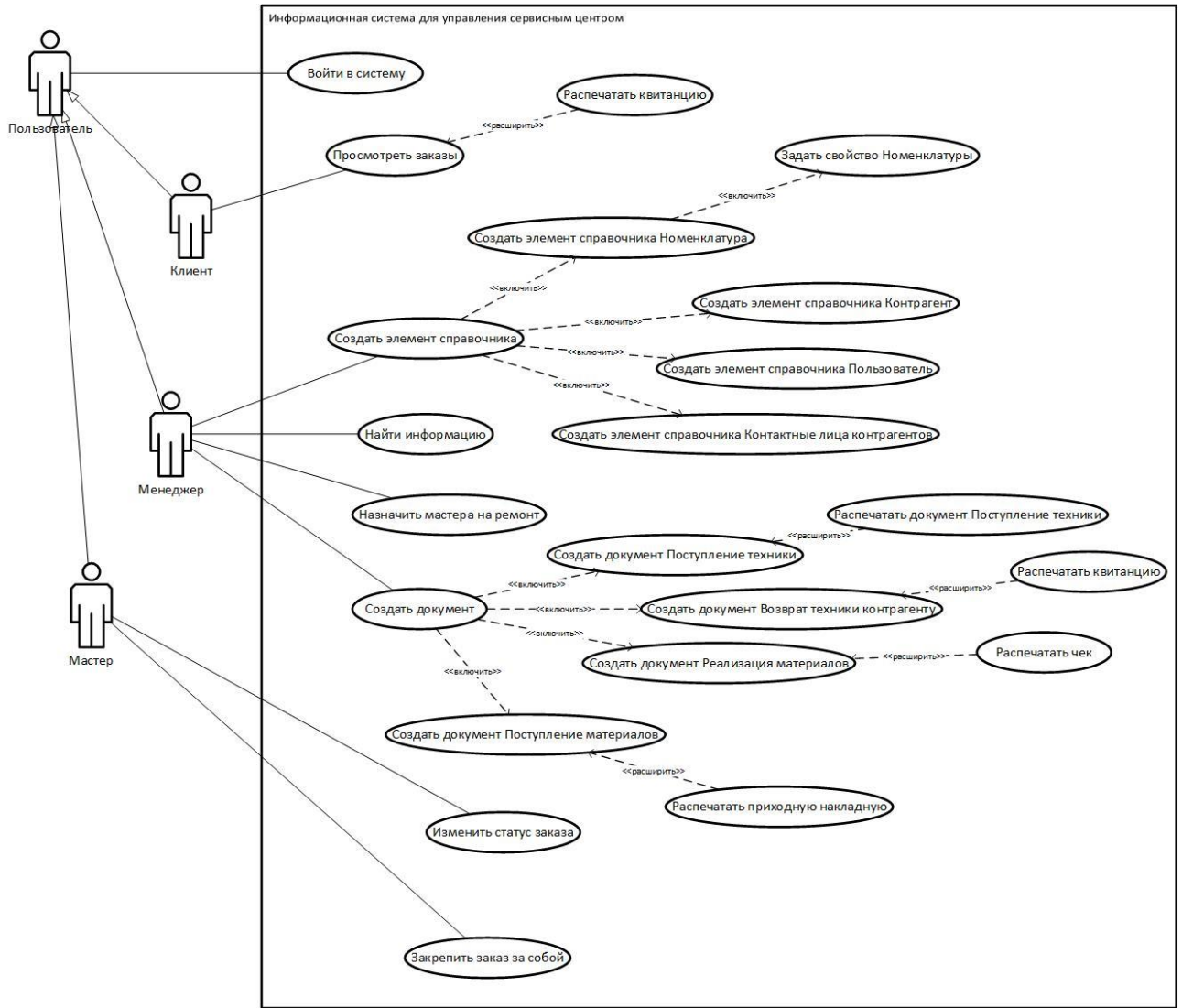


Рисунок 11 – Диаграмма прецедентов

Описание вариантов использования приведено в таблице 6.

Таблица 6 – Описание вариантов использования

Действующее лицо	Вариант использования	Описание
Пользователь	Войти в систему	Пользователь инициирует вход в систему
Клиент	Просмотреть заказы	Клиент может просмотреть информацию о своих заказах
	Распечатать квитанцию	Клиент может распечатать квитанцию
Мастер	Изменить статус заказа	Мастер при необходимости может менять статус ремонта техники

Продолжение таблицы 6

Действующее лицо	Вариант использования	Описание
	Закрепить заказ за собой	Мастер может закреплять за собой технику, которая не передана другому мастеру
Менеджер	Создать элемент справочника	Менеджер может создавать справочники
	Создать элемент справочника Номенклатура	Менеджер может создать элемент справочника, содержащий информацию о номенклатуре
	Создать элемент справочника Контрагент	Менеджер может создать элемент справочника, содержащий информацию о контрагенте
	Создать элемент справочника Пользователь	Менеджер может создать элемент справочника, содержащий информацию о пользователе
	Создать элемент справочника Контактное лицо контрагента	Менеджер может создать элемент справочника, содержащий информацию о контактном лице контрагента
	Задать свойство номенклатуры	Менеджер может указывать свойства номенклатуры
	Найти информацию	Менеджер может найти информацию в системе о справочниках и документах
	Назначить мастера на ремонт	Менеджер имеет возможность назначать мастеров на ремонт техники
	Создать документ	Менеджер может создавать документы
	Создать документ Поступление техники	Менеджер фиксирует операцию поступления техники от контрагента
	Создать документ Реализация материалов	Менеджер фиксирует операцию расхода материалов
	Создать документ Установка цен	Менеджер устанавливает цены на материалы и услуги
	Создать документ Возврат техники контрагенту	Когда техника готова к выдаче, менеджер может создать документ, который фиксирует операцию возврата техники контрагенту
	Создать документ Поступление материалов	Менеджер фиксирует операцию поступления материалов на склад организации
	Распечатать приходную накладную	Менеджер имеет возможность создать печатную форму приходной накладной и распечатать ее
	Распечатать квитанцию	Менеджер имеет возможность создать печатную форму квитанции о возврате техники клиенту и распечатать ее
	Распечатать чек	Менеджер имеет возможность создать печатную форму реализации материалов и распечатать ее

В результате проведенного анализа были определены и описаны

действующие лица системы и прецеденты, а также построена диаграмма вариантов использования.

3.3.2 Диаграмма состояний

Диаграмма состояний – это графическое представление модели поведения системы, которое описывает возможные состояния объектов системы и переходы между ними. Диаграмма состояний состоит из состояний, переходов и событий.

Диаграмма состояний используется для моделирования поведения системы и определения ее требований. Она показывает, как объекты системы должны вести себя в различных ситуациях и как они могут быть связаны друг с другом [12].

Для отображения смены состояний объекта Реализация материалов, была разработана диаграмма состояний, представленная на рисунке 12.

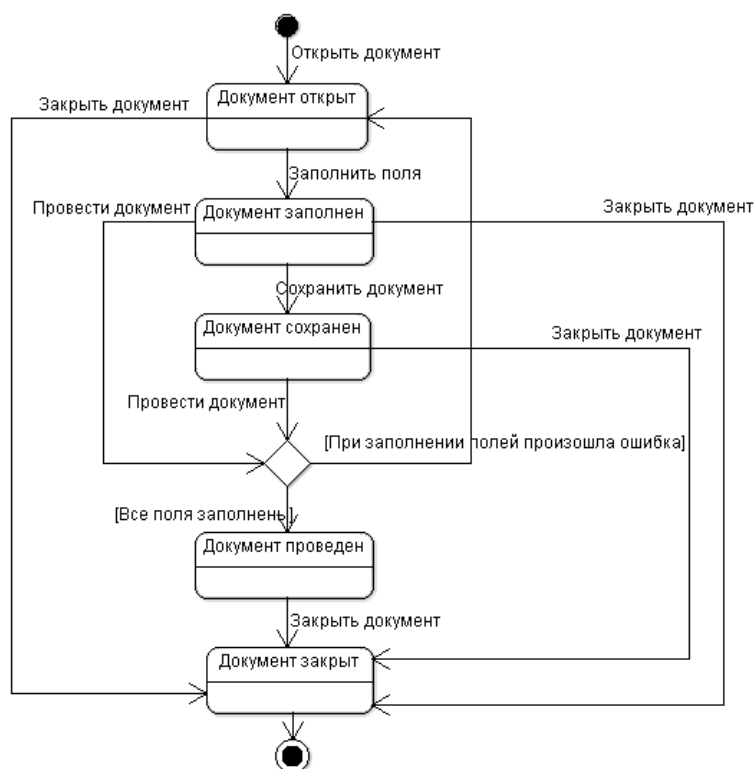


Рисунок 12 – Диаграмма состояний

Спецификации переходов объекта из одного состояния в другое определены в таблице 7.

Таблица 7 – Спецификации переходов объекта из состояния в состояние

Состояние 1	Событие	Состояние 2	Описание перехода
Начальное состояние	Открыть документ	Документ открыт	При создании документ переходит из начального состояния в состояние Документ открыт
Документ открыт	Заполнить поля	Документ заполнен	При заполнении полей документ переходит из состояния Документ открыт в состояние Документ заполнен
	Заккрыть документ	Документ закрыт	При закрытии документ переходит из состояния Документ открыт в состояние Документ закрыт
Документ заполнен	Сохранить документ	Документ сохранен	При сохранении документ переходит из состояния Документ заполнен в состояние Документ сохранен
	Провести документ	Документ проведен	Если все обязательные поля заполнены, то при проведении документ переходит из состояния Документ заполнен в состояние Документ проведен
	Провести документ	Документ открыт	Если при заполнении полей произошла ошибка, то документ переходит из состояния Документ заполнен в состояние Документ открыт
	Заккрыть документ	Документ закрыт	При закрытии документ переходит из состояния Документ заполнен в состояние Документ закрыт
Документ сохранен	Заккрыть документ	Документ закрыт	При закрытии документ переходит из состояния Документ сохранен в состояние Документ закрыт
	Провести документ	Документ проведен	Если все обязательные поля заполнены, то при проведении документ переходит из состояния Документ сохранен в состояние Документ проведен
	Провести документ	Документ открыт	Если при заполнении полей произошла ошибка, то документ переходит из состояния Документ сохранен в состояние Документ открыт
Документ проведен	Заккрыть документ	Документ закрыт	При закрытии документ переходит из состояния Документ проведен в состояние Документ закрыт

Таким образом, была разработана диаграмма состояний, которая показала возможные состояния объекта Реализация материалов.

3.3.3 Диаграмма деятельности

Диаграмма деятельности – это графическое представление последовательности действий или процесса в системе. Она состоит из узлов и

связей, которые показывают порядок выполнения задач и переходы между ними [13].

Для разработки диаграммы деятельности был выбран вариант использования «Создать документ Реализация материалов». Диаграмма деятельности в нотации UML приведена на рисунке 13.

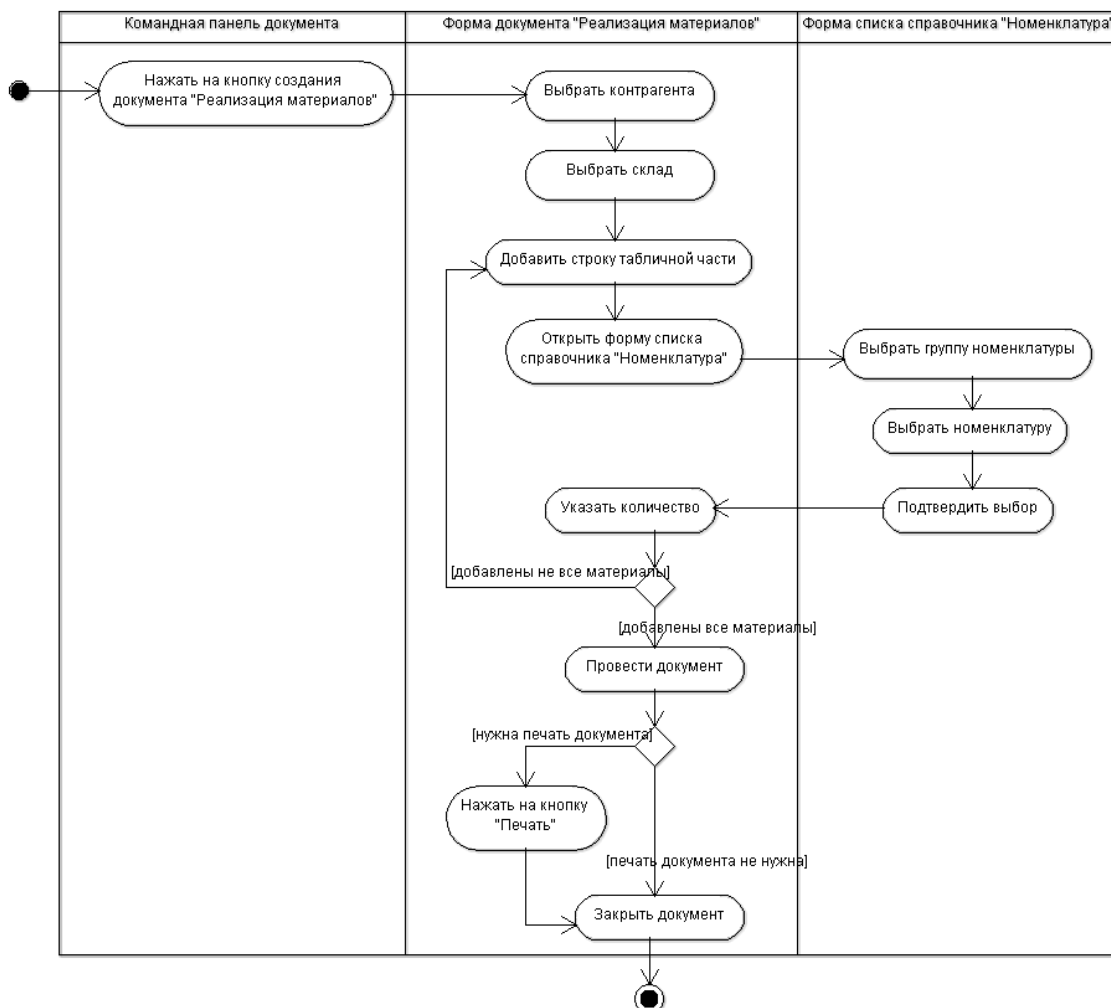


Рисунок 13 – Диаграмма деятельности

Рассмотрим приведенную диаграмму деятельности. Для создания документа Реализация материалов необходимо нажать на кнопку создания документа. После открытия формы документа нужно выбрать контрагента и склад. Далее необходимо добавить пустую строку в табличную часть документа. Затем пользователь выбирает номенклатуру, открыв из поля ввода форму списка справочника номенклатуры. После выбора номенклатуры необходимо подтвердить выбор и ввести количество. Также при необходимости можно

создать еще одну позицию в табличной части. После того, как все позиции будут добавлены необходимо провести и закрыть документ. Таким образом, в системе создается новый документ Реализация материалов.

3.3.4 Диаграмма последовательности

Диаграмма последовательности является типом UML-диаграммы, которая описывает взаимодействие объектов в системе в течение определенного времени. Диаграмма последовательности используется для отображения последовательности сообщений, которые пересылаются между объектами или компонентами в рамках определенной функциональности.

На диаграмме последовательности объекты представляются вертикальными линиями, а сообщения – стрелками, которые указывают направление передачи информации между объектами. Каждое сообщение имеет свое название и может быть дополнительно описано в текстовом блоке.

Временные ограничения на диаграмме последовательности показывают, когда и как долго происходит взаимодействие между объектами. Они могут быть представлены в виде вертикальных линий или горизонтальных прямоугольников.

Диаграмма последовательности помогает понять, как работает система во времени, особенно с точки зрения взаимодействия между объектами. Диаграмма может использоваться как для моделирования текущей архитектуры, так и для планирования будущих изменений в системе [14].

Для разработки диаграммы последовательности был выбран вариант использования «Добавить документ Возврат техники контрагенту». В создании документа участвует один актер (менеджер) и 4 элемента системы: документ Возврат техники контрагенту, документ Поступление техники от контрагента, документ Реализация материалов, регистр накопления.

Разработанная диаграмма последовательности представлена на рисунке 14.

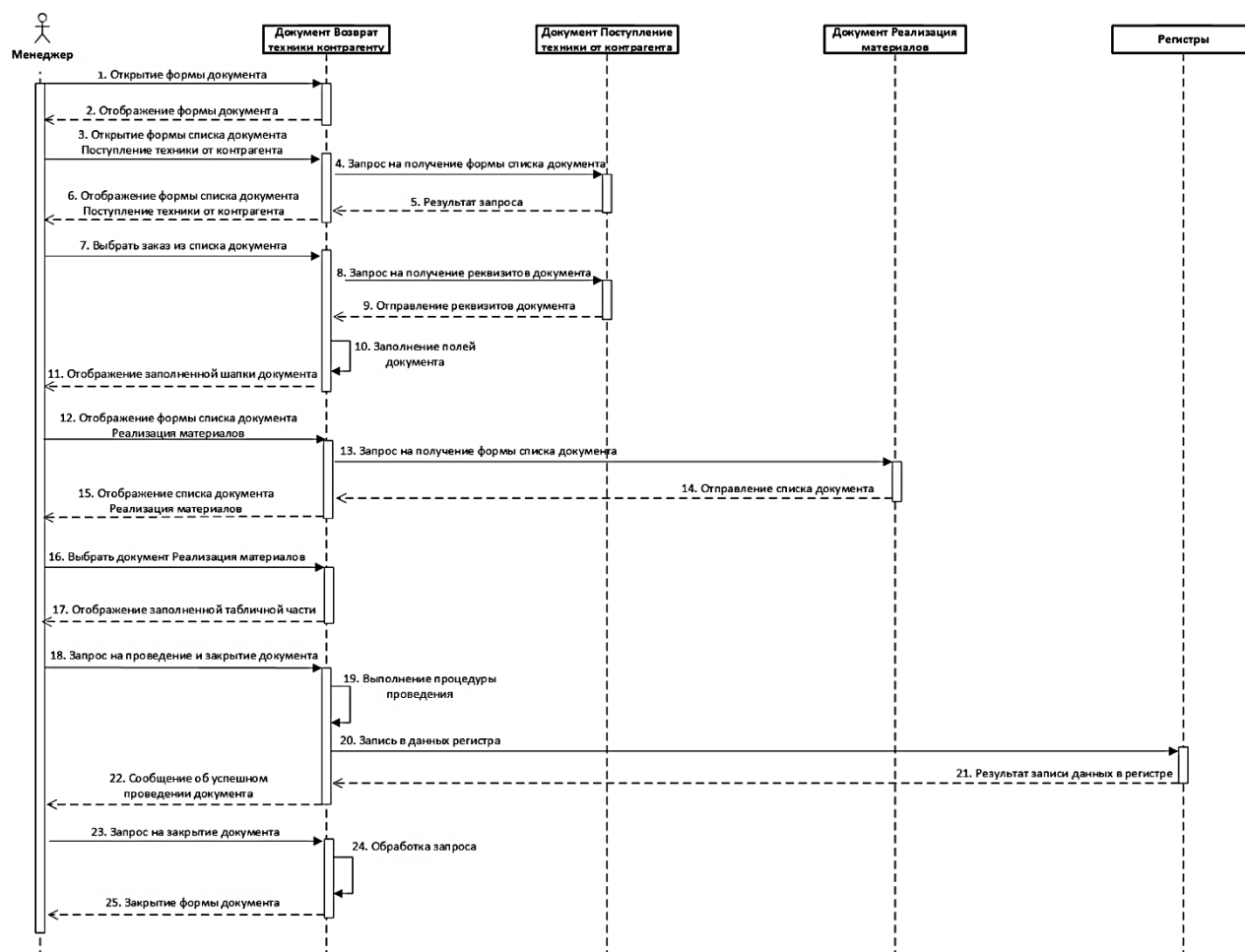


Рисунок 14 – Диаграмма последовательности

Для оформления возврата техники клиенту менеджер посылает сообщение документу Возврат техники контрагенту об открытии формы. После отображается форма документа. Далее менеджер выбирает нужную заявку (ссылку документа) из выпадающего списка и через форму документа Возврат техники контрагенту менеджер открывает форму списка документа Поступление техники от контрагента. Менеджер выбирает номер заказа, далее реквизиты документа Возврат техники контрагенту заполняются автоматически данными из заказа.

Затем нужно указать ссылку на документ Реализация материалов, для этого менеджер посылает сообщение документу об открытии формы списка. Далее менеджер выбирает нужный документ и отправляет запрос на проведение документа. После на стороне формы документа Возврат техники контрагента

выполняется процедура проведения и создается новая запись в регистре накоплений (вид движения – расход). Затем менеджер отправляет запрос на закрытие документа, запрос обрабатывается и документ закрывается.

Таким образом, была построена диаграмма последовательности в нотации UML.

3.3.5 Диаграмма кооперации

Диаграмма кооперации – это диаграмма UML, которая показывает взаимодействие между объектами в системе в процессе выполнения задачи или действия. В этой диаграмме акцент сделан на том, как объекты взаимодействуют друг с другом для достижения задачи [15].

Для разработки диаграммы кооперации был выбран вариант использования «Добавить документ Поступление техники от контрагента». При проектировании были выделены следующие объекты: компоненты приложения React, хранилище состояний MobX, web-сервер Apache и 1С:Предприятие 8.3.

Построенная диаграмма кооперации приведена на рисунке 15.

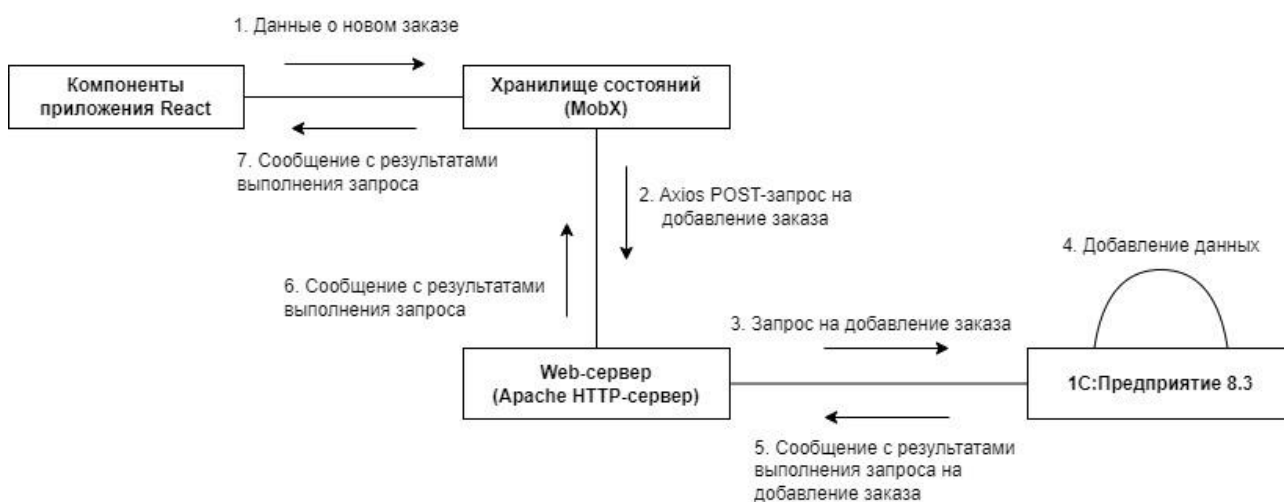


Рисунок 15 – Диаграмма кооперации

Диаграмма описывает процесс добавления заказа в базу данных 1С:Предприятия. После добавления нового заказа, компонент приложения через хранилище состояний с помощью Axios POST-запроса отправляет сообщение с информацией о создаваемом заказе на Web-сервер. Затем сообщение

обрабатывается и формируется запрос к БД на создание заказа. 1С:Предприятие сохраняет данные о заказе и формирует ответ, который приходит на Web-сервер. После формируется сообщение с результатами добавления новой записи в базу данных и через хранилище состояний отправляется в компонент приложения.

4 Реализация информационной системы

4.1 Реализация структур программного средства

Работа в конфигураторе платформы 1С:Предприятие 8.3 начинается с создания объектов конфигурации – средств, предназначенных для отражения реальных объектов и явлений предметной области. В ходе разработки информационной системы управления сервисным центром были использованы следующие объекты конфигурации: справочники, документы, перечисления, отчеты, регистры накопления, регистры сведений, общие формы, общие макеты, HTTP-сервисы, константы, планы видов характеристик, обработки.

С использованием UML-средств можно построить диаграмму компонентов, которая позволяет проиллюстрировать все компоненты, используемые в разрабатываемой информационной системе.

Диаграмма компонентов является высокоуровневой диаграммой, которая отображает архитектуру системы и ее компоненты. Она помогает понять, как компоненты соединены между собой и как они взаимодействуют друг с другом.

Разработанная диаграмма компонентов представлена в приложении Б.

Рассмотрим подробнее структуру и особенности основных объектов конфигурации.

Справочники. Справочники позволяют хранить в информационной базе данные, имеющие одинаковую структуру и списочный характер. Всего в разрабатываемой информационной системе представлено 13 справочников.

1. Справочник «Номенклатура» предназначен для хранения информации о бытовой технике, принятой у клиента для ремонта; об услугах, которые предоставляет организация; о материалах, которые используются для ремонта. Содержит стандартные реквизиты «Наименование», «Код» и «Группа». Имеет дополнительные реквизиты «Вид номенклатуры» и «Основная ставка НДС». Форма справочника «Номенклатура» представлена на рисунке 16.

Рисунок 16 – Форма справочника «Номенклатура»

2. Справочник «Значения характеристик номенклатуры» позволяет управлять списком характеристик, определять их значения, связывать с конкретным материалом, услугой или техникой. Форма справочника приведена на рисунке 17.

Рисунок 17 – Форма справочника «Значения характеристик номенклатуры»

3. Справочник «Контрагенты» предназначен для хранения и управления информацией о взаимоотношениях организации с ее контрагентами. Контрагентами могут быть клиенты, поставщики и другие. Форма справочника представлена на рисунке 18.

Рисунок 18 – Форма справочника «Контрагенты»

4. Справочник «Значения характеристик контрагента» предназначен для хранения информации о характеристиках, которые могут быть присвоены каждому конкретному контрагенту. Он позволяет управлять списком характеристик, определять их значения, связывать с конкретными контрагентами. Форма справочника представлена на рисунке 19.

Рисунок 19 – Форма справочника «Значения характеристик контрагента»

5. Справочник «Контактные лица контрагентов» предназначен для хранения информации о контактных лицах, которые связаны с определенным контрагентом. Он позволяет установить связь между контрагентом и его контактными лицами, а также определить их роли в рамках работы с данной организацией. Это помогает улучшить коммуникацию с партнерами, а также более эффективно организовать работу с контактами. Форма справочника «Контактные лица контрагентов» представлена на рисунке 20.

Рисунок 20 – Форма справочника «Контактные лица контрагентов»

6. Справочник «Роли контактных лиц» предназначен для определения роли контактного лица в рамках процессов взаимодействия с контрагентами. Форма справочника приведена на рисунке 21.

The screenshot shows a web application window titled '☆ Ответственный по закупкам (Роли конт...'. It contains two buttons at the top: 'Записать и закрыть' (Record and Close) and 'Записать' (Record), along with a dropdown menu 'Еще ▾'. Below these are two input fields: 'Код:' with the value '000000003' and 'Наименование:' with the value 'Ответственный по закупкам'.

Рисунок 21 – Форма справочника «Роли контактных лиц»

7. Справочник «Сотрудники» предназначен для хранения и управления информацией о сотрудниках организации. Форма справочника представлена на рисунке 22.

The screenshot shows a web application window titled '☆ Васильев Антон Владимирович (Список сотрудников)'. It contains two buttons at the top: 'Записать и закрыть' (Record and Close) and 'Записать' (Record), along with a dropdown menu 'Еще ▾'. Below these are several input fields for employee data: 'Код:' (000000002), 'Ф.И.О.:' (Васильев Антон Владимирович), 'Группа сотрудника:' (Мастера), 'Должность:' (Мастер), 'Дата рождения:' (18.01.1985), 'Телефон мобильный:' (+7 (987) 512 45 72), 'Адрес регистрации:' (440066, г. Пенза, ул. Пушкина, д. 158, кв. 2), 'Адрес проживания:' (440066, г. Пенза, ул. Пушкина, д. 158, кв. 2), 'Оклад:' (52 000), 'ИНН:' (7743013902), 'СНИЛС:' (258-424-114 25), 'Паспорт:' (5812 78545), 'Кем выдан паспорт:' (УМВД по Пензенской области), 'Дата выдачи паспорта:' (02.02.2007), and 'Email:' (vasilev@mail.ru).

Рисунок 22 – Форма справочника «Сотрудники»

8. Справочник «Должности» предназначен для хранения информации о должностях сотрудников организации. Он содержит информацию о названии должности и ее коде. Форма справочника представлена на рисунке 23.

Рисунок 23 – Форма справочника «Должности»

9. Справочник «Пользователи» предназначен для ведения списков пользователей системы для авторизации через веб-интерфейс, содержит реквизиты «Почта», «ФИО», «Ссылка на пользователя», «Роль» и «Пользователь заблокирован». На форме представлена кнопка, которая позволяет отправить приглашение на электронную почту пользователя с ссылкой на установку пароля. Форма справочника приведена на рисунке 24.

Рисунок 24 – Форма справочника «Пользователи»

10. Справочник «Учетные записи электронной почты» предназначен для хранения информации о почтовых ящиках, которые используются в рамках работы предприятия. Он позволяет управлять списком учетных записей, определять параметры подключения и использовать при отправке электронной почты из системы 1С. С помощью этого справочника настроена отправка уведомлений о заказах и отправка ссылки на установку пароля для входа в систему через веб-интерфейс. Форма справочника приведена на рисунке 25.

Рисунок 25 – Форма справочника «Учетные записи электронной почты»

11. Справочник «Склады» предназначен для хранения списка складов. С помощью этого справочника возможно организовать учет движения товаров на складах предприятия, контролировать выполнение поставок, анализировать запасы материалов. Форма справочника представлена на рисунке 26.

Рисунок 26 – Форма справочника «Склады»

12. Справочник «Ставки НДС» предназначен для хранения информации о различных ставках налога на добавленную стоимость, которые используются в рамках работы предприятия. Он позволяет управлять списком ставок, определять их значения. С помощью этого справочника автоматически рассчитывается сумма налога на добавленную стоимость при продаже материалов и услуг. Форма справочника представлена на рисунке 27.

Рисунок 27 – Форма справочника «Ставки НДС»

13. Справочник «Организации» предназначен для хранения и управления информацией об организации, к которой относится учетная база. Форма справочника приведена на рисунке 28.

Рисунок 28 – Форма справочника «Организации»

Документы. Документы используются для описания различных операций, происходящих в предприятии. Они служат для фиксации факта произведенных операций, отслеживания текущей деятельности, а также позволяют проводить учет и анализ хозяйственной деятельности. Для системы управления сервисным центром были созданы 6 документов.

1. Документ «Поступление техники от контрагента» фиксирует факт поступления техники для ремонта. Документ предоставляет информацию об ответственном лице, принявшем технику, о контрагенте и его контактном лице, в нем содержатся список принятой техники с указанием количества техники и описанием неисправности со слов клиента. Документ выполняет движения по регистру накопления «Остатки техники на складе». Форма документа «Поступление техники от контрагента» представлена на рисунке 29.

N	Устройство	Заявленная неисправность	Статус	Комментарии
1	Холодильник многодверный ...	Холодильник включается и сразу выключается	Принят в ремонт	Состав: холодильник, документация

Рисунок 29 – Форма документа «Поступление техники от контрагента»

Для документа «Поступление техники от контрагента» создана печатная форма, которая представлена на рисунке 30.

Квитанция о приеме в ремонт № 000000007 от 12 июня 2023 г. в 11:42

Организация

ОАО "Сервис"

Заказчик

Белякова Кристина Дмитриевна

№	Устройство	Заявленная неисправность
1	Холодильник многодверный Samsung RF61K90407F	Холодильник включается и сразу выключается

Оборудование с согласия клиента принято без разборки и проверки неисправностей, без проверки внутренних повреждений. Клиент согласен, что все неисправности, внутренние повреждения, которые могут быть обнаружены в оборудовании при техническом обслуживании возникли до приема оборудования по данной расписке. При выполнении ремонта все настройки могут быть возвращены к заводским настройкам. Срок хранения техники 2 месяца, после чего техника утилизируется.

Оборудование в ремонт принял:

Оборудование в ремонт сдал:

Рисунок 30 – Печатная форма документа

2. Документ «Передача техники мастеру» предназначен для отражения передачи техники от кладовщика мастеру для выполнения работ по ремонту. В документе указываются данные о передаваемой технике, ее состоянии, а также информация о мастере, которому она передается. Документ позволяет контролировать движение техники в организации. Форма документа «Передача техники мастеру» представлена на рисунке 31.

🏠 ⬅ ➡ ☆

Передача техники мастеру 000000004 от 13.06.2023 18:35:50 *

🔗 ⓘ ✕

Провести и закрыть

Записать

Провести

Печать

Создать на основании ▾

Еще ▾

Заявка:

Поступление техники от контрагента ▾ ⓘ

Дата заказа:

13.06.2023 17:32:09

Менеджер:

Иванов Иван Иванович ⓘ

Мастер:

Васильев Антон Владимирович ▾ ⓘ

Клиент:

Белякова Кристина Дмитриевна ⓘ

Склад:

Основной ▾ ⓘ

Кладовщик:

Ларин Сергей Владимирович ⓘ

Техника:

Добавить

⬆ ⬇ ⬆

Поиск (Ctrl+F)

✕

Еще ▾

N	Номенклатура	Заявленная неисправность	Статус	Комментарии	Ре
1	Холодильник многодверный ...	Громко шумит	Согласование с клиентом	Состав: холодильник, документация. В ходе диагностики было принято решение заменить ...	Ну ор

Рисунок 31 – Форма документа «Передача техники мастеру»

3. Документ «Возврат техники контрагенту» предназначен для отражения возврата техники контрагенту после выполнения работ по ремонту или обслуживанию. В документе указываются данные о возвращаемой технике,

ее состоянии после ремонта, дате заказа, стоимости ремонта, а также информация о контрагенте, которому она возвращается. Форма документа «Возврат техники контрагенту» представлена на рисунке 32.

Возврат техники контрагенту 000000002 от 13.06.2023 18:43:21 *

Провести и закрыть Записать Провести Печать

Еще ▾

Заявка: Поступление техники от контрагента 000000002 ▾ Дата заказа: 12.06.2023 23:32:14

Менеджер: Акимова Анастасия Владимировна ▾

Мастер: Васильев Антон Владимирович ▾

Клиент: Белякова Кристина Дмитриевна ▾

Склад: Основной ▾

Кладовщик: Ларин Сергей Владимирович ▾

Техника:

Добавить ↑ ↓ Поиск (Ctrl+F) × Еще ▾

N	Номенклатура	Заявленная неисправность	Статус	Комментарии	Результат	Договор на оплату	Стоимость
1	Холодильник LG DoorCooling+...	Холодильник слишком морозит	Получен	Состояние холодильника на момент приема: царапины на металлической обшивке	Заменен датчик температурный морозильной камеры	Реализация материалов 000000002 от ...	

Рисунок 32 – Форма документа «Возврат техники контрагенту»

Для документа «Возврат техники контрагенту» создана печатная форма, которая представлена на рисунке 33.

Квитанция о выдаче оборудования № 000000002 от 13 июня 2023 г. в 09:43

Организация **ОАО "Сервис"**

Менеджер **Акимова Анастасия Владимировна**

Заказчик **Белякова Кристина Дмитриевна**

№	Номенклатура	Результат
1	Холодильник LG DoorCooling+ GA-B509CCIL	Заменен датчик температурный морозильной камеры

Оборудование получил исправным, претензий не имею:

" " " г.

подпись клиента

расшифровка подписи

Рисунок 33 – Печатная форма документа «Возврат техники контрагенту»

4. Документ «Поступление материалов» предназначен для отражения поступления материалов на склад организации. В документе указываются данные о поставщике, складе, количестве и стоимости поступивших материалов. Сумма рассчитывается автоматически на основе введенных данных. Документ позволяет контролировать остатки материалов на складе. Форма документа приведена на рисунке 34.

Поступление материалов 000000001 от 01.06.2022 12:00:00

Провести и закрыть Записать Провести Печать

Дата: 01.06.2022 12:00:00

Склад: Основной

Кладовщик: Ларин Сергей Владимирович

Поставщик: САМСУНГ

Контактное лицо поставщика: Михайлова Марина Алексеевна

Перечень материалов:

Добавить Поиск (Ctrl+F) Еще

N	Материал	Количество	Цена	Сумма
1	Амортизаторы для стиральной машины Samsung WW65K52E...	10,000	540,00	5 400,00
2	Тэн (нагревательный элемент) для стиральной машины Sams...	5,000	2 700,00	13 500,00
3	Датчик температурный морозильной камеры холодильника Sa...	10,000	2 960,00	29 600,00
4	Опорно цепной двигатель в сборе Samsung RF61K90407F	10,000	4 110,00	41 100,00

Рисунок 34 – Форма документа «Поступление материалов»

Для документа «Поступление материалов» создана печатная форма, которая представлена на рисунке 35.

Приходная накладная № 000000003 от 20 декабря 2022 г.

Поставщик: ЛЕНОВО

Контактное лицо поставщика: Калинин Андрей Васильевич

Покупатель: ОАО "Сервис"

№	Материал	Количество	Цена	Сумма
1	Опорно цепной двигатель в сборе Samsung RF61K90407F	3,000	4 110,00	12 330,00
2	Насос (помпа) для стиральной машины LG AIDD F4V5VS0W	5,000	680,00	3 400,00
Итого:				15 730,00

Всего наименований 2, на сумму 15 730,00 руб.
Пятнадцать тысяч семьсот тридцать рублей 00 копеек

Отпустил _____ Получил _____

Рисунок 35 – Печатная форма документа «Поступление материалов»

5. Документ «Установка цен» предназначен для установки цен на материалы и услуги, которые организация продает. Форма документа представлена на рисунке 36.

Установка цен 000000001 от 10.05.2022 0:00:00

Провести и закрыть Записать Провести Печать

Дата: 10.05.2022 0:00:00

Организация: ОАО "Сервис"

Список цен на номенклатуру:

Добавить Поиск (Ctrl+F) Еще

N	Номенклатура	Цена
1	Насос (помпа) для стиральной машины LG AIDD F4V5VS0W	680,00
2	Тэн (нагревательный элемент) для стиральной машины LG AIDD F4V5VS0W	2 850,00
3	Амортизаторы для стиральной машины Samsung WW65K52E69W	540,00
4	Тэн (нагревательный элемент) для стиральной машины Samsung WW65K52E69W	2 700,00

Рисунок 36 – Форма документа «Установка цен»

6. Документ «Реализация материалов» предназначен для отражения продажи материалов со склада организации. В документе указываются данные о покупателе, количестве и стоимости проданных материалов. Форма документа представлена на рисунке 37.

Реализация материалов 000000006 от 22.06.2023 1:16:11

Провести и закрыть Записать Провести Печать

Дата: 22.06.2023 1:16:11

Менеджер: Иванов Иван Иванович

Контрагент: Агапов Алексей Владимирович

Склад: Основной

Кладовщик: Ларин Сергей Владимирович

Перечень номенклатуры:

N	Номенклатура	Количество	Цена	Сумма	Став...	Сумма НДС	Всего
1	Датчик температурный морозильно...	1,000	4 440,00	4 440,00	10%	444,00	4 884,00
2	Замена датчика температурного мо...	1,000	2 600,00	2 600,00	20%	520,00	3 120,00

Рисунок 37 – Форма справочника «Реализация материалов»

Для документа «Реализация материалов» создана печатная форма, которая представлена на рисунке 38.

ОАО "Сервис", ИНН 7768437658
(наименование организации, ИНН)

440025, г. Пенза, ул. Московская, д. 75 корп. 1
(адрес расчета)

Товарный чек № 000000006 от 13 июня 2023 г. в 11:26

№	Номенклатура	Количество	Цена	Сумма НДС	Сумма
1	Датчик температурный морозильной камеры холодильника Samsung RF61K90407F	1,000	4 440,00	444,00	4 884,00
2	Замена датчика температурного морозильной камеры холодильника	1,000	2 600,00	520,00	3 120,00
Итого:					8 004,00
В том числе НДС:					964,00

Всего отпущено наименований 2, на сумму 8 004,00 руб.
Восемь тысяч четыре рубля 00 копеек

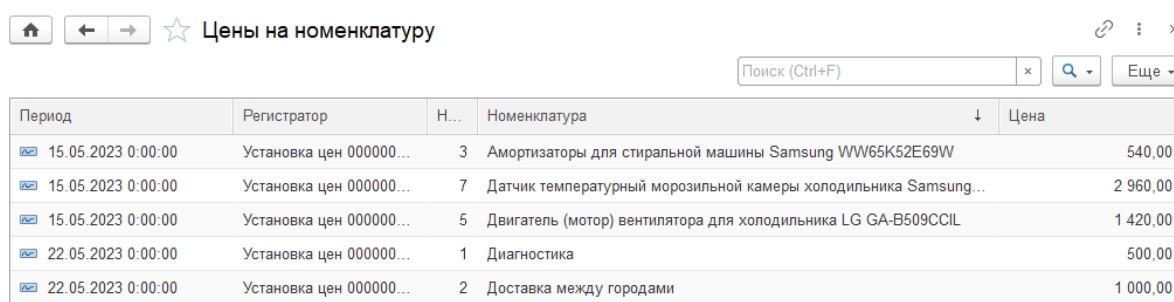
Менеджер Иванов Иван Иванович Подпись: _____

Рисунок 38 – Печатная форма документа «Реализация материалов»

Регистры сведений. Регистры сведений предназначены для хранения различных данных в разрезе измерений. В разрабатываемой системе представлено 6 регистров сведений.

1. Регистр сведений «Цены на номенклатуру» предназначен для хранения и управления информацией о ценах на материалы и услуги, которые

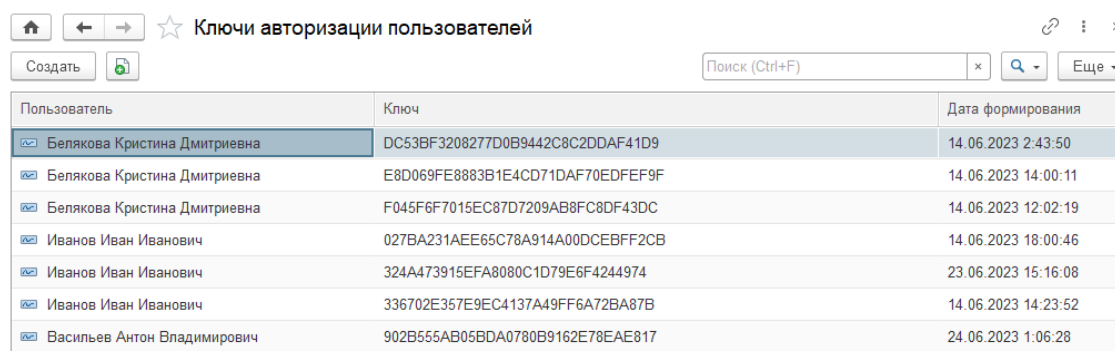
организация продает. В регистре сведений «Цены на номенклатуру» могут быть записаны цены на различные периоды времени, поэтому регистр позволяет быстро получать информацию о текущих и исторических ценах на номенклатуру. Данный регистр имеет периодичность по позиции регистратора. Регистратором для регистра «Цены на номенклатуру» является документ «Установка цен». Форма списка рассматриваемого регистра сведений представлена на рисунке 39.



Период	Регистратор	Н...	Номенклатура	Цена
15.05.2023 0:00:00	Установка цен 000000...	3	Амортизаторы для стиральной машины Samsung WW65K52E69W	540,00
15.05.2023 0:00:00	Установка цен 000000...	7	Датчик температурный морозильной камеры холодильника Samsung...	2 960,00
15.05.2023 0:00:00	Установка цен 000000...	5	Двигатель (мотор) вентилятора для холодильника LG GA-B509CCIL	1 420,00
22.05.2023 0:00:00	Установка цен 000000...	1	Диагностика	500,00
22.05.2023 0:00:00	Установка цен 000000...	2	Доставка между городами	1 000,00

Рисунок 39 – Форма списка регистра сведений «Цены на номенклатуру»

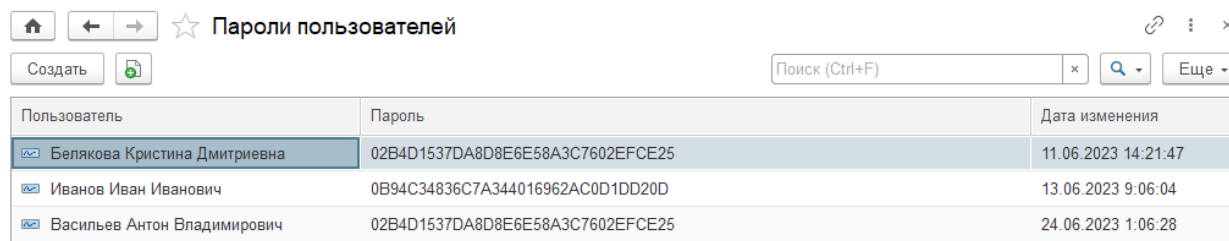
2. Регистр сведений «Ключи авторизации пользователей» предназначен для хранения и управления информацией о ключах авторизации пользователей системы. Ключи авторизации используются для защиты от несанкционированного доступа к информации и функционалу системы. Регистр позволяет быстро получать информацию о зарегистрированных ключах авторизации, а также использовать эту информацию для контроля доступа пользователей в системе. Форма списка рассматриваемого регистра сведений представлена на рисунке 40.



Пользователь	Ключ	Дата формирования
Белякова Кристина Дмитриевна	DC53BF3208277D0B9442C8C2DDAF41D9	14.06.2023 2:43:50
Белякова Кристина Дмитриевна	E8D069FE8883B1E4CD71DAF70EDFEF9F	14.06.2023 14:00:11
Белякова Кристина Дмитриевна	F045F6F7015EC87D7209AB8FC8DF43DC	14.06.2023 12:02:19
Иванов Иван Иванович	027BA231AEE65C78A914A00DCEBFF2CB	14.06.2023 18:00:46
Иванов Иван Иванович	324A473915EFA8080C1D79E6F4244974	23.06.2023 15:16:08
Иванов Иван Иванович	336702E357E9EC4137A49FF6A72BA87B	14.06.2023 14:23:52
Васильев Антон Владимирович	902B555AB05BDA0780B9162E78EAE817	24.06.2023 1:06:28

Рисунок 40 – Форма списка регистра сведений «Ключи авторизации пользователей»

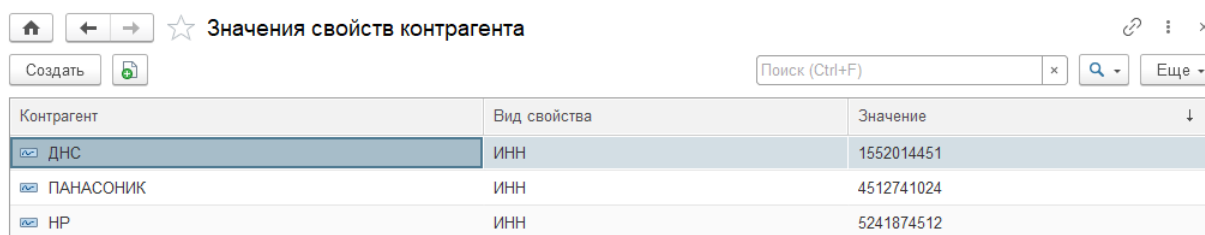
3. Регистр сведений «Пароли пользователей» предназначен для хранения и управления информацией о паролях пользователей системы. Пароли используются для защиты от несанкционированного доступа к информации и функционалу системы. Форма списка регистра приведена на рисунке 41.



Пользователь	Пароль	Дата изменения
Белякова Кристина Дмитриевна	02B4D1537DA8D8E6E58A3C7602EFCE25	11.06.2023 14:21:47
Иванов Иван Иванович	0B94C34836C7A344016962AC0D1DD20D	13.06.2023 9:06:04
Васильев Антон Владимирович	02B4D1537DA8D8E6E58A3C7602EFCE25	24.06.2023 1:06:28

Рисунок 41 – Форма списка регистра сведений «Пароли пользователей»

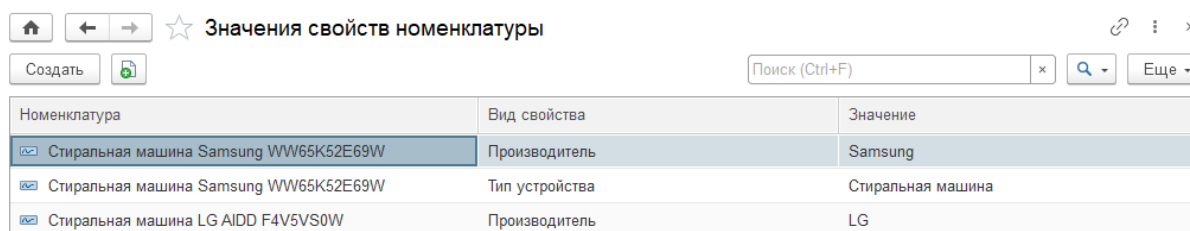
4. Регистр сведений «Значения свойств контрагента» предназначен для хранения и управления информацией о свойствах контрагентов, таких как юридический адрес, ИНН, КПП и другие. Форма списка регистра представлена на рисунке 42.



Контрагент	Вид свойства	Значение
ДНС	ИНН	1552014451
ПАНАСОНИК	ИНН	4512741024
НР	ИНН	5241874512

Рисунок 42 – Форма списка регистра сведений «Значения свойств контрагента»

5. Регистр сведений «Значения свойств номенклатуры» предназначен для хранения и управления информацией о свойствах номенклатуры, таких как артикул, производитель, размер и другие. Форма списка регистра сведений представлена на рисунке 43.



Номенклатура	Вид свойства	Значение
Стиральная машина Samsung WW65K52E69W	Производитель	Samsung
Стиральная машина Samsung WW65K52E69W	Тип устройства	Стиральная машина
Стиральная машина LG AIDD F4V5VS0W	Производитель	LG

Рисунок 43 – Форма списка регистра сведений «Значения свойств номенклатуры»

Регистры накопления. Регистры накопления предназначены для хранения и накопления таких данных, которые изменяются постоянно в течение определенного периода времени. Они используются для учета различных видов исторических фактов, расчетов и операций, связанных с использованием ресурсов организации. Они могут использоваться для учета расходов, доходов, остатков и т. д. В разрабатываемой системе представлено 4 регистра накопления.

1. Регистр накопления «Продажи» (вид – обороты) предназначен для хранения сведений по реализации материалов и услуг. Форма списка рассматриваемого регистра накопления представлена на рисунке 44.

🏠 ⬅ ➡ ☆ Движение по регистру Продажи 🔗 ⋮ ✕

Поиск (Ctrl+F) 🔍 ⌵ Еще ▾

Период	Номенклатура ↓	Контрагент	Менеджер	Количество	Выручка	Стоимость
• 22.06.2023 ...	Амортизаторы для стир...	Фролов Артём Андреевич	Иванов Иван Ива...	1,000	972,00	486,00
• 16.06.2023 ...	Датчик температурный ...	Белякова Кристина Дмит...	Иванов Иван Ива...	2,000	9 768,00	4 884,00
• 13.06.2023 ...	Датчик температурный ...	Агапов Алексей Владим...	Иванов Иван Ива...	1,000	4 884,00	2 442,00
• 12.06.2023 ...	Диагностика	Белякова Кристина Дмит...	Акимова Анастас...	1,000	550,00	550,00

Рисунок 44 – Форма списка регистра накопления «Продажи»

2. Регистр накопления «Остатки материалов» (вид – остатки) предназначен для хранения сведений по остаткам материалов. Форма списка регистра накопления приведена на рисунке 45.

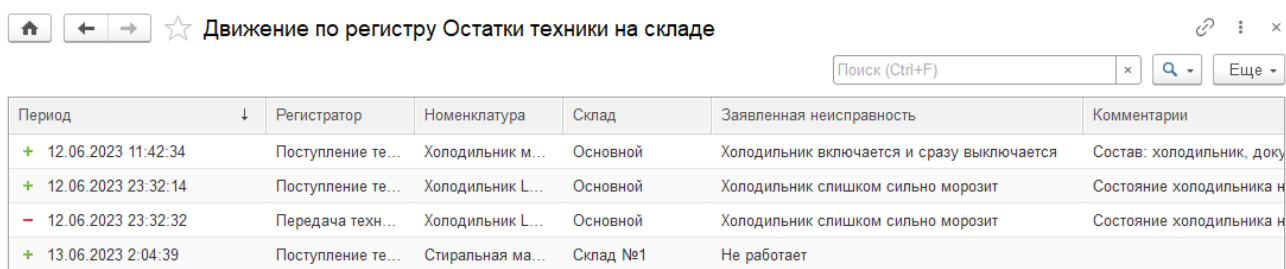
🏠 ⬅ ➡ ☆ Движение по регистру Остатки материалов 🔗 ⋮ ✕

Поиск (Ctrl+F) 🔍 ⌵ Еще ▾

Период	Регистратор	Номер строки	Материал ↓	Склад	Количество
+ 01.06.2022 12:00:00	Поступление материало...	1	Амортизаторы для стира...	Основной	10,000
- 22.06.2023 1:13:07	Реализация материалов ...	1	Амортизаторы для стира...	Основной	1,000
+ 01.06.2022 12:00:00	Поступление материало...	3	Датчик температурный м...	Основной	10,000
- 16.06.2023 3:15:24	Реализация материалов ...	1	Датчик температурный м...	Основной	2,000

Рисунок 45 – Форма списка регистра накопления «Остатки материалов»

3. Регистр накопления «Остатки техники на складе» (вид – остатки) предназначен для хранения информации о наличии и движении устройств, принятых у контрагентов для ремонта. Форма списка регистра представлена на рисунке 46.



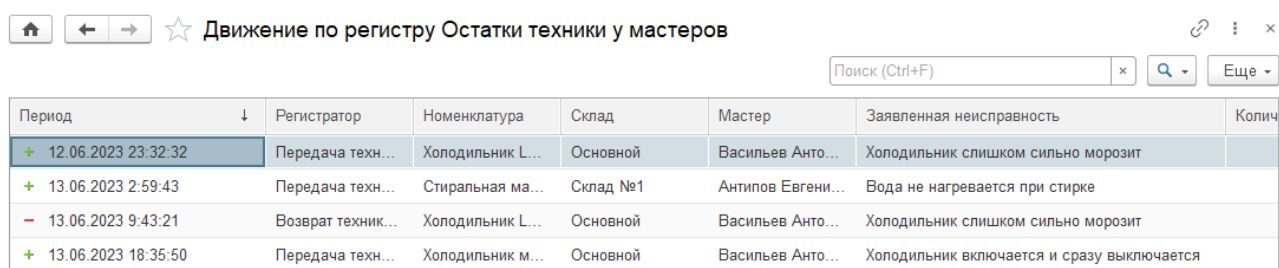
Движение по регистру Остатки техники на складе

Поиск (Ctrl+F) [x] [Q] [Еще]

Период	Регистратор	Номенклатура	Склад	Заявленная неисправность	Комментарии
+ 12.06.2023 11:42:34	Поступление те...	Холодильник м...	Основной	Холодильник включается и сразу выключается	Состав: холодильник, доку
+ 12.06.2023 23:32:14	Поступление те...	Холодильник L...	Основной	Холодильник слишком сильно морозит	Состояние холодильника н
- 12.06.2023 23:32:32	Передача техн...	Холодильник L...	Основной	Холодильник слишком сильно морозит	Состояние холодильника н
+ 13.06.2023 2:04:39	Поступление те...	Стиральная ма...	Склад №1	Не работает	

Рисунок 46 – Форма списка регистра накопления «Остатки техники на складе»

4. Регистр накопления «Остатки техники у мастеров» (вид – остатки) предназначен для хранения информации об устройствах, принятых мастерами для ремонта. Форма списка рассматриваемого регистра накопления приведена на рисунке 47.



Движение по регистру Остатки техники у мастеров

Поиск (Ctrl+F) [x] [Q] [Еще]

Период	Регистратор	Номенклатура	Склад	Мастер	Заявленная неисправность	Колич
+ 12.06.2023 23:32:32	Передача техн...	Холодильник L...	Основной	Васильев Анто...	Холодильник слишком сильно морозит	
+ 13.06.2023 2:59:43	Передача техн...	Стиральная ма...	Склад №1	Антипов Евгени...	Вода не нагревается при стирке	
- 13.06.2023 9:43:21	Возврат техник...	Холодильник L...	Основной	Васильев Анто...	Холодильник слишком сильно морозит	
+ 13.06.2023 18:35:50	Передача техн...	Холодильник м...	Основной	Васильев Анто...	Холодильник включается и сразу выключается	

Рисунок 47 – Форма списка регистра накопления «Остатки техники у мастеров»

Перечисления. Перечисления в 1С:Предприятие – это справочники, которые содержат перечень значений определенного типа. Назначение перечислений заключается в упрощении работы с данными и уменьшении количества возможных ошибок при их вводе. Они позволяют определить заранее набор возможных значений для конкретного поля или свойства объекта, а пользователь может выбрать нужное значение из списка. В разрабатываемой системе выделено 5 перечислений.

1. Перечисление «Виды номенклатуры» предназначено для классификации номенклатуры по её видам. Имеет следующие значения: материал, услуга, техника.

2. Перечисление «Виды контрагентов» предназначено для классификации контрагентов по их видам. Имеет следующие значения: физическое лицо, юридическое лицо, государственный орган, обособленное

подразделение.

3. Перечисление «Роли пользователей» предназначено для классификации пользователей системы по их ролям. Включает в себя следующие значения: менеджер, мастер, клиент.

4. Перечисление «Статусы заказов» предназначено для классификации статусов заказов в компании. Имеет следующие значения: принят в ремонт, передан мастеру, проведение диагностики, диагностика выполнена, согласование с клиентом, ожидание решения клиента, ожидание комплектующих, выполнение работ, готов к выдаче без ремонта, готов к выдаче, получен.

5. Перечисление «Типы складов» предназначено для классификации складов организации по их типам. Включает в себя следующие значения: оптовый, розничный.

Отчеты. Отчеты в 1С:Предприятие предназначены для анализа и отображения информации о бизнес-процессах организации. Они могут быть различной степени детализации и содержать информацию о финансовых показателях, продажах, закупках, остатках товаров на складах и других аспектах деятельности организации. В разрабатываемой системе представлены следующие отчеты: «Движение материалов», «Остатки техники на складе», «Ведомость продаж по контрагентам».

1. Отчет «Движение материалов» предназначен для анализа и отображения информации о движении товаров на складах сервисного центра. Он содержит информацию о поступлении и списании материалов, остатках на складах. Сформированный отчет представлен на рисунке 48.

Движения материалов

Параметры: Склад:
Период: 01.01.2023 - 30.06.2023

Склад				
Материал	Начальный остаток	Приход	Расход	Конечный остаток
Основной				
Двигатель (мотор) вентилятора для холодильника LG GA-B509CCIL	10,000			10,000
Опорно-цельный двигатель в сборе Samsung RF61K90407F	10,000			10,000
Амортизаторы для стиральной машины Samsung WW65K52E69W	10,000		1,000	9,000
Ручка двери для холодильника LG GA-B509CCIL	10,000		2,000	8,000
Датчик температурный морозильной камеры холодильника Samsung RF61K90407F	10,000		3,000	7,000
Насос (помпа) для стиральной машины LG AIDD F4V5S0W	10,000		4,000	6,000
Тэн (нагревательный элемент) для стиральной машины Samsung WW65K52E69W	5,000			5,000

Рисунок 48 – Отчет «Движение материалов»

2. Отчет «Остатки техники на складе» предназначен для анализа и отображения информации о наличии техники на складе. Сформированный отчет «Остатки техники на складе» представлен на рисунке 49.

Остатки техники на складе

Параметры: Период: 01.01.2021 - 31.12.2023

Кладовщик	Количество
Техника	
Кладовщики	3,000
Анисимов Никита Васильевич	1,000
Стиральная машина LG AIDD F4V5VS0W	1,000
Ларин Сергей Владимирович	2,000
Холодильник многодверный Samsung RF61K90407F	1,000
Стиральная машина LG AIDD F4V5VS0W	1,000
Итого	3,000

Рисунок 49 – Отчет «Остатки техники на складе»

3. Отчет «Ведомость продаж по контрагентам» предназначен для анализа и отображения информации о продажах материалов и услуг компании по каждому контрагенту за определенный период времени. Сформированный отчет «Ведомость продаж по контрагентам» представлен на рисунке 50.

Ведомость продаж по контрагентам

Параметры: Период: 01.05.2022 - 30.06.2023

Контрагент	Количество	Себестоимость	Выручка	Прибыль
Номенклатура				
Клиенты	10,000	14 558,00	29 116,00	14 558
Агапов Алексей Владимирович	1,000	2 442,00	4 884,00	2 442
Датчик температурный морозильной камеры холодильника Samsung RF61K90407F	1,000	2 442,00	4 884,00	2 442
Белякова Кристина Дмитриевна	8,000	11 630,00	23 260,00	11 630
Ручка двери для холодильника LG GA-B509CCIL	2,000	5 250,00	10 500,00	5 250
Датчик температурный морозильной камеры холодильника Samsung RF61K90407F	2,000	4 884,00	9 768,00	4 884
Насос (помпа) для стиральной машины LG AIDD F4V5VS0W	4,000	1 496,00	2 992,00	1 496
Фролов Артём Андреевич	1,000	486,00	972,00	486
Амортизаторы для стиральной машины Samsung WW65K52E69W	1,000	486,00	972,00	486
Итого	10,000	14 558,00	29 116,00	14 558

Рисунок 50 – Отчет «Ведомость продаж по контрагентам»

Общие формы. Общие формы представляют собой шаблоны, которые используются для создания и отображения списка объектов определенного типа. Для разрабатываемой ИС создана общая форма для рабочего стола, которая настроена для отображения списка остатков материалов и остатков техники на складе. Общая форма «Остатки на складе» представлена на рисунке 51.

Остатки на складе

Остатки материалов на складе на сегодня

Склад материалов:

Материал	Количество
Тэн (нагревательный элемент) для стиральной машины LG AIDD F4V5VS0W	4,000
Тэн (нагревательный элемент) для стиральной машины Samsung WW65K52E69W	5,000
Насос (помпа) для стиральной машины LG AIDD F4V5VS0W	6,000
Датчик температурный морозильной камеры холодильника Samsung RF61K90407F	7,000
Ручка двери для холодильника LG GA-B509CCIL	8,000
Амортизаторы для стиральной машины Samsung WW65K52E69W	9,000

Остатки техники на складе на сегодня

Склад техники:

Техника	Количество	Клиент
Стиральная машина LG AIDD F4V5VS0W	1,000	Белякова Кристина Дмитриевна
Холодильник многодверный Samsung RF61K90407F	1,000	Анисимова Ольга Дмитриевна

Рисунок 51 – Общая форма «Остатки на складе»

Константы. В 1С:Предприятие константы представляют собой специальные объекты, которые используются для хранения постоянных значений. В разрабатываемой системе используется константа «Адрес веб-приложения» для использования при отправке письма пользователю на электронную почту с ссылкой на установку нового пароля. Форма константы «Адрес веб-приложения» представлена на рисунке 52.

Адрес веб-приложения

Адрес веб-приложения:

Рисунок 52 – Форма константы «Адрес веб-приложения»

Планы видов характеристик. План видов характеристик в 1С:Предприятие представляет собой структуру, которая определяет набор характеристик, которые могут быть присвоены объектам в программе. В разрабатываемой системе используется 2 плана видов характеристик: «Характеристика контрагента», «Характеристика номенклатуры».

1. План видов характеристик «Характеристика контрагента» предназначен для определения дополнительных свойств контрагентов. План

видов характеристик представлен на рисунке 53.

Код	Наименование	Тип значения
000000001	ИНН	Строка
000000003	КПП	Значения характеристик контрагента
000000002	ОГРН	Значения характеристик контрагента
000000004	Юридический адрес	Значения характеристик контрагента

Рисунок 53 – План видов характеристик «Характеристика контрагента»

2. План видов характеристик «Характеристика номенклатуры» предназначен для определения дополнительных свойств номенклатуры. План видов характеристик представлен на рисунке 54.

Код	Наименование	Тип значения
000000004	Класс энергопотребления	Значения характеристик номенклатуры
000000006	Материал корпуса	Значения характеристик номенклатуры
000000001	Производитель	Значения характеристик номенклатуры
000000005	Размер	Значения характеристик номенклатуры
000000002	Тип устройства	Значения характеристик номенклатуры

Рисунок 54 – План видов характеристик «Характеристика номенклатуры»

Общие макеты. Общие макеты письма в 1С:Предприятие предназначены для упрощения процесса создания и отправки писем. Они содержат заранее заданный шаблон письма, который можно настроить под свои нужды. Описание общего макета письма включает в себя заголовок письма, текст письма, подпись и поля для заполнения. В разрабатываемой системе созданы 2 общих макета.

1. Общий макет «Письмо восстановления пароля» предназначен для отправки проверочного кода для сброса пароля. Общий макет представлен на рисунке 55.

Восстановление пароля

Здравствуйте! Забыли свой пароль от сервисного центра?
 Если вы хотите сбросить пароль для kr.belyakova19@gmail.com,
 Скопируйте код подтверждения в форму восстановления на сайте.

Код подтверждения: e2e91c

Если вы не запрашивали восстановление пароля, проигнорируйте это письмо.

Рисунок 55 – Общий макет «Письмо восстановления пароля»

Общий макет «Письмо для уведомления клиента» предназначен для автоматических уведомлений клиента об изменении статуса его заказа. Включает в себя следующие параметры: имя пользователя, номер заказа, наименование техники, текущий статус. Общий макет представлен на рисунке 56.

Информация по заказу

Здравствуйте, Белякова Кристина Дмитриевна!

Изменен статус вашего заказа №000000007 для техники Холодильник многодверный Samsung RF61K90407F.

Текущий статус: Принят в ремонт

Подробную информацию о заказе вы можете просмотреть в вашем личном кабинете клиента.

Рисунок 56 – Общий макет «Письмо для уведомления клиента»

Таким образом, разрабатываемая ИС управления сервисным центром состоит из 13 справочников, 6 документов, 5 регистров сведений, 4 регистров накопления, 5 перечислений, 3 отчетов, 1 константы, 2 планов видов характеристик, 1 общей формы, 1 общего макета.

4.2 Реализация конфигурации информационной системы

Реализация конфигурации в 1С:Предприятие 8.3 осуществляется на языке высокого уровня 1С. Для реализации системы были использованы следующие инструменты: модули серверного приложения и HTTP -сервисы.

Описание процедур и функций конфигурации системы представлено в таблице 8.

Таблица 8 – Описание процедур и функций конфигурации системы

Объект	Имя функции/процедуры	Описание функции/процедуры
Общий модуль «Общего назначения»	ОбработатьJSON(СтруктураОтвета, СтрокаJSON)	Функция предназначена для обработки JSON-данных, полученных от веб-сервиса. Функция разбирает строку JSON и заполняет соответствующую структуру объекта-ответа
	ДобавитьОшибку(СтруктураОтвета, ОписаниеОшибки)	Служит для добавления информации об ошибке в структуру ответа HTTP-сервиса

Продолжение таблицы 8

Объект	Имя функции/процедуры	Описание функции/процедуры
	ОтправитьПользователюПисьмо (ПочтовыйАдрес, Текст, Тема, ТипТекстаСообщения)	Функция позволяет отправить электронное письмо, указанному пользователю системы
Общий модуль «Авторизация»	Логин(ЛогинПользователя, Пароль)	Используется для авторизации пользователя в системе. Функция проверяет корректность введенного пользователем логина и пароля, и если они верны, то пользователь получает доступ к системе
	СформироватьТокены(Пользователь, Имя)	Используется для генерации токенов безопасности, которые используются для авторизации пользователя в веб-приложении. Токены создаются на основе логина пользователя и имеют ограниченный срок действия
	УстановитьПароль(КодПользователя, НовыйПароль)	Функция предназначена для установки нового пароля пользователя в системе
	ПолучитьКодПодтверждения(Логин)	Функция служит для генерации одноразового кода, который используется в процессе восстановления пароля
	ОтправитьПриглашение(Логин)	С помощью функции можно отправить пользователям сообщение с предложением зарегистрироваться на указанный адрес электронной почты. После получения приглашения, пользователь сможет зарегистрироваться и получить доступ к веб-интерфейсу системы
Общий модуль «Арі»	Добавить(СтруктураОтвета, Запрос)	Функция для добавления параметров ответа в структуру ответа веб-сервера при работе с HTTP-сервисами
	НазначитьМастера(СтруктураОтвета, Данные)	Используется для назначения ответственного мастера на выполнение ремонта техники
	ВыдачаТехники(СтруктураОтвета, Данные)	Функция для создания и проведения документа «Возврат техники контрагенту»
	Реализация(СтруктураОтвета, Данные)	Служит для создания и проведения документа «Реализация материалов»

Окончание таблицы 8

Объект	Имя функции/процедуры	Описание функции/процедуры
Общий модуль «Арі»	ПоступлениеТехники (СтруктураОтвета, Данные)	Функция предназначена для создания и проведения документа «Поступление техники от контрагента»
	ПоступлениеМатериалов (СтруктураОтвета, Данные)	Используется для создания и проведения документа «Поступление материалов»
	УстановкаЦен(СтруктураОтвета, Данные)	Функция служит для создания и проведения документа «Установка цен»
Модуль HTTP-сервиса «Веб-интерфейс»	АвторизацияPOST(Запрос)	Используется для авторизации пользователя в системе через HTTP POST запрос. Функция возвращает ответ сервера, который содержит информацию об авторизации пользователя
	ПередачаТехникиМастеруPOST (Запрос)	Предназначена для создания документа «Передача техники мастеру»
	ПолучитьСтруктуруОтвета()	Используется для получения структурированной информации об ответе, полученном от сервера после передачи запроса
	СформироватьОтвет (СтруктураОтвета, КодОтвета)	Функция нужна для формирования и отправки ответа на HTTP-запрос
	СформироватьJSON (СтруктураДанных)	Используется для создания JSON-строки на основе переданной ей структуры данных
	ВозвратТехникиPOST(Запрос)	Предназначена для создания документа «Возврат техники контрагенту»
	РеализацияМатериаловPOST(Запрос)	Служит для создания документа «Возврат техники контрагенту»
	ПоступлениеТехникиPOST(Запрос)	Функция для создания документа «Поступление техники от контрагента»
	УстановкаЦенPOST(Запрос)	Предназначена для создания документа «Установка цен»
	ПоступлениеМатериаловPOST (Запрос)	Используется для создания документа «Поступление материалов»
	НоменклатураPOST(Запрос)	Используется для создания нового элемента справочника «Номенклатура»

Исходные коды всех модулей конфигурации представлены в

приложении В.

Также в разрабатываемой системе включена поддержка автоматического REST-сервиса по протоколу OData. Он позволяет оперировать данными, используя в качестве запросов HTTP-команды. Для получения данных по протоколу OData используется HTTP-метод – GET, для обновления данных – PATCH/PUT, для удаления данных – DELETE.

4.3 Реализация веб-интерфейса

Веб-интерфейс реализован на языке JavaScript с использованием библиотеки React, которая использует компонентный подход. Компонентный подход – это методология программирования, в которой весь пользовательский интерфейс разбивается на небольшие, изолированные компоненты, которые могут взаимодействовать между собой, чтобы создать более крупные компоненты.

Описание основных компонентов системы представлено в таблице 9.

Таблица 9 – Описание основных компонентов системы

Наименование компонента	Описание	Основные функции
LoginPage	Страница авторизации пользователей	validation(login, password) – проверка логина и пароля
GetCode	Страница восстановления пароля	validation(login) – проверка логина
		codeRequested(email) – отправка кода подтверждения на почту
		confCodeRequested(data) – подтверждение кода
PasswordChange	Страница установки нового пароля	validation(password) – проверка пароля
		passwordSave(email, password) – установка нового пароля
StartPage	Стартовая страница, где происходит получение роли авторизованного пользователя и в соответствии с ней предоставляются права доступа к функциям системы	localStorage.getItem('userID') – получение логина пользователя из локального хранилища
		userData.getUserData() – получение данных о пользователе
ClientPage	Страница клиента сервисного центра	staff.getPostsStaff() – получение данных о сотрудниках

Продолжение таблицы 9

Наименование компонента	Описание	Основные функции
ClientPage	Страница клиента сервисного центра	items.getItems() – получение списка номенклатуры
		orders.getOrders() – получение данных о заказах
		getNameById(id, data) – получение наименования по идентификатору
		colorTag(status) – настройка цветов для статусов заказа
ManagerPage	Страница менеджера сервисного центра	staff.getPostStaff() – получение должностей
		items.getItems() – получение номенклатуры
		ref.getContactsProvider() – получение контактных лиц контрагента
		ref.getContactsProviderRole() – получение списка ролей контактных лиц контрагентов
		register.getPrice() – получение цен на номенклатуру
		ref.getNDS() – получение ставок НДС
ManagerPage	Страница менеджера сервисного центра	ref.getStocks() – получение списка складов
		getNameById(id, data) – получение наименования по идентификатору
SetMaster	Форма для закрепления заказа за мастером	getRoleByContact(id) – получение роли контактного лица контрагента по его идентификатору
		doc.createDocSelectMaster() – создание документа «Передача техники мастеру»
AddOrder	Форма для добавления нового заказа на ремонт	docInputDevices.createDoc() – создание документа «Поступление техники от контрагента»
ReturnDevice	Форма для оформления выдачи техники контрагенту	doc.createDocReturn() – создание документа «Возврат техники контрагенту»
AddMaterials	Форма для оформления приходной накладной	docInputMaterials.createMaterials() – создание документа «Поступление материалов»
AddPayment	Форма для оформления продажи материалов и услуг	register.getLeftover() – получение остатков материалов
		getTypeById(id) – получение типа номенклатуры по идентификатору
		doc.createDocPayment() – создание документа «Реализация материалов»
ItemsPage	Страница отображения списка номенклатуры	filterItems(devicesAll, type) – фильтрация номенклатуры по типу
AddItem	Форма добавления номенклатуры	item.create() – создание нового элемента справочника «Номенклатура»
AddPrice	Форма установки цен на номенклатуру	doc.createPrice() – создание документа «Установка цен»

Окончание таблицы 9

Наименование компонента	Описание	Основные функции
MasterPage	Страница мастера сервисного центра	orders.getOrders() – получение данных о заказах
		items.getItems() – получение списка номенклатуры

Исходные коды всех компонентов и функций системы представлены в приложении Г.

4.4 Реализация пользовательского интерфейса

Для удобства разрабатываемая ИС управления сервисным центром имеет веб-интерфейс, который позволит пользователям получать доступ к системе с компьютера, который подключен к интернету.

Для доступа к функциям системы необходимо войти в свой аккаунт. Страница авторизации представлена на рисунке 57.

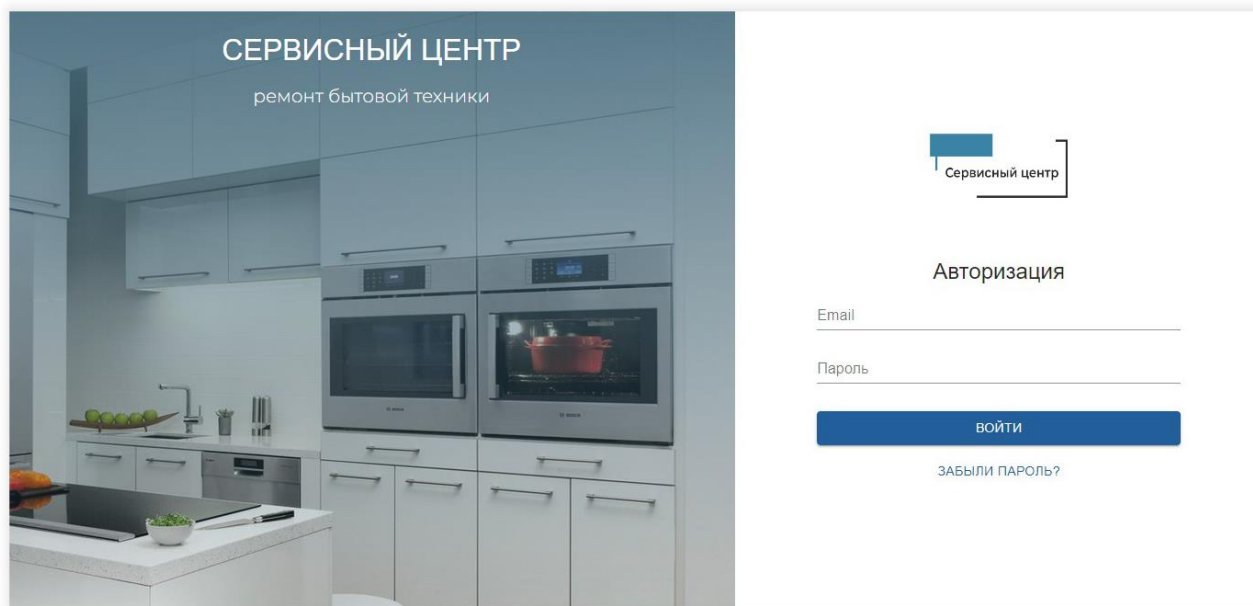


Рисунок 57 – Страница авторизации

На странице авторизации содержатся поля для ввода электронной почты и пароля, кнопка «Войти», кнопка «Забыли пароль». При нажатии на кнопку «Забыли пароль» пользователь перенаправляется на форму восстановления пароля, в которой нужно указать адрес электронной почты (рисунок 58).

Сервисный центр

Восстановление пароля

На указанный адрес будет отправлено письмо с кодом подтверждения

Email

ПОЛУЧИТЬ КОД ПОДТВЕРЖДЕНИЯ

Рисунок 58 – Форма восстановления пароля пользователя

При вводе адреса электронной почты и нажатии на кнопку «Получить под подтверждения» пользователь перенаправляется на форму ввода кода подтверждения, который был отправлен на электронную почту. Письмо с кодом представлено на рисунке 59.

Запрос на сброс пароля Входящие x

Сервисный центр <kr.belyakova@mail.ru>
кому: мне ▾

Восстановление пароля

Здравствуйте! Забыли свой пароль от сервисного центра?
Если вы хотите сбросить пароль для kr.belyakova19@gmail.com,
Скопируйте код подтверждения в форму восстановления на сайте.

Код подтверждения: e2e91c

Если вы не запрашивали восстановление пароля, проигнорируйте это письмо.

← Ответить → Переслать

Рисунок 59 – Письмо «Запрос на сброс пароля»

Форма восстановления пароля представлена на рисунке 60.

Восстановление пароля

Пожалуйста, проверьте электронную почту. Вы получите код подтверждения, который нужно будет ввести здесь, чтобы сбросить пароль

Email
kr.belyakova19@gmail.com

Код подтверждения

ПОДТВЕРДИТЬ

Рисунок 60 – Форма восстановления пароля

На форме указано поле для ввода кода подтверждения и кнопка «Подтвердить», при нажатии на которую открывается форма установки нового пароля (рисунок 61).

Рисунок 61 – Форма установки нового пароля

После того, как пользователь войдет в систему, он будет направлен на главную страницу, на которой будут отображаться все доступные для него заказы (рисунок 62).

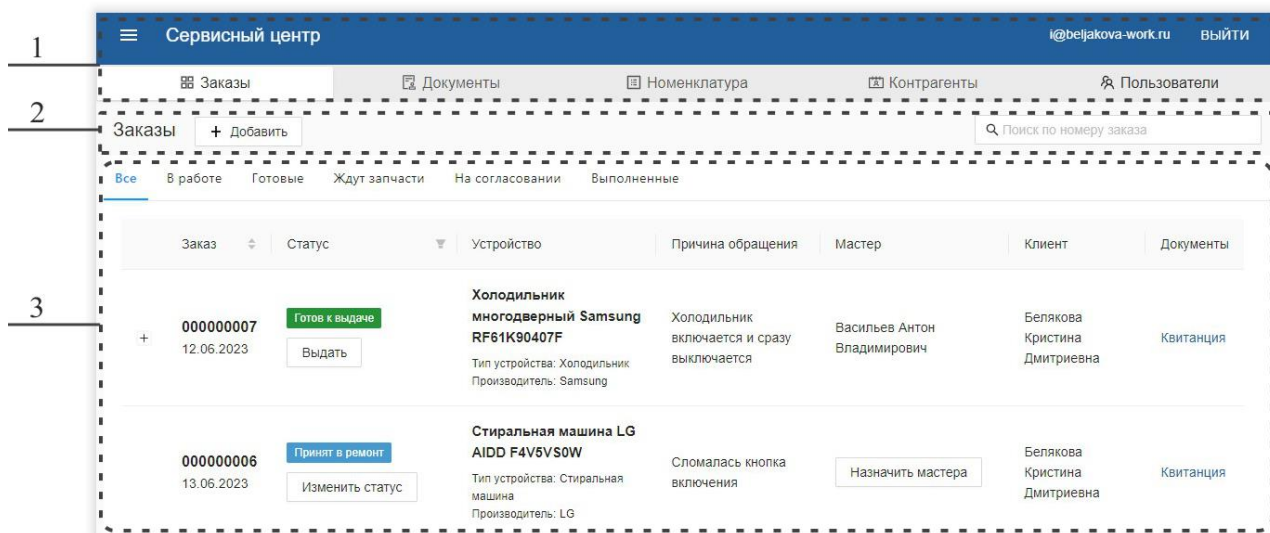


Рисунок 62 – Главная страница

Основные группы интерфейса главной страницы, изображенные на рисунке 62:

- 1 – панель навигации;
- 2 – командная панель;

3 – список заказов.

Назначение панели навигации заключается в том, чтобы помочь пользователю быстро и легко ориентироваться в структуре сайта и находить нужную информацию. Панель навигации содержит ссылки на следующие основные разделы системы: заказы, документы, номенклатура, контрагенты, пользователи. А также содержит кнопку «Выйти» для перехода на страницу авторизации.

Командная панель системы содержит кнопку добавления нового документа и поле для поиска заказа по его номеру.

Список заказов представлен в виде таблицы. Он содержит информацию о номере и дате заказа, статусе заказа, об устройстве, о проблеме в технике, о имени клиента и мастера. В колонке «Документы» содержатся ссылки на квитанции. Предоставлена возможность отсортировать заказы по номеру и дате, а также отфильтровать заказы по статусу. В колонке статуса заказа есть возможность изменить статус заказа, а также выдать технику, если заказ готов к выдаче. Если техника не закреплена за мастером, то в колонке «Мастер» отображается кнопка «Назначить мастера». Форма для закрепления техники за мастером представлена на рисунке 63.

Рисунок 63 – Форма для назначения мастера на заказ

Поле «Мастер» является обязательным, при раскрытии списка отображаются мастера сервисного центра (рисунок 64).

Рисунок 64 – Поле для выбора значения

Рядом с номером заказа присутствует кнопка раскрытия подробной информации о заказе (рисунок 65).

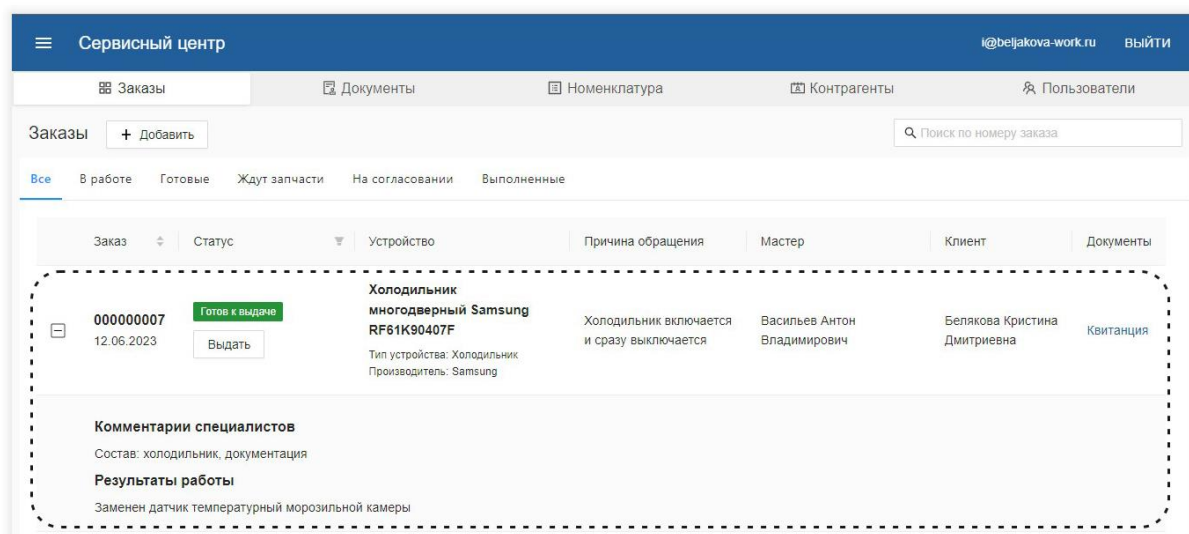


Рисунок 65 – Подробная информация о заказе

Для добавления нового заказа от клиента необходимо нажать на кнопку «Добавить» на командной панели. Форма для добавления нового документа «Поступление техники от контрагента» представлена на рисунке 66.

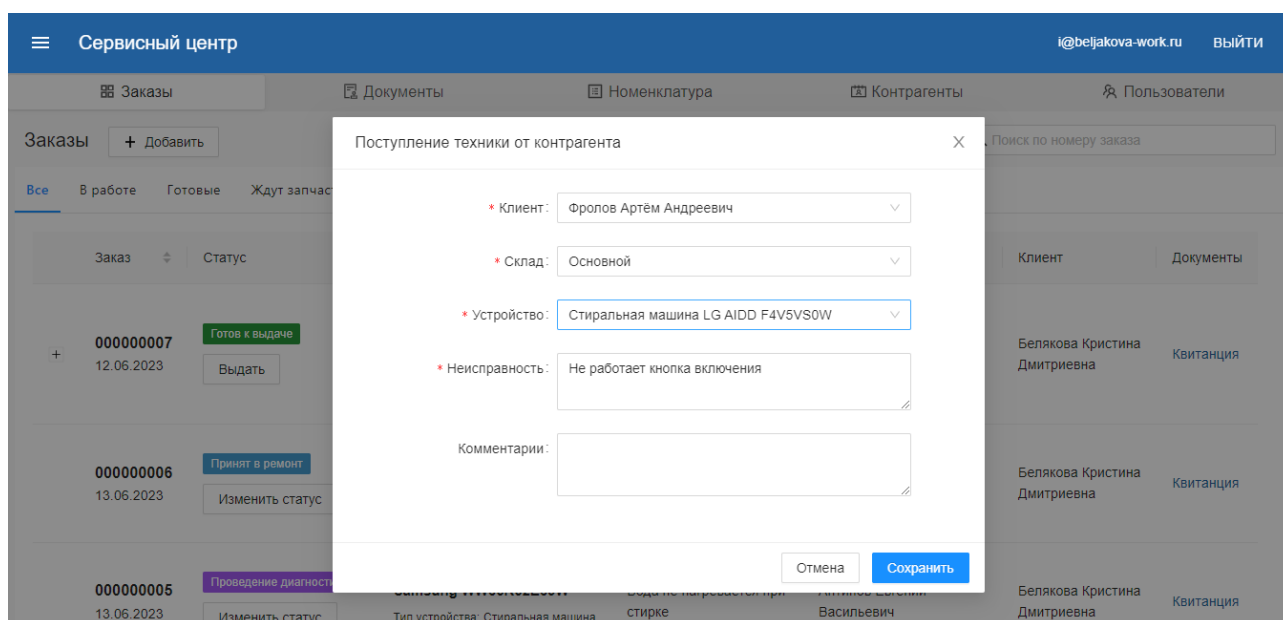


Рисунок 66 – Форма для добавления нового заказа

Обязательные поля помечены звездочкой красного цвета. На форме содержатся кнопки «Отмена» и «Сохранить». При нажатии на «Сохранить»

происходит создание нового документа «Поступление техники от контрагента».

Форма подбора номенклатуры для добавления документов с табличной частью представлена на рисунке 67.

Номенклатура	Вид	Цена	Остаток	Количество
☒ Насос (помпа) для стиральной машины LG AIDD F4V5VS0W	Материал	1122P	6	2
☐ Ручка двери для холодильника LG GA-B509CCIL	Материал	6300P	8	1
☐ Двигатель (мотор) вентилятора для холодильника LG GA-B509CCIL	Материал	2556P	10	1

Номенклатура	Количество	Цена	Сумма
Насос (помпа) для стиральной машины LG AIDD F4V5VS0W	2	1122P	2244P

Рисунок 67 – Форма подбора номенклатуры

Группы интерфейса формы подбора номенклатуры, изображенной на рисунке 67:

- 1 – строка поиска номенклатуры по наименованию;
- 2 – перечень номенклатуры;
- 3 – выбранная номенклатура.

В группе перечня номенклатуры представлена таблица, в которой содержится список услуг и остатков материалов. В колонке «Остаток» указан остаток материалов для выбранного склада на предыдущем этапе оформления документа. Поле «Количество» является доступным для изменения, минимальное количество равно 1, максимальное равно остатку материала. Для переноса номенклатуры в документ необходимо слева от наименования номенклатуры выбрать нужные элементы, нажав на чекбокс.

В таблице с выбранной номенклатурой предоставлена возможность изменения количества материала и удаления элемента из списка при нажатии на иконку «Корзина».

Для сохранения выбранной номенклатуры нежно нажать на кнопку

«Перенести в документ».

Для документов и справочников доступны действия «Редактировать» и «Удалить», для справочника «Пользователи» добавлена возможность отправки письма для регистрации в системе (рисунок 68).

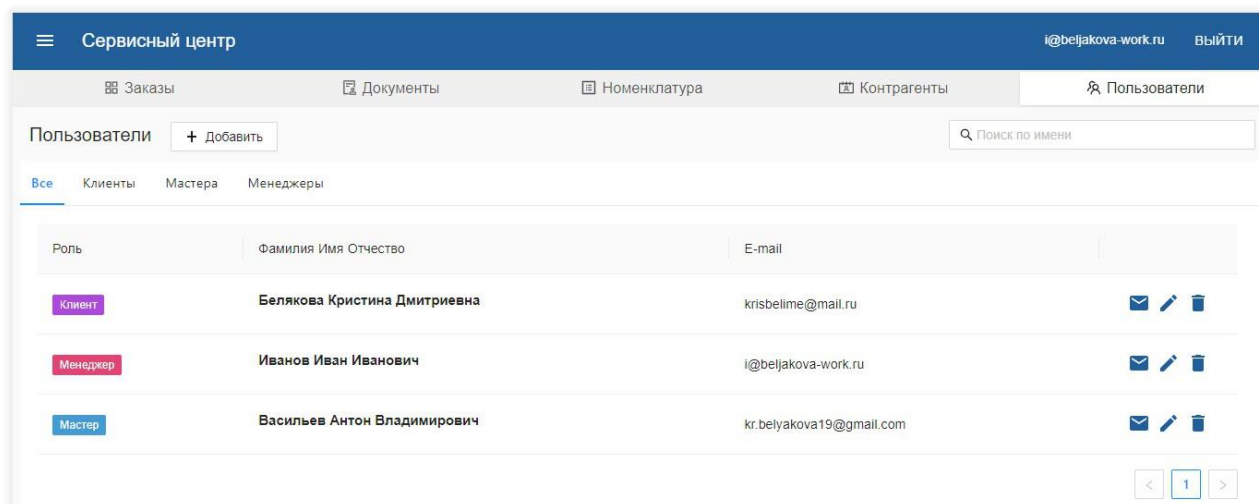


Рисунок 68 – Список пользователей системы

Таким образом, был разработан логически связанный и интуитивно понятный интерфейс, который обеспечивает удобство и эффективность пользовательского опыта. В разработку информационной системы были внедрены современные принципы дизайна, такие как единый стиль графических элементов и функциональная однородность, позволяющие осуществлять сходные действия с аналогичными элементами системы.

5 Тестирование информационной системы

Тестирование информационной системы является важной частью процесса разработки. Оно направлено на проверку работоспособности и соответствия системы требованиям.

Цель тестирования информационной системы управления сервисным центром заключается в обнаружении и устранении ошибок в работе системы.

Результатом тестирования должно быть устранение всех выявленных проблем и ошибок в системе и проверка ее работоспособности и безопасности. Критерии качества тестирования – это надежность, эффективность, производительность и удобство использования системы.

5.1 Функциональное тестирование

Функциональное тестирование информационной системы управления сервисным центром на платформе 1С:Предприятие с веб-интерфейсом включает проверку работы всех основных функций системы, таких как создание и проведение документов, формирование печатных форм и отчетов и т.д.

В процессе тестирования необходимо проверить корректность работы всех функций системы и ее удобство для пользователя.

В таблице 10 приведен список тестов функционального тестирования, которые были проведены для проверки работоспособности системы. В таблице приведены все необходимые данные об этих тестах, такие как их названия, описания и результаты.

Таблица 10 – Функциональное тестирование работы системы

№ п/п	Название теста	Предполагаемый результат	Полученный результат	Результаты тестирования
1	Проверка функции восстановления пароля при неверно введенной почты	Сообщение об ошибке, что в системе не зарегистрирована учетная запись на данную почту	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.1
2	Проверка функции восстановления пароля, если почта указана верно	Отправлен код подтверждения на электронную почту	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.2

Продолжение таблицы 10

№ п/п	Название теста	Предполагаемый результат	Полученный результат	Результаты тестирования
3	Проверка функции восстановления пароля при указании неверного кода подтверждения	Сообщение об ошибке «Неверный код подтверждения»	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.3
4	Проверка установки нового пароля при вводе 6 строчных латинских букв	Сообщение о том, что пароль должен содержать не менее 8 символов, одну заглавную латинскую букву, одну строчную латинскую букву, одну цифру и один специальный символ	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.4
5	Проверка установки нового пароля при вводе другого пароля в поле подтверждения	Сообщение об ошибке «Пароли должны совпадать»	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.5
6	Проверка авторизации в системе при вводе неверных данных	Сообщение об ошибке «Неверное имя пользователя или пароль»	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.6
7	Проверка авторизации при верном вводе данных	Произведен вход в личный кабинет. В веб-хранилище localStorage прописаны логин пользователя и токен доступа	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.7
8	Проверка назначения мастера за заказом 000000006	Создан документ «Передача техники мастеру» и изменен статус заказа на «Передан мастеру». На почту клиента отправлено уведомление об изменении статуса заказа	Соответствует ожидаемым результатам	Тест пройден Рисунки Д.8, Д.9
9	Проверка изменения статуса заказа 000000006 на «Проведение диагностики»	Статус заказа изменен и на почту клиента отправлено уведомление об изменении статуса заказа	Соответствует ожидаемым результатам	Тест пройден Рисунки Д.10, Д.11, Д.12
10	Проверка оформления выдачи техники по заказу 000000007	Создан документ «Возврат техники контрагенту» и изменен статус заказа на «Получен». Появится возможность просмотреть чек	Соответствует ожидаемым результатам	Тест пройден Рисунки Д.13, Д.14, Д.15

Окончание таблицы 10

11	Проверка создания документа «Реализация материалов»	Создан новый документ «Реализация материалов»	Соответствует ожидаемым результатам	Тест пройден Рисунки Д.16, Д.17
12	Проверка создания документа «Поступление материалов»	Создан новый документ «Поступление материалов»	Соответствует ожидаемым результатам	Тест пройден Рисунки Д.18, Д.19
13	Проверка создания нового пользователя в системе и отправка ссылки для регистрации на почту	Создан новый пользователь системы. На почту пользователя отправлена ссылка для установки пароля	Соответствует ожидаемым результатам	Тест пройден Рисунки Д.20, Д.21
14	Проверка формирования отчета «Ведомость продаж по контрагентам»	Открыть отчет «Ведомость продаж по контрагентам», выставить период, нажать на кнопку «Сформировать». Отчет успешно сформируется и выдаст соответствующий результат	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.22
15	Проверка формирования отчета «Движение материалов»	Открыть отчет «Движение материалов», указать период, нажать на кнопку «Сформировать». Отчет успешно сформируется и выдаст соответствующий результат	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.24
16	Проверка формирования отчета «Остатки техники на складе»	Открыть отчет «Остатки техники на складе», указать период, нажать на кнопку «Сформировать». Отчет успешно сформируется и выдаст соответствующий результат	Соответствует ожидаемым результатам	Тест пройден Рисунок Д.24

После выполнения функционального тестирования для разработанной системы управления сервисным центром можно сделать вывод о ее корректной работе и отсутствии явных ошибок и сбоев. Тестирование позволило подтвердить, что система функционирует без сбоев и эффективно выполняет свои функции. Данный результат свидетельствует о высоком качестве разработки системы и гарантирует ее надежное и безопасное использование в реальных условиях работы сервисного центра.

5.2 Метрика Холстеда

Метрика Холстеда – это метод измерения сложности программного кода. Метрика Холстеда состоит из двух основных параметров: объема и сложности кода. Объем кода определяется количеством строк кода, а сложность – количеством уникальных операторов и операндов [16].

Для функции «ДобавитьДокументВыдачаТехники» из общего модуля «Арі» был произведен расчет количественных показателей с использованием метрики Холстеда. Код функции приведен в листинге 1.

Листинг 1 – Функция «ДобавитьДокументВыдачаТехники»

```

Функция ДобавитьДокументВыдачаТехники(СтруктураОтвета, ЗапросНТТР)
    Ответ = Новый Структура("Данные, Код", СтруктураОтвета, 200);

    ДанныеJSON = ЗапросНТТР.ПолучитьТелоКакСтроку();
    Данные = ОбщегоНазначения.ОбработатьJSON(СтруктураОтвета,
ДанныеJSON);
    Если НЕ Данные = Неопределено Тогда
        Попытка
            НачатьТранзакцию();
            ВозвратТехники(СтруктураОтвета, Данные);
            ЗафиксироватьТранзакцию();
        Исключение
            Если ТранзакцияАктивна() Тогда
                ОтменитьТранзакцию();
            КонецЕсли;

            ОбщегоНазначения.ДобавитьОшибку(СтруктураОтвета, Ошибка())
        );
        Ответ.Код = 400;
        КонецПопытки;

    Иначе
        Ответ.Код = 400;
        КонецЕсли;

    Возврат Ответ;
КонецФункции

```

Для оценки сложности функции по метрике Холстеда, необходимо посчитать количество общих и уникальных операторов и операндов. Операторы представлены в таблице 11.

Таблица 11 – Операторы процедуры

Операторы	Число вхождений
Функция / КонецФункции	1
=	6
.	5
,	6
;	14
Если / Тогда	2
Иначе	1
КонецЕсли	2
Не	1
Попытка / Исключение / КонецПопытки	1
Возврат	1
Новый Структура	1
НачатьТранзакцию	1
ЗафиксироватьТранзакцию	1
ОтменитьТранзакцию	1
ТранзакцияАктивна	1
n1 = 16	N1 = 45

Операнды представлены в таблице 12.

Таблица 12 – Операнды процедуры

Операнды	Число вхождений
Данные	4
Код	1
200	1
400	2
ДанныеJSON	2
Неопределено	1
СтруктураОтвета	3
ЗапросHTTP	2
n2 = 8	N2 = 16

Подсчитав количество операторов и операндов, были рассчитаны длина, словарь, объем и сложность функции.

Длина функции рассчитывается по формуле:

$$N = N_1 + N_2 = 45 + 16 = 61,$$

где N – длина функции;

N_1 – количество использований операторов;

N_2 – количество использований операндов.

Словарь функции рассчитывается по формуле:

$$n = n_1 + n_2 = 16 + 8 = 24,$$

где n – словарь функции;
 n_1 – количество операторов;
 n_2 – количество операндов.

Объем функции рассчитывается по формуле:

$$V = N \cdot \log_2 n = 61 \cdot 4,585 = 279,685,$$

где V – объем функции;
 N – длина функции;
 n – словарь функции.

Сложность функции рассчитывается по формуле:

$$HD = \left(\frac{n_1}{2}\right) \cdot \left(\frac{N_2}{n_2}\right) = \left(\frac{16}{2}\right) \cdot \left(\frac{16}{8}\right) = 16,$$

где HD – сложность функции;
 n_1 – количество операторов;
 n_2 – количество операндов;
 N_2 – количество использований операндов.

Таким образом, сложность функции приблизительно равна 16, что говорит о её относительной простоте.

Заключение

В результате выполнения выпускной квалификационной работы была разработана информационная система управления сервисным центром на платформе 1С:Предприятие с веб-интерфейсом.

В процессе выполнения работы было проведено детальное исследование предметной области, включая поиск и анализ систем аналогичного назначения. Путем сравнительного анализа были выявлены их функциональные возможности, что помогло определить основные требования к разрабатываемой системе. На основании этого были выбраны средства для проектирования и разработки, представляющие наилучшую комбинацию функциональности и производительности.

Кроме того, было проведено детальное планирование разработки, учитывающее все этапы работы, и составлен бюджет на разработку информационной системы.

Проведено проектирование информационной системы, которое включало в себя создание диаграмм UML, построение диаграммы бизнес-процессов и формирование логической модели базы данных.

Была проведена разработка системы с использованием средств платформы 1С: Предприятие 8.3. Затем был создан веб-интерфейс системы, который основывается на HTTP-сервисах, протоколе oData, веб-сервисе Apache, языке программирования JavaScript и библиотеке компонентов React, которая дополнительно подключает MobX для управления состоянием приложения.

После завершения разработки системы было проведено комплексное тестирование, включающее в себя функциональное тестирование и тестирование метрикой Холстеда. Результаты тестирования показали, что разработанная система управления сервисным центром работает корректно и соответствует заявленным требованиям.

В результате разработки системы управления сервисным центром были успешно достигнуты все поставленные цели, включая проектирование,

реализацию и тестирование программной информационной системы. Одним из главных достижений является повышение оперативности работы сервисного центра за счет автоматизации процессов приема и обработки заявок, а также улучшение качества обслуживания клиентов благодаря созданию личного кабинета клиента и рассылка уведомлений по электронной почте.

Список использованных источников

1. Gincore // Startpack [Электронный ресурс]. URL: <https://startpack.ru/application/gincore/> (дата обращения 20.04.2023).
2. Программы для сервисных центров и ремонтных мастерских // LiveBusiness [Электронный ресурс]. URL: <https://www.livebusiness.ru/tools/service/> (дата обращения 20.04.2023).
3. 1С:Предприятие 8. Управление сервисным центром // 1С:Предприятие 8 [Электронный ресурс]. URL: <https://solutions.1c.ru/catalog/service-center> (дата обращения 20.04.2023).
4. Возможности Visio 2021// Microsoft [Электронный ресурс]. URL: <https://support.microsoft.com/ru-ru/office/> (дата обращения 21.04.2023).
5. Функциональные возможности AllFusion ERwin Data Modeler // Interface [Электронный ресурс]. URL: <http://www.interface.ru/home.asp?artId=5817> (дата обращения 21.04.2023).
6. Обзор системы 1С:Предприятие 8 // 1С:Предприятие 8 [Электронный ресурс]. URL: <https://v8.1c.ru/tekhnologii/overview/> (дата обращения 22.04.2023).
7. Что такое React и как его освоить // Академия Яндекса [Электронный ресурс]. URL: <https://academy.yandex.ru/journal/chto-takoe-react-i-kak-ego-osvoit> (дата обращения 22.04.2023).
8. Apache // Skillfactory [Электронный ресурс]. URL: <https://blog.skillfactory.ru/glossary/apache/> (дата обращения 21.04.2023).
9. Диаграмма Ганта и способы ее использования // GeekBrains [Электронный ресурс]. URL: <https://gb.ru/blog/diagramma-ganta/> (дата обращения 24.04.2023).
10. Методология IDEF1X // ITteach [Электронный ресурс]. URL: <https://itteach.ru/bpwin/metodologiya-idef1x> (дата обращения 29.04.2023).
11. Использование диаграммы вариантов использования UML при проектировании программного обеспечения // Хабр [Электронный ресурс]. URL: <https://habr.com/ru/articles/566218/> (дата обращения 02.05.2023).

12. UML для бизнес-моделирования: зачем нужны диаграммы процессов // Evergreen [Электронный ресурс]. URL: <https://evergreens.com.ua/ru/articles/uml-diagrams.html> (дата обращения 02.05.2023).

13. Простое руководство по диаграммам активности UML // Creately [Электронный ресурс]. URL: <https://creately.com/blog/ru/uncategorized-ru/учебник-по-диаграмме-активности/> (дата обращения 04.05.2023).

14. Элементы графической нотации диаграммы последовательности // НОУ ИНТУИТ [Электронный ресурс]. URL: <https://intuit.ru/studies/courses/32/32/lecture/1014> (дата обращения 12.05.2023).

15. Элементы графической нотации диаграммы кооперации // НОУ ИНТУИТ [Электронный ресурс]. URL: <https://intuit.ru/studies/courses/32/32/lecture/1012> (дата обращения 16.05.2023).

16. Программный код и его метрики // Хабр [Электронный ресурс]. URL: <https://habr.com/ru/companies/intel/articles/106082/> (дата обращения 29.05.2023).

ПРИЛОЖЕНИЕ А
ГЛОССАРИЙ
(справочное)

HTTP-сервис – это служба, которая предоставляет доступ к веб-страницам, приложениям или сервисам через протокол HTTP (Hypertext Transfer Protocol).

JavaScript – это язык программирования, используемый для создания интерактивных веб-страниц и приложений.

oData (Open Data Protocol) – это протокол, позволяющий доступ к данным через стандартизированный RESTful API.

REST – это архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

RESTful API (Representational State Transfer API) – это механизм передачи данных между клиентом и сервером через интернет на основе архитектуры REST.

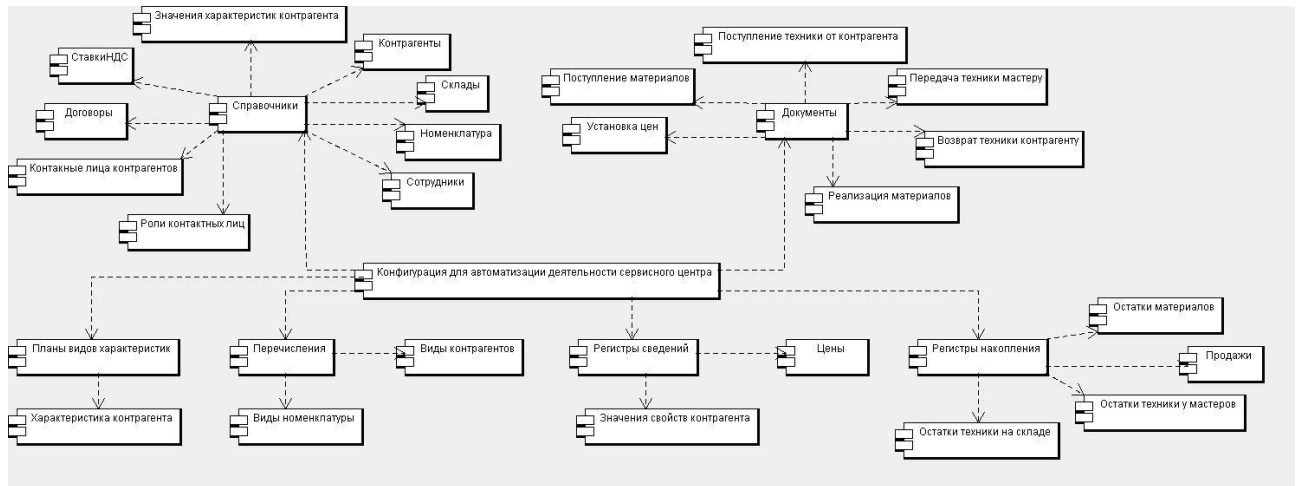
UML (Unified Modeling Language) – это язык графического моделирования, используемый для создания, визуализации, документирования и анализа системных проектов.

База данных (БД) – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.

Информационная система (ИС) – это совокупность взаимосвязанных компонентов (аппаратных, программных, информационных и организационных), которые используются для получения, обработки, хранения, передачи и использования информации.

Токен – это уникальная последовательность символов, которая выделяется для идентификации пользователя или приложения в системе.

ПРИЛОЖЕНИЕ Б
ДИАГРАММА КОМПОНЕНТОВ
(рекомендуемое)



ПРИЛОЖЕНИЕ В
ЛИСТИНГ КОНФИГУРАЦИИ
(рекомендуемое)

Листинг В.1 – Модуль объекта документа «Поступление техники от контрагента»

Процедура ОбработкаПроведения (Отказ, Режим)

```
// регистр ПоступлениеТехникиОтКонтрагента
Движения.ПоступлениеТехникиОтКонтрагента.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Движение = Движения.ПоступлениеТехникиОтКонтрагента.Добавить ();
    Движение.Период = Дата;
    Движение.ДатаЗаказа = Дата;
    Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Склад = Склад;
    Движение.Клиент = Клиент;
    Движение.Менеджер = Менеджер;
    Движение.ЗаявленнаяНеисправность =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
    Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;
    Движение.НомерЗаказа = Номер;
    Движение.Статус = ТекСтрокаПереченьНоменклатуры.Статус;
КонецЦикла;

// регистр Заявки
Движения.Заявки.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Движение = Движения.Заявки.Добавить ();
    Движение.Период = Дата;
    Движение.НомерЗаказа = Номер;
    Движение.ПричинаОбращения =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
    Движение.Клиент = Клиент;
    Движение.Менеджер = Менеджер;
    Движение.ДатаЗаказа = Дата;
    Движение.Заявка = Ссылка;
    Движение.Статус = ТекСтрокаПереченьНоменклатуры.Статус;
    Движение.Устройство = ТекСтрокаПереченьНоменклатуры.Номенклатура;

    Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;

КонецЦикла;

// регистр ОстаткиТехникиНаСкладе Приход
Движения.ОстаткиТехникиНаСкладе.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Движение = Движения.ОстаткиТехникиНаСкладе.Добавить ();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Клиент = Клиент;
    Движение.Склад = Склад;
    Движение.Менеджер = Менеджер;
    Движение.ЗаявленнаяНеисправность =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
    Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;
    Движение.Количество = 1;
КонецЦикла;

ТабДок = Новый ТабличныйДокумент;
```

```

НайденныйДокумент =
Документы.ПоступлениеТехникиОтКонтрагента.НайтиПоНомеру (Номер) ;
Документы.ПоступлениеТехникиОтКонтрагента.Печать (ТабДок,
НайденныйДокумент) ;

МестоположениеФайла = "C:\Users\kristina\Desktop\service-
center\src\doc\receipts\" + Номер + ".pdf";
ТабДок.Записать (МестоположениеФайла, ТипФайлаТабличногоДокумента.PDF) ;
КонецПроцедуры

```

Листинг В.2 – Модуль объекта документа «Передача техники мастеру»

Процедура ОбработкаПроведения (Отказ, Режим)

```

// регистр ПередачаТехникиМастеру
Движения.ПередачаТехникиМастеру.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Движение = Движения.ПередачаТехникиМастеру.Добавить () ;
    Движение.Период = Дата;
    Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Склад = Склад;
    Движение.Клиент = Клиент;
    Движение.Менеджер = Менеджер;
    Движение.ЗаявленнаяНеисправность =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
    Движение.НомерЗаказа = Заявка.Номер;
    Движение.Мастер = Мастер;
    Движение.ДатаЗаказа = ДатаЗаказа;
    Движение.Результат = ТекСтрокаПереченьНоменклатуры.Результат;
    Движение.Статус = ТекСтрокаПереченьНоменклатуры.Статус;
    Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;
КонецЦикла;

Запрос = Новый Запрос;
Запрос.Текст ="ВЫБРАТЬ
                | ПоступлениеТехникиОтКонтрагента.Номер КАК Номер
                | ИЗ
                | Документ.ПоступлениеТехникиОтКонтрагента КАК
ПоступлениеТехникиОтКонтрагента
                | ГДЕ
                | ПоступлениеТехникиОтКонтрагента.Ссылка = &Заявка";
Запрос.УстановитьПараметр ("Заявка", Заявка);
РезультатЗапроса = Запрос.Выполнить () ;
Выборка = РезультатЗапроса.Выбрать () ;
НомерЗаявки = 0;
Пока Выборка.Следующий () Цикл
    НомерЗаявки = Выборка.Номер;
КонецЦикла;

// регистр Заявки
Движения.Заявки.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Движение = Движения.Заявки.Добавить () ;
    Движение.Период = Дата;
    Движение.Клиент = Клиент;
    Движение.Менеджер = Менеджер;
    Движение.Мастер = Мастер;
    Движение.НомерЗаказа = НомерЗаявки;
    Движение.ДатаЗаказа = ДатаЗаказа;
    Движение.Устройство = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Статус = ТекСтрокаПереченьНоменклатуры.Статус;

```

```

        Движение.ПричинаОбращения =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
        Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;
        Движение.Результат = ТекСтрокаПереченьНоменклатуры.Результат;
        Движение.Заявка = Заявка;
    КонечЦикла;

    // регистр ОстаткиТехникиУМастеров Приход
    Движения.ОстаткиТехникиУМастеров.Записывать = Истина;
    Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
        Движение = Движения.ОстаткиТехникиУМастеров.Добавить ();
        Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
        Движение.Период = Дата;
        Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Склад = Склад;
        Движение.Клиент = Клиент;
        Движение.Менеджер = Менеджер;
        Движение.Мастер = Мастер;
        Движение.ЗаявленнаяНеисправность =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
        Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;
        Движение.Результат = ТекСтрокаПереченьНоменклатуры.Результат;
        Движение.Количество = 1;
    КонечЦикла;

    // регистр ОстаткиТехникиНаСкладе Расход
    Движения.ОстаткиТехникиНаСкладе.Записывать = Истина;
    Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
        Движение = Движения.ОстаткиТехникиНаСкладе.Добавить ();
        Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
        Движение.Период = Дата;
        Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Склад = Склад;
        Движение.Клиент = Клиент;
        Движение.Менеджер = Менеджер;
        Движение.ЗаявленнаяНеисправность =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
        Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;
        Движение.Количество = 1;
    КонечЦикла;

    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
        |     Пользователи.Код КАК Код,
        |     Пользователи.Наименование КАК Наименование
        | ИЗ
        |     Справочник.Пользователи КАК Пользователи
        | ГДЕ
        |     Пользователи.СсылкаНаПользователя = &Клиент" ;
    Запрос.УстановитьПараметр("Клиент", Клиент);
    РезультатЗапроса = Запрос.Выполнить ();
    Выборка = РезультатЗапроса.Выбрать ();

    Если Выборка.Количество() > 0 Тогда

        Пока Выборка.Следующий() Цикл
            СтруктураПользователя = Выборка;
        КонечЦикла;

        Тема = "Изменение статуса заказа в сервисном центре";

        Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры
Цикл

```

```

        Статус = ТекСтрокаПереченьНоменклатуры.Статус;

        ШаблонHTML =
ПолучитьОбщийМакет ("ПисьмоУведомленияКлиента") .ПолучитьТекст ();

        ТекстСообщенияHTML = СтрЗаменить (ШаблонHTML, "&Статус",
ПолучитьСтатус (Статус) );
        ТекстСообщенияHTML = СтрЗаменить (ТекстСообщенияHTML,
"&Техника", ТекСтрокаПереченьНоменклатуры.Номенклатура);
        ТекстСообщенияHTML = СтрЗаменить (ТекстСообщенияHTML,
"&ИмяПользователя", СтруктураПользователя.Наименование);
        ТекстСообщенияHTML = СтрЗаменить (ТекстСообщенияHTML,
"&НомерЗаказа", НомерЗаявки);

        Авторизация.ОтправитьПользователюПисьмо (СтруктураПользователя.Код,
ТекстСообщенияHTML, Тема, ТипТекстаПочтовогоСообщения.HTML);
        КонецЦикла;

        КонецЕсли;

КонецПроцедуры

Функция ПолучитьСтатус (Статус)
    Если Статус = Перечисления.СписокСтатусов.ВыполнениеРабот Тогда
        Возврат ("Выполнение работ");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.ГотовКВыдаче Тогда
        Возврат ("Готов к выдаче");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.ГотовКВыдачеБезРемонта
Тогда
        Возврат ("Готов к выдаче без ремонта");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.ДиагностикаВыполнена Тогда
        Возврат ("Диагностика выполнена");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.ОжиданиеКомплектующих Тогда
        Возврат ("Ожидание комплектующих");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.ОжиданиеРешенияКлиента
Тогда
        Возврат ("Ожидание решения клиента");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.ПереданМастеру Тогда
        Возврат ("Передан мастеру");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.Получен Тогда
        Возврат ("Получен");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.ПринятВРемонт Тогда
        Возврат ("Принят в ремонт");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.ПроведениеДиагностики Тогда
        Возврат ("Проведение диагностики");
    ИначеЕсли Статус = Перечисления.СписокСтатусов.СогласованиеСКлиентом Тогда
        Возврат ("Согласование с клиентом");
    КонецЕсли;
КонецФункции;

```

Листинг В.3 – Модуль объекта документа «Возврат техники контрагенту»

Процедура ОбработкаПроведения (Отказ, Режим)

```

// регистр ВозвратТехникиКонтрагенту
Движения.ВозвратТехникиКонтрагенту.Записывать = Истина;
Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
    Движение = Движения.ВозвратТехникиКонтрагенту.Добавить ();
    Движение.Период = Дата;
    Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
    Движение.Склад = Склад;

```

```

        Движение.Клиент = Клиент;
        Движение.Менеджер = Менеджер;
        Движение.ЗаявленнаяНеисправность =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
        Движение.НомерЗаказа = Заявка.Номер;
        Движение.Мастер = Мастер;
        Движение.ДатаЗаказа = ДатаЗаявки;
        Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Стоимость;
        Движение.Результат = ТекСтрокаПереченьНоменклатуры.Результат;
        Движение.Статус = ТекСтрокаПереченьНоменклатуры.Статус;
        Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;
        Движение.ДокументНаОплату =
ТекСтрокаПереченьНоменклатуры.ДоговорНаОплату;
        КонечЦикла;

        Запрос = Новый Запрос;
        Запрос.Текст = "ВЫБРАТЬ
                        | ПоступлениеТехникиОтКонтрагента.Номер КАК Номер
                        | ИЗ
                        | Документ.ПоступлениеТехникиОтКонтрагента КАК
ПоступлениеТехникиОтКонтрагента
                        | ГДЕ
                        | ПоступлениеТехникиОтКонтрагента.Ссылка = &Заявка";
        Запрос.УстановитьПараметр("Заявка", Заявка);
        РезультатЗапроса = Запрос.Выполнить();
        Выборка = РезультатЗапроса.Выбрать();
        НомерЗаявки = 0;
        Пока Выборка.Следующий() Цикл
            НомерЗаявки = Выборка.Номер;
        КонечЦикла;

        // регистр Заявки
        Движения.Заявки.Записывать = Истина;
        Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
            Движение = Движения.Заявки.Добавить();
            Если
ЗначениеЗаполнено(ТекСтрокаПереченьНоменклатуры.ДоговорНаОплату) Тогда
                Движение.СчетНаОплату =
ТекСтрокаПереченьНоменклатуры.ДоговорНаОплату.Номер;
                КонечЕсли;
            Движение.Период = Дата;
            Движение.ПричинаОбращения =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
            Движение.Клиент = Клиент;
            Движение.Заявка = Заявка;
            Движение.Менеджер = Менеджер;
            Движение.НомерЗаказа = НомерЗаявки;
            Движение.Мастер = Мастер;
            Движение.ДатаЗаказа = ДатаЗаявки;
            Движение.Устройство = ТекСтрокаПереченьНоменклатуры.Номенклатура;
            Движение.Статус = ТекСтрокаПереченьНоменклатуры.Статус;
            Движение.СтоимостьЗаказа = ТекСтрокаПереченьНоменклатуры.Стоимость;
            Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;
            Движение.Результат = ТекСтрокаПереченьНоменклатуры.Результат;
        КонечЦикла;

        // регистр ОстаткиТехникиУМастеров Расход
        Движения.ОстаткиТехникиУМастеров.Записывать = Истина;
        Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
            Движение = Движения.ОстаткиТехникиУМастеров.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
            Движение.Период = Дата;
            Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;

```

```

        Движение.Склад = Склад;
        Движение.Клиент = Клиент;
        Движение.Менеджер = Менеджер;
        Движение.Мастер = Мастер;
        Движение.ЗаявленнаяНеисправность =
ТекСтрокаПереченьНоменклатуры.ЗаявленнаяНеисправность;
        Движение.Комментарии = ТекСтрокаПереченьНоменклатуры.Комментарии;
        Движение.Результат = ТекСтрокаПереченьНоменклатуры.Результат;
        Движение.Количество = 1;
    КонечЦикла;

КонечПроцедуры

```

Листинг В.4 – Модуль объекта документа «Реализация материалов»

Процедура ОбработкаПроведения(Отказ, Режим)

```

    СуммаВсего = 0;
    Для Каждого ТекСтрокаТовары Из ПереченьНоменклатуры Цикл
        СуммаВсего = СуммаВсего + ТекСтрокаТовары.Всего;
    КонечЦикла;

    Всего = СуммаВсего;
    Записать ();

    // регистр ОстаткиМатериалов Расход
    Движения.ОстаткиМатериалов.Записывать = Истина;
    Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
        Вид =
ПолучитьВидНоменклатуры(ТекСтрокаПереченьНоменклатуры.Номенклатура);
        Если (Вид = ВидМатериал()) Тогда
            Движение = Движения.ОстаткиМатериалов.Добавить ();
            Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
            Движение.Период = Дата;
            Движение.Материал =
ТекСтрокаПереченьНоменклатуры.Номенклатура;
            Движение.Склад = Склад;
            Движение.Количество =
ТекСтрокаПереченьНоменклатуры.Количество;
        КонечЕсли;
    КонечЦикла;

    // регистр Продажи
    Движения.Продажи.Записывать = Истина;
    Для Каждого ТекСтрокаПереченьНоменклатуры Из ПереченьНоменклатуры Цикл
        Движение = Движения.Продажи.Добавить ();
        Движение.Период = Дата;
        Движение.Номенклатура = ТекСтрокаПереченьНоменклатуры.Номенклатура;
        Движение.Контрагент = Контрагент;
        Движение.Менеджер = Менеджер;
        Движение.Количество = ТекСтрокаПереченьНоменклатуры.Количество;
        Движение.Выручка = ТекСтрокаПереченьНоменклатуры.Всего;

        Вид =
ПолучитьВидНоменклатуры(ТекСтрокаПереченьНоменклатуры.Номенклатура);
        Если (Вид = ВидУслуга()) Тогда
            Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Всего;
        Иначе
            Движение.Стоимость = ТекСтрокаПереченьНоменклатуры.Всего * 50
/ 100;
        КонечЕсли;
    КонечЦикла;

```

```

ТабДок = Новый ТабличныйДокумент;

НайденныйДокумент = Документы.РеализацияМатериалов.НайтиПоНомеру (Номер) ;
Документы.РеализацияМатериалов.Печать (ТабДок, НайденныйДокумент) ;

МестоположениеФайла = "C:\Users\kristina\Desktop\service-
center\src\doc\checks\" + Номер + ".pdf";
ТабДок.Записать (МестоположениеФайла, ТипФайлаТабличногоДокумента.PDF) ;

КонецПроцедуры

&НаСервере
Функция ПолучитьВидНоменклатуры (ЭлементНоменклатуры)
    Возврат ЭлементНоменклатуры.ВидНоменклатуры;
КонецФункции

&НаСервере
Функция ВидУслуга ()
    Возврат Перечисления.ВидыНоменклатуры.Услуга;
КонецФункции

&НаСервере
Функция ВидМатериал ()
    Возврат Перечисления.ВидыНоменклатуры.Материал;
КонецФункции

```

Листинг В.5 – Модуль объекта документа «Поступление материалов»

Процедура ОбработкаПроведения (Отказ, Режим)

```

СуммаВсего = 0;
Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
    СуммаВсего = СуммаВсего + ТекСтрокаМатериалы.Сумма;
КонецЦикла;

Всего = СуммаВсего;
Записать ();

// регистр ОстаткиМатериалов Приход
Движения.ОстаткиМатериалов.Записывать = Истина;
Для Каждого ТекСтрокаМатериалы Из Материалы Цикл
    Движение = Движения.ОстаткиМатериалов.Добавить ();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Материал = ТекСтрокаМатериалы.Материал;
    Движение.Склад = Склад;
    Движение.Количество = ТекСтрокаМатериалы.Количество;
КонецЦикла;
КонецПроцедуры

```

Листинг В.6 – Модуль объекта документа «Установка цен»

Процедура ОбработкаПроведения (Отказ, Режим)

```

// регистр Цены
Движения.Цены.Записывать = Истина;
Для Каждого ТекСтрокаСписокЦен Из СписокЦен Цикл
    Движение = Движения.Цены.Добавить ();
    Движение.Период = Дата;
    Движение.Номенклатура = ТекСтрокаСписокЦен.Номенклатура;
    Движение.Цена = ТекСтрокаСписокЦен.Цена;

```

КонецЦикла;
КонецПроцедуры

Листинг В.7 – Модуль HTTP-сервиса «Веб-интерфейс»

Функция АвторизацияPOST(Запрос)

```

    УстановитьПривилегированныйРежим(Истина);

    ЗаписьЖурналаРегистрации("POST",,,, "POST");
    Ответ = Новый HTTPСервисОтвет(200);

    CookieСтр = Запрос.Заголовки.Получить("cookie");

    Если CookieСтр<>Неопределено Тогда
        Cookie = ПолучитьCookie(CookieСтр);
    Иначе
        Cookie = Новый Соответствие();
    КонецЕсли;

    Тип = Запрос.ПараметрыЗапроса.Получить("typerequest");
    Тело = Запрос.ПолучитьТелоКакСтроку();

    Чтение = Новый ЧтениеJSON;
    Чтение.УстановитьСтроку(Тело);
    ДанныеЗапроса = ПрочитатьJSON(Чтение);
    Чтение.Закрыть();

    ДанныеОтвета = Новый Структура;

    recipientKey = Запрос.Заголовки.Получить("recipientKey");

    Если Тип = "login" Тогда

        ДанныеОтвета = Авторизация.Логин(ДанныеЗапроса.email,
ДанныеЗапроса.password);

        ИначеЕсли Тип = "refresh" Тогда
            ДанныеОтвета =
Авторизация.ПолучитьТокен(Cookie.Получить("refresh"));
            Если ЗначениеЗаполнено(ДанныеОтвета.error) Тогда
                Ответ = Новый HTTPСервисОтвет(401);
            КонецЕсли;

        ИначеЕсли Тип = "getCode" Тогда
            ДанныеОтвета = Авторизация.ПолучитьКод(ДанныеЗапроса.userID);

        ИначеЕсли Тип = "getKeyPassword" Тогда
            ДанныеОтвета =
Авторизация.ПолучитьКлючСменыПароля(ДанныеЗапроса.userID, ДанныеЗапроса.request,
ДанныеЗапроса.code);

        ИначеЕсли Тип = "passwordChange" Тогда
            ДанныеОтвета =
Авторизация.УстановитьПароль(ДанныеЗапроса.passwordChangeKey,ДанныеЗапроса.passw
ord);

        Иначе
            access = Запрос.Заголовки.Получить("authorization");

            СтруктураПользователя = Авторизация.ПолучитьПользователя(access);

```

```

        Если ЗначениеЗаполнено (СтруктураПользователя.Ошибка) Тогда
            Ответ = Новый HTTPСервисОтвет(401);
            ДанныеОтвета = Новый
Структура ("Ошибка", СтруктураПользователя.Ошибка);
        Иначе
            Пользователь = СтруктураПользователя.Код;
            ПараметрыСеанса.ТекущийПользователь = Пользователь;
        КонецЕсли;

    КонецЕсли;

    Запись = Новый ЗаписьJSON;
    Запись.УстановитьСтроку();
    ЗаписатьJSON(Запись, ДанныеОтвета);
    ДанныеJSON = Запись.Закрыть();

    Ответ.Заголовки.Вставить("Content-Type", "text/text; charset=UTF-8");
    Ответ.Заголовки.Вставить("Content-Type", "application/json");
    Ответ.Заголовки.Вставить("Access-Control-Allow-Origin",
"http://localhost:3000");
    Ответ.Заголовки.Вставить("Access-Control-Allow-Methods", "GET, PUT, POST,
DELETE, HEAD, OPTIONS");
    Ответ.Заголовки.Вставить("Access-Control-Allow-Credentials", "true");
    Ответ.Заголовки.Вставить("Access-Control-Allow-Headers", "X-Requested-With,
authorization, origin, content-type, accept");

    refresh = "";
    Если ДанныеОтвета.Свойство("refresh", refresh) Тогда
        Ответ.Заголовки.Вставить("Set-Cookie", "refresh='" + refresh + "'";
HttpOnly");
    КонецЕсли;

    Ответ.УстановитьТелоИзСтроки(ДанныеJSON);
    Возврат Ответ;

КонецФункции

Функция ПолучитьCookie(стрCookies)

    Массив = стрCookies;
    Результат = Новый Соответствие();
    Для Каждого Элемент ИЗ Массив Цикл
        КлючЗначение = StrSplit(Элемент, "=");
        Значение = КлючЗначение[1];
        Результат.Вставить(Ключ, Значение);
    Конеццикла;
    Возврат Результат;

КонецФункции

Функция ItemsAllGET(Запрос)
    Ответ = Новый HTTPСервисОтвет(200);
    СтрокаJSON = ОбщегоНазначения.ПолучитьНоменклатуру();
    Ответ.УстановитьТелоИзСтроки(СтрокаJSON);
    Возврат Ответ;
КонецФункции

Функция ПередачаТехникиМастеруPOST(Запрос)
    Ответ = api.Добавить(ПолучитьСтруктуруОтвета(), Запрос);
    Возврат СформироватьОтвет(Ответ.Данные, Ответ.Код);
КонецФункции

Функция ПолучитьСтруктуруОтвета()

```

```

        СтруктураОтвета = Новый Структура;
        СтруктураОтвета.Вставить("dataOrder", Новый Структура("masterId,
orderCode, deviceId, comment, problem"));
        СтруктураОтвета.Вставить("results", Новый Массив);
        СтруктураОтвета.Вставить("errors", Новый Структура("exist, title", false,
""));
        Возврат СтруктураОтвета;
КонiecФункции

Функция СформироватьОтвет(СтруктураОтвета, КодОтвета)
    Ответ = Новый HTTPСервисОтвет(КодОтвета);
    Ответ.УстановитьТелоИзСтроки(СформироватьJSON(СтруктураОтвета),
КодировкаТекста.UTF8, ИспользованиеByteOrderMark.НеИспользовать);
    Возврат Ответ;
КонiecФункции

Функция СформироватьJSON(СтруктураДанных)
    Попытка
        ЗаписьJSON = Новый ЗаписьJSON;
        ЗаписьJSON.УстановитьСтроку(Новый ПараметрыЗаписиJSON(,
Символы.Таб));

        JsonSerializerSettings = New JsonSerializerSettings;
        JsonSerializerSettings.DateWritingVariant =
JSONDateWritingVariant.LocalDateWithOffset;
        JsonSerializerSettings.DateSerializationFormat = JSONDateFormat.ISO;

        ЗаписатьJSON(ЗаписьJSON, СтруктураДанных, JsonSerializerSettings);

        Возврат ЗаписьJSON.Закрыть();
    Исключение
        ЗаписьЖурналаРегистрации("Vue",,,, ОписаниеОшибки());
        ЧистыйОтвет = ПолучитьСтруктуруОтвета();
        ОбщегоНазначения.ДобавитьОшибку(ЧистыйОтвет, ОписаниеОшибки());
        Возврат СформироватьJSON(ЧистыйОтвет);
    КонiecПопытки;
КонiecФункции

Функция ItemsDELETE(Запрос)
    Ответ = Новый HTTPСервисОтвет(200);
    Возврат Ответ;
КонiecФункции

Функция ПолучитьСтруктуруОтветаВозвратТехники()
    СтруктураОтвета = Новый Структура;
    СтруктураОтвета.Вставить("data", Новый Структура("comment, result,
paymentId, deviceId, problem, orderId"));
    СтруктураОтвета.Вставить("results", Новый Массив);
    СтруктураОтвета.Вставить("errors", Новый Структура("exist, title", false,
""));

    Возврат СтруктураОтвета;
КонiecФункции

Функция ВозвратТехникиPOST(Запрос)
    Ответ = api.ДобавитьВыдачуТехники(ПолучитьСтруктуруОтветаВозвратТехники(),
Запрос);
    Возврат СформироватьОтвет(Ответ.Данные, Ответ.Код);
КонiecФункции

Функция РеализацияМатериаловPOST(Запрос)
    Ответ =
api.ДобавитьРеализациюМатериалов(ПолучитьСтруктуруОтветаРеализация(), Запрос);

```

```

        Возврат СформироватьОтвет(Ответ.Данные, Ответ.Код);
КонецФункции

Функция ПолучитьСтруктуруОтветаРеализация()
    СтруктураОтвета = Новый Структура;
    ТЧ = Новый Структура("itemId, count");
    Данные = Новый Структура("managerId, clientId, stockId");
    Данные.Вставить("tab", ТЧ);
    СтруктураОтвета.Вставить("data", Данные);
    СтруктураОтвета.Вставить("results", Новый Массив);
    СтруктураОтвета.Вставить("errors", Новый Структура("exist, title", false,
""));

    Возврат СтруктураОтвета;
КонецФункции

```

Листинг В.8 – Общий модуль «Авторизация»

```

Функция Логин(Пользователь, Пароль) Экспорт

    Ответ = Новый Структура("error, access, refresh");

    СтруктураПользователя = ПолучитьПользователяПоКоду(Пользователь);

    Если ЗначениеЗаполнено(СтруктураПользователя.Ошибка) Тогда
        Ответ.error = СтруктураПользователя.Ошибка;
        Возврат Ответ;
    КонецЕсли;

    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
|         ПаролиПользователей.Пользователь
| ИЗ
|         РегистрСведений.ПаролиВПользователей КАК ПаролиПользователей
| ГДЕ
|         ПаролиПользователей.Пользователь = &Пользователь
|         И ПаролиВнешнихПользователей.Пароль = &Пароль";

    Запрос.УстановитьПараметр("Пользователь",
СтруктураПользователя.Пользователь);
    Запрос.УстановитьПараметр("Пароль", Пароль);

    Выборка = Запрос.Выполнить().Выбрать();

    Если Выборка.Следующий() Тогда

        Токены = Токены(СтруктураПользователя.Пользователь, Пользователь);
        Ответ.access = Токены.access;
        Ответ.refresh = Токены.refresh;

    Иначе
        Ответ.error = "Неверное имя пользователя или пароль";
    КонецЕсли;

    Возврат Ответ;

КонецФункции

Функция Токены(Пользователь, ИмяПользователя)

    МаксимальноеКоличествоТокенов = 3;

```

```

    Результат = Новый Структура("access, refresh");

    Набор =
    РегистрыСведений.КлючиАвторизацииПользователей.СоздатьНаборЗаписей();
    Набор.Отбор.Пользователь.Установить(Пользователь);
    Набор.Прочитать();

    Токены = Набор.Выгрузить();
    Токены.Сортировать("ДатаФормирования убыв");
    Набор.Очистить();

    Для Каждого Строка ИЗ Токены Цикл

        МаксимальноеКоличествоТокенов = МаксимальноеКоличествоТокенов - 1;

        Если МаксимальноеКоличествоТокенов = 0 Тогда
            Прервать;
        КонецЕсли;

        ЗаполнитьЗначенияСвойств(Набор.Добавить(), Строка)

    КонецЦикла;

    jti = Строка(Новый УникальныйИдентификатор);
    access = КлючДоступа(ИмяПользователя, jti, "access");
    refresh = КлючДоступа(ИмяПользователя, jti, "refresh");

    НоваяСтрока = Набор.Добавить();
    НоваяСтрока.Пользователь = Пользователь;
    НоваяСтрока.ДатаФормирования = ТекущаяДата();
    НоваяСтрока.Ключ = jti;

    Набор.Записать();

    Результат.access = access;
    Результат.refresh = refresh;
    Возврат Результат;

КонецФункции

```

Листинг В.9 – Общий модуль «Арі»

```

Функция Добавить(СтруктураОтвета, ЗапросНТТР) Экспорт
    Ответ = Новый Структура("Данные, Код", СтруктураОтвета, 200);

    ДанныеJSON = ЗапросНТТР.ПолучитьТелоКакСтроку();
    Данные = ОбщегоНазначения.ОбработатьJSON(СтруктураОтвета, ДанныеJSON);
    Если НЕ Данные = Неопределено Тогда
        Попытка
            НачатьТранзакцию();
            НазначитьМастера(СтруктураОтвета, Данные);
            ЗафиксироватьТранзакцию();
        Исключение
            Если ТранзакцияАктивна() Тогда
                ОтменитьТранзакцию();
            КонецЕсли;
            ОбщегоНазначения.ДобавитьОшибку(СтруктураОтвета,
ОписаниеОшибки());
            Ответ.Код = 400;
        КонецПопытки;
    КонецЕсли;

```

```

Иначе
    Ответ.Код = 400;
КонецЕсли;

Возврат Ответ;
КонецФункции

Процедура НазначитьМастера (СтруктураОтвета, Данные) Экспорт

    УстановитьПривилегированныйРежим (Истина) ;

    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
        | ПоступлениеТехникиОтКонтрагента.Ссылка КАК Ссылка,
        | ПоступлениеТехникиОтКонтрагента.Номер КАК Номер,
        | ПоступлениеТехникиОтКонтрагента.Дата КАК Дата,
        | ПоступлениеТехникиОтКонтрагента.Организация КАК
Организация,
        | ПоступлениеТехникиОтКонтрагента.Клиент КАК Клиент,
        | ПоступлениеТехникиОтКонтрагента.Склад КАК Склад,
        | ПоступлениеТехникиОтКонтрагента.Кладовщик КАК Кладовщик,
        | ПоступлениеТехникиОтКонтрагента.Менеджер КАК Менеджер
        | ИЗ
        | Документ.ПоступлениеТехникиОтКонтрагента КАК
ПоступлениеТехникиОтКонтрагента
        | ГДЕ
        | ПоступлениеТехникиОтКонтрагента.Номер = &Заявка";
    Запрос.УстановитьПараметр ("Заявка", Данные["orderCode"]);
    Результат = Запрос.Выполнить ();
    Выборка = Результат.Выбрать ();

    НовДок = Документы.ПередачаТехникиМастеру.СоздатьДокумент ();
    НовДок.Дата = ТекущаяДата ();

    Пока Выборка.Следующий () Цикл
        НовДок.Организация = Выборка.Организация;
        НовДок.Заявка = Выборка.Ссылка;
        НовДок.Клиент = Выборка.Клиент;
        НовДок.Склад = Выборка.Склад;
        НовДок.Кладовщик = Выборка.Кладовщик;
        НовДок.ДатаЗаказа = Выборка.Дата;
        НовДок.Менеджер = Выборка.Менеджер;
    КонецЦикла;

    НовДок.Мастер = Справочники.Сотрудники.ПолучитьСсылку (Новый
УникальныйИдентификатор (Данные["masterId"]));

    НовДок.УстановитьНовыйНомер ();

    НовСтр = НовДок.ПереченьНоменклатуры.Добавить ();
    НовСтр.Номенклатура = Справочники.Номенклатура.ПолучитьСсылку (Новый
УникальныйИдентификатор (Данные["deviceId"]));
    НовСтр.Статус = Перечисления.СписокСтатусов.ПереданМастеру;
    НовСтр.Комментарии = Данные["comment"];
    НовСтр.ЗаявленнаяНеисправность = Данные["problem"];

    НовДок.Записать (РежимЗаписиДокумента.Проведение);
КонецПроцедуры

Функция ДобавитьВыдачуТехники (СтруктураОтвета, ЗапросНТТР) Экспорт
    Ответ = Новый Структура ("Данные, Код", СтруктураОтвета, 200);

    ДанныеJSON = ЗапросНТТР.ПолучитьТелоКакСтроку ();

```

```

Данные = ОбщегоНазначения.ОбработатьJSON(СтруктураОтвета, ДанныеJSON);
Если НЕ Данные = Неопределено Тогда
    Попытка
        НачатьТранзакцию();
        ВозвратТехники(СтруктураОтвета, Данные);
        ЗафиксироватьТранзакцию();
    Исключение
        Если ТранзакцияАктивна() Тогда
            ОтменитьТранзакцию();
        КонецЕсли;
        ОбщегоНазначения.ДобавитьОшибку(СтруктураОтвета,
ОписаниеОшибки());
        Ответ.Код = 400;
    КонецПопытки;

Иначе
    Ответ.Код = 400;
КонецЕсли;

Возврат Ответ;
КонецФункции

Процедура ВозвратТехники(СтруктураОтвета, Данные) Экспорт

    УстановитьПривилегированныйРежим(Истина);

    Запрос = Новый Запрос;
    Запрос.Текст = "ВЫБРАТЬ
        | ПередачаТехникиМастеру.Заявка КАК Заявка,
        | ПередачаТехникиМастеру.ДатаЗаказа КАК ДатаЗаказа,
        | ПередачаТехникиМастеру.Менеджер КАК Менеджер,
        | ПередачаТехникиМастеру.Склад КАК Склад,
        | ПередачаТехникиМастеру.Кладовщик КАК Кладовщик,
        | ПередачаТехникиМастеру.Клиент КАК Клиент,
        | ПередачаТехникиМастеру.Мастер КАК Мастер,
        | ПередачаТехникиМастеру.Организация КАК Организация
        | ИЗ
        | Документ.ПередачаТехникиМастеру КАК
ПередачаТехникиМастеру
        | ГДЕ
        | ПередачаТехникиМастеру.Заявка = &Заявка";

    СсылкаНаДокумент =
Документы.ПоступлениеТехникиОтКонтрагента.ПолучитьСсылку(Новый
УникальныйИдентификатор(Данные["orderId"]));
    Запрос.УстановитьПараметр("Заявка", СсылкаНаДокумент);
    Результат = Запрос.Выполнить();
    Выборка = Результат.Выбрать();

    НовДок = Документы.ВозвратТехникиКонтрагенту.СоздатьДокумент();
    НовДок.Дата = ТекущаяДата();

    Пока Выборка.Следующий() Цикл
        НовДок.Организация = Выборка.Организация;
        НовДок.Заявка = Выборка.Заявка;
        НовДок.Клиент = Выборка.Клиент;
        НовДок.Склад = Выборка.Склад;
        НовДок.Кладовщик = Выборка.Кладовщик;
        НовДок.ДатаЗаявки = Выборка.ДатаЗаказа;
        НовДок.Менеджер = Выборка.Менеджер;
        НовДок.Мастер = Выборка.Мастер;
    КонецЦикла;

    НовДок.УстановитьНовыйНомер();

```

```

        НовСтр = НовДок.ПереченьНоменклатуры.Добавить ();
        НовСтр.Номенклатура = Справочники.Номенклатура.ПолучитьСсылку (Новый
УникальныйИдентификатор (Данные ["deviceId"]));
        НовСтр.Результат = Данные ["result"];
        НовСтр.Статус = Перечисления.СписокСтатусов.Получен;
        НовСтр.Комментарии = Данные ["comment"];
        НовСтр.ЗаявленнаяНеисправность = Данные ["problem"];
        СчетНаОплату = Документы.РеализацияМатериалов.ПолучитьСсылку (Новый
УникальныйИдентификатор (Данные ["paymentId"]));
        Если Не СчетНаОплату.Пустая () Тогда
            НовСтр.ДоговорНаОплату = СчетНаОплату;
            ЗапросОплата = Новый Запрос;
            ЗапросОплата.Текст = "ВЫБРАТЬ
                | РеализацияМатериалов.Всего КАК Всего
                | ИЗ
                | Документ.РеализацияМатериалов КАК
РеализацияМатериалов
                | ГДЕ
                | РеализацияМатериалов.Ссылка = &Ссылка";
            ЗапросОплата.УстановитьПараметр ("Ссылка", СчетНаОплату);;
            РезультатОплата = ЗапросОплата.Выполнить ();
            ВыборкаОплата = РезультатОплата.Выбрать ();

            Пока ВыборкаОплата.Следующий () Цикл
                НовСтр.Стоимость = ВыборкаОплата.Всего;
            КонечЦикла;
        КонечЕсли;

        НовДок.Записать (РежимЗаписиДокумента.Проведение);
    КонечПроцедуры

Функция ДобавитьРеализациюМатериалов (СтруктураОтвета, ЗапросНТТР) Экспорт
    Ответ = Новый Структура ("Данные, Код", СтруктураОтвета, 200);

    ДанныеJSON = ЗапросНТТР.ПолучитьТелоКакСтроку ();
    Данные = ОбщегоНазначения.ОбработатьJSON (СтруктураОтвета, ДанныеJSON);

    Если НЕ Данные = Неопределено Тогда
        Попытка
            НачатьТранзакцию ();
            Реализация (СтруктураОтвета, Данные);
            ЗафиксироватьТранзакцию ();
        Исключение
            Если ТранзакцияАктивна () Тогда
                ОтменитьТранзакцию ();
            КонечЕсли;
            ОбщегоНазначения.ДобавитьОшибку (СтруктураОтвета,
ОписаниеОшибки ());
            Ответ.Код = 400;
        КонечПопытки;

    Иначе
        Ответ.Код = 400;
    КонечЕсли;

    Возврат Ответ;
КонечФункции

Процедура Реализация (СтруктураОтвета, Данные) Экспорт
    УстановитьПривилегированныйРежим (Истина);

```

```

НовДок = Документы.РеализацияМатериалов.СоздатьДокумент();
НовДок.Дата = ТекущаяДата();

НовДок.Организация = Справочники.Организация.Сервис;
НовДок.Контрагент = Справочники.Контрагенты.ПолучитьСсылку(Новый
УникальныйИдентификатор(Данные["clientId"]));
НовДок.Склад = Справочники.Склады.ПолучитьСсылку(Новый
УникальныйИдентификатор(Данные["stockId"]));
НовДок.Кладовщик = Справочники.Склады.ПолучитьСсылку(Новый
УникальныйИдентификатор(Данные["stockId"])).Ответственный;
НовДок.Менеджер = Справочники.Сотрудники.ПолучитьСсылку(Новый
УникальныйИдентификатор(Данные["managerId"]));

НовДок.УстановитьНовыйНомер();
СуммаВсего = 0;

Для каждого Материал Из Данные["tab"] Цикл
    НовСтр = НовДок.ПереченьНоменклатуры.Добавить();
    мНоменклатура = Справочники.Номенклатура.ПолучитьСсылку(Новый
УникальныйИдентификатор(Материал["itemId"]));
    НовСтр.Номенклатура = мНоменклатура;
    НовСтр.Количество = Материал["count"];
    Если (мНоменклатура.ВидНоменклатуры =
Перечисления.ВидыНоменклатуры.Услуга) Тогда
        НовСтр.Цена = ПолучитьЦену(НовДок.Дата, мНоменклатура);
    Иначе
        НовСтр.Цена = ПолучитьЦену(НовДок.Дата, мНоменклатура) * 1.5;
    КонецЕсли;
    НовСтр.Сумма = НовСтр.Количество * НовСтр.Цена;
    НовСтр.СтавкаНДС = мНоменклатура.ОсновнаяСтавкаНДС;
    НовСтр.СуммаНДС = НовСтр.Сумма * НовСтр.СтавкаНДС.Ставка / 100;
    НовСтр.Всего = НовСтр.Сумма + НовСтр.СуммаНДС;
    СуммаВсего = СуммаВсего + НовСтр.Всего;
КонецЦикла;

НовДок.Всего = СуммаВсего;

НовДок.Записать(РежимЗаписиДокумента.Проведение);
КонецПроцедуры

Функция ПолучитьЦену(АктуальнаяДата, ЭлементНоменклатуры) Экспорт
    Отбор = Новый Структура("Номенклатура", ЭлементНоменклатуры);
    ЗначенияРесурсов = РегистрыСведений.Цены.ПолучитьПоследнее(АктуальнаяДата,
Отбор);
    Возврат ЗначенияРесурсов.Цена;
КонецФункции

```

ПРИЛОЖЕНИЕ Г
ЛИСТИНГ ВЕБ-ИНТЕРФЕЙСА
(рекомендуемое

Листинг Г.1 – Файл data.js

```
import axios from 'axios';
import { API_URL } from '../Constants'

export const _getItems = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_Номенклатура?$format=json`);
export const _getDocPayment = () =>
axios.get(`${API_URL}/odata/standard.odata/Document_РеализацияМатериалов?$format=json`);
export const _getDocInputItems = () =>
axios.get(`${API_URL}/odata/standard.odata/Document_ПоступлениеМатериалов?$format=json`);
export const _getContactsProvider = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_КонтактныеЛицаКонтрагентов?$format=json`);
export const _getContactsProviderRole = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_РолиКонтактныхЛиц?$format=json`);
export const _getDocInputDevices = () =>
axios.get(`${API_URL}/odata/standard.odata/InformationRegister_ПоступлениеТехникИОтКонтрагента_RecordType/SliceLast)?$format=json`);
export const _getDocDevicesMaster = () =>
axios.get(`${API_URL}/odata/standard.odata/InformationRegister_ПередачаТехникиМастеру_RecordType/SliceLast)?$format=json`);
export const _getDocOutputDevices = () =>
axios.get(`${API_URL}/odata/standard.odata/InformationRegister_ВозвратТехникиКонтрагенту_RecordType/SliceLast)?$format=json`);
export const _getUserData = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_Пользователи?$format=json`);
export const _getStocks = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_Склады?$format=json`);
export const _getLeftover = () =>
axios.get(`${API_URL}/odata/standard.odata/AccumulationRegister_ОстаткиМатериалов/Balance)?$format=json`);
export const _getPrice = () =>
axios.get(`${API_URL}/odata/standard.odata/InformationRegister_Цены_RecordType/SliceLast)?$format=json`);
export const _getOrders = () =>
axios.get(`${API_URL}/odata/standard.odata/InformationRegister_Заявки?$format=json`);
export const _getClients = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_Контрагенты?$format=json`);
export const _getStaff = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_Сотрудники?$format=json`);
export const _getPostsStaff = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_Должности?$format=json`);
export const _getNDS = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_СтавкиНДС?$format=json`);
export const _getValuesPropsItems = () =>
axios.get(`${API_URL}/odata/standard.odata/InformationRegister_ЗначенияСвойствНоменклатуры?$format=json`);
export const _getTitlePropsItems = () =>
axios.get(`${API_URL}/odata/standard.odata/ChartOfCharacteristicTypes_ХарактеристикаНоменклатуры?$format=json`);
export const _getAllValuesPropsItems = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_ЗначенияХарактеристикНоменклатуры?$format=json`);
export const _getValuesPropsProviders = () =>
axios.get(`${API_URL}/odata/standard.odata/InformationRegister_ЗначенияСвойствКонтрагента?$format=json`);
```

```

export const _getTitlePropsProviders = () =>
axios.get(`${API_URL}/odata/standard.odata/ChartOfCharacteristicTypes_Характерис
тикаКонтрагента?$format=json`);
export const _getAllValuesPropsProviders = () =>
axios.get(`${API_URL}/odata/standard.odata/Catalog_ЗначенияХарактеристикКонтраге
нта?$format=json`);

export const _createOrderMaster = (dataOrder) =>
axios.post(`${API_URL}/hs/api/doc/master`, dataOrder);
export const _createReturn = (dataOrder) =>
axios.post(`${API_URL}/hs/api/doc/return`, dataOrder);
export const _createPayment = (dataOrder) =>
axios.post(`${API_URL}/hs/api/doc/payment`, dataOrder);
export const _createDocInputDevice = (dataOrder) =>
axios.post(`${API_URL}/hs/api/doc/inputdevice`, dataOrder);
export const _createDocMaster = (dataOrder) =>
axios.post(`${API_URL}/hs/api/doc/setmaster`, dataOrder);
export const _createMaterialsNew = (dataOrder) =>
axios.post(`${API_URL}/hs/api/doc/materials`, dataOrder);
export const _createDocPriceNew = (dataOrder) =>
axios.post(`${API_URL}/hs/api/doc/price`, dataOrder);

```

Листинг Г.2 – Модуль состояния clients.js

```

import {runInAction, makeAutoObservable} from "mobx";
import {
  _getClients,
  _getTitlePropsProviders,
  _getValuesPropsProviders
} from "../../api/http/data";

class Clients {
  items = [];
  valuesProps = [];
  titleProps = [];

  constructor() {
    makeAutoObservable(this);
  }

  getClients = async () => {
    const {data} = await _getClients();

    runInAction(() => {
      this.items = data.value;
    });
  };

  getValuesProps = async () => {
    const {data} = await _getValuesPropsProviders();

    runInAction(() => {
      this.valuesProps = data.value;
    });
  };

  getTitleProps = async () => {
    const {data} = await _getTitlePropsProviders();

    runInAction(() => {
      this.titleProps = data.value;
    });
  };
}

```

```

    };
}

export default new Clients();

```

Листинг Г.3 – Модуль состояния doc.js

```

import {makeAutoObservable, runInAction} from "mobx";
import {
  _createDocPriceNew,
  _createOrderMaster,
  _createPayment,
  _createReturn,
  _getDocPayment
} from "../../api/http/data";

class Doc {
  items = [];
  payment = [];
  newDocSelectMaster = {
    orderCode: '',
    problem: '',
    deviceId: '',
    comment: '',
    masterId: '',
  };

  newReturn = {
    comment: '',
    result: '',
    paymentId: '',
    deviceId: '',
    problem: '',
    orderId: '',
  };

  newPayment = {
    managerId: '',
    clientId: '',
    stockId: '',
    tab: []
  };

  newPrice = {
    tab: []
  };

  loader = false;
  error = '';

  constructor() {
    makeAutoObservable(this);
  }

  changeDocSelectMaster = (data) => {
    this.newDocSelectMaster.orderCode = data.orderCode;
    this.newDocSelectMaster.problem = data.problem;
    this.newDocSelectMaster.deviceId = data.deviceId;
    this.newDocSelectMaster.comment = data.comment;
    this.newDocSelectMaster.masterId = data.masterId;
  };
}

```

```

changeReturn = (data) => {
  this.newReturn.comment = data.comment;
  this.newReturn.result = data.result;
  this.newReturn.paymentId = data.paymentId;
  this.newReturn.deviceId = data.deviceId;
  this.newReturn.problem = data.problem;
  this.newReturn.orderId = data.orderId;
};

changePayment = (data) => {
  this.newPayment.managerId = data.managerId;
  this.newPayment.clientId = data.clientId;
  this.newPayment.stockId = data.stockId;
  data.tab.map((d) => {
    this.newPayment.tab.push({
      itemId: d.itemId,
      count: d.count,
    })
  })
};

changePrice = (data) => {
  data.tab.map((d) => {
    this.newPrice.tab.push({
      itemId: d.itemId,
      count: d.count,
    })
  })
};

createDocSelectMaster = async () => {
  this.loader = true;

  try {
    await _createOrderMaster(this.newDocSelectMaster);
    this.loader = false;
  } catch (e) {
    console.log(e);
    this.loader = false;
  }
};

createPrice = async () => {
  this.loader = true;

  try {
    await _createDocPriceNew(this.newPrice);
    this.loader = false;
  } catch (e) {
    console.log(e);
    this.loader = false;
  }
};

createDocReturn = async () => {
  this.loader = true;

  try {
    await _createReturn(this.newReturn);
    this.loader = false;
  } catch (e) {
    console.log(e);
  }
};

```

```

        this.loader = false;
    }
};

createDocPayment = async () => {
    this.loader = true;

    try {
        await _createPayment(this.newPayment);
        this.loader = false;
    } catch (e) {
        console.log(e);
        this.loader = false;
    }
};

getDocPayment = async () => {
    const { data } = await _getDocPayment();

    runInAction(() => {
        this.payment = data.value;
    });
};

}

export default new Doc();

```

Листинг Г.4 – Модуль состояния docInputDevices.js

```

import {makeAutoObservable, runInAction} from "mobx";
import {
    _createDocInputDevice,
    _getDocInputDevices,
} from "../../api/http/data";

class DocInputDevices {
    items = [];
    newDoc = {
        clientId: '',
        stockId: '',
        managerId: '',
        deviceId: '',
        problem: '',
        comment: '',
    };

    constructor() {
        makeAutoObservable(this);
    }

    changeDoc = (data) => {
        this.newDoc.clientId = data.clientId;
        this.newDoc.stockId = data.stockId;
        this.newDoc.managerId = data.managerId;
        this.newDoc.deviceId = data.deviceId;
        this.newDoc.problem = data.problem;
        this.newDoc.comment = data.comment;
    };

    createDoc = async () => {

```

```

    this.loader = true;

    try {
      await _createDocInputDevice(this.newDoc);
      this.loader = false;
    } catch (e) {
      console.log(e);
      this.loader = false;
    }
  };

  getDocInputDevices = async () => {
    const { data } = await _getDocInputDevices();

    runInAction(() => {
      this.items = data.value;
    });
  };
};

export default new DocInputDevices();

```

Листинг Г.5 – Модуль состояния docInputMaterials.js

```

import {makeAutoObservable, runInAction} from "mobx";
import {_createMaterialsNew, _getDocInputItems} from "../api/http/data";

class DocInputMaterials {
  items = [];

  newDoc = {
    stockId: '',
    providerId: '',
    contactId: '',
    tab: []
  };

  constructor() {
    makeAutoObservable(this);
  }

  changeMaterials = (data) => {
    this.newDoc.stockId = data.stockId;
    this.newDoc.providerId = data.providerId;
    this.newDoc.contactId = data.contactId;
    data.tab.map((d) => {
      this.newDoc.tab.push({
        itemId: d.itemId,
        count: d.count,
        price: d.price,
        sum: d.count * d.price,
      })
    })
  };

  createMaterials = async () => {
    this.loader = true;

    try {
      await _createMaterialsNew(this.newDoc);

```

```

        this.loader = false;
    } catch (e) {
        console.log(e);
        this.loader = false;
    }
};

getDocInputMaterials = async () => {
    const { data } = await _getDocInputItems();

    runInAction(() => {
        this.items = data.value;
    });
};

}

export default new DocInputMaterials();

```

Листинг Г.6 – Модуль состояния items.js

```

import {runInAction, makeAutoObservable} from "mobx";
import {_getAllValuesPropsItems, _getItems, _getTitlePropsItems,
_getValuesPropsItems} from "../../api/http/data";

class Items {
    data = [];
    valuesProps = [];
    titleProps = [];
    allValuesProps = [];

    constructor() {
        makeAutoObservable(this);
    }

    getItems = async () => {
        const { data } = await _getItems();

        runInAction(() => {
            this.data = data.value;
        });
    };

    getValuesProps = async () => {
        const { data } = await _getValuesPropsItems();

        runInAction(() => {
            this.valuesProps = data.value;
        });
    };

    getTitleProps = async () => {
        const { data } = await _getTitlePropsItems();

        runInAction(() => {
            this.titleProps = data.value;
        });
    };

    getAllValuesProps = async () => {
        const { data } = await _getAllValuesPropsItems();
    };
}

```

```

    runInAction(() => {
      this.allValuesProps = data.value;
    });
  };
}

export default new Items();

```

Листинг Г.7 – Компонент LoginPage.js

```

import React from 'react';
import {useFormik} from 'formik';

const validationSchema = yup.object({
  email: yup
    .string('Введите адрес электронной почты')
    .email('Введите корректный адрес электронной почты')
    .required('Введите адрес электронной почты'),
});

const LoginPage = (props) => {

  const dispatch = useDispatch();
  const err = useSelector(state => state.user.err);
  const loggingIn = useSelector(state => state.user.loggingIn);

  let history = useHistory();
  let location = useLocation();

  let {from} = location.state || {from: {pathname: "/"}};

  const sb = () => {
    history.replace(from);
  }

  const formik = useFormik({
    initialValues: {
      email: localStorage.getItem('userID') || '',
      password: ''
    },
    validationSchema: validationSchema,
    onSubmit: (values) => {
      dispatch(login(values.email, values.password, sb))
    },
  });

  const classes = useStyles();

  return (
    <div className={styles.container}>
      <div>
        <div className={styles.banner}>
          <img className={styles.banner_img}
src={require('../assets/banner.jpg')} alt={"banner"}/>
          <img style={{zIndex: '100', position: 'absolute', minHeight: '100%',
width: '100%', top: '0'}}
            src={require('../assets/banner.png')} alt={"banner_png"}/>
          <h2 className={styles.h2}><span className={styles.span}>СЕРВИСНЫЙ
ЦЕНТР</span></h2>

```

```

        <h2 className={styles.h2__2}><span className={styles.span}>ремонт
бытовой техники</span></h2>
      </div>
    </div>
    <div className={styles.box}>
      <img className={styles.logo} src={require('../assets/logo.png')}
alt={"logo"}/>

      <Container component="main" maxWidth="xs">
        <div className={classes.paper}>

          <Typography component="h1" variant="h5">
            Авторизация
          </Typography>

          <form onSubmit={formik.handleSubmit}>
            <TextField
              fullWidth id="email" name="email"
              label="Email"
              helperText={formik.touched.email && formik.errors.email}
              error={formik.touched.email && Boolean(formik.errors.email)}
              value={formik.values.email}
              onChange={formik.handleChange}
            />

            <TextField style={{marginTop: '10px'}}
              fullWidth id="password" type="password"
              name="password" label="Пароль" error={formik.touched.password &&
Boolean(formik.errors.password)}
              helperText={formik.touched.password && formik.errors.password}
              onChange={formik.handleChange} value={formik.values.password}
              className={classes.margin}
            />

            <Button style={{marginTop: '30px'}} color="primary"
variant="contained" fullWidth type="submit" disabled={loggingIn}>
              Войти
            </Button>

            <Button style={{marginTop: '10px'}} color="primary"
              onClick={()=>dispatch(openGetCode(props.history))}>Забыли
пароль?</Button>

            {err && !loggingIn && <Alert severity="error">
              <AlertTitle> {err}</AlertTitle>
            </Alert>}
          </form>
        </div>
      </Container>
    </div>
  </div>
);
};

export default withRouter(LoginPage)

```

Листинг Г.8 – Компонент ManagerPage.js

```

import React, {useEffect, useState} from 'react';
import {observer} from "mobx-react-lite";
import items from "../../redux/ItemsStore/items";

```

```

import userData from "../../redux/UserStore/userData";
import orders from "../../redux/OrdersStore/orders";
import moment from "moment";
import staff from "../../redux/StaffStore/staff";
import clients from "../../redux/ClientsStore/clients";
import {Box, styled, Tab} from "@material-ui/core";
import Tabs from '@mui/material/Tabs';
import {TabContext, TabPanel} from "@material-ui/lab";
import {Table, Tag, Button, Segmented, Input, Space} from 'antd';
import Typography from "@material-ui/core/Typography";
import styles from "../styles/index.css"
import ref from "../../redux/RefStore/ref";
import SetMaster from "../SetMaster";
import ReturnClient from "../ReturnClient";
import {
  AppstoreOutlined,
  PlusOutlined,
  AuditOutlined,
  ProfileOutlined,
  ContactsOutlined,
  TeamOutlined, SearchOutlined
} from '@ant-design/icons';
import DocsPage from "../DocsPage";
import register from "../../redux/RegisterStore/register";
import ItemsPage from "../ItemsPage";
import ClientsPage from "../ClientsPage";
import UsersPage from "../UsersPage";
import AddOrder from "../AddOrder";
import SetStatus from "../SetStatus";

const Manager = observer(() => {

  const [userInfo, setUserInfo] = useState([]);
  const [ordersByUser, setOrdersByUser] = useState([]);
  const [devices, setDevices] = useState([]);
  const [masters, setMasters] = useState([]);
  const [managers, setManagers] = useState([]);
  const [client, setClient] = useState([]);
  const [posts, setPosts] = useState([]);
  const [filters, setFilters] = useState([]);
  const [filtersStatus, setFiltersStatus] = useState([]);
  const [propsItem, setPropsItem] = useState([]);
  const [propsProvider, setPropsProvider] = useState([]);
  const [orderAdd, setOrderAdd] = useState(false);
  const [contactsProvider, setContactsProvider] = useState([]);
  const [providerRoles, setProviderRoles] = useState([]);
  const [nds, setNds] = useState([]);
  const [stockers, setStockers] = useState([]);
  const [price, setPrice] = useState([]);
  const [value, setValue] = React.useState('1');
  const [valueSegm, setValueSegm] = React.useState('orders');
  const [usersAll, setUsersAll] = useState([]);
  const [stocks, setStocks] = useState([]);
  const [statusSelect, setStatusSelect] = useState(false);

  useEffect(() => {
    userData.getUserData();
    staff.getPostsStaff();
    items.getItems();
    items.getValuesProps();
    items.getTitleProps();
    items.getAllValuesProps();
    ref.getContactsProvider();
  });

```

```

    ref.getContactsProviderRole();
    register.getPrice();
    ref.getNDS();
    clients.getValuesProps();
    clients.getTitleProps();
    ref.getStocks();
  }, []);

  useEffect(() => {
    let data = userData.items.filter((item) => item.Code ===
    localStorage.getItem('userID'));
    data.map((i) => {
      setUserInfo({
        code: i.Code,
        fullname: i.Description,
        role: i.Роль,
        idUser: i.СсылкаНаПользователя,
      });
    });
    orders.getOrders();
    clients.getClients();

    let tmp = [];
    userData.items.map((i) => {
      !i.IsFolder && tmp.push({
        id: i.Ref_Key,
        code: i.Code,
        name: i.Description,
        role: i.Роль,
        linkUser: i.СсылкаНаПользователя,
        linkType: i.СсылкаНаПользователя_Type,
      })
    });
    setUsersAll(tmp);

  }, [userData.items]);

  useEffect(() => {
    let tmp = [];
    ref.stocks.map((i) => {
      tmp.push({
        id: i.Ref_Key,
        code: i.Code,
        name: i.Description,
      })
    });
    setStocks(tmp);
  }, [ref.stocks]);

  useEffect(() => {
    let tmp = [];
    ref.nds.map((i) => {
      tmp.push({
        id: i.Ref_Key,
        code: i.Code,
        name: i.Description,
        bid: i.Ставка,
      })
    });
    setNds(tmp);
  }, [ref.nds]);

  useEffect(() => {

```

```

let tmp = [];
ref.contactsProvider.map((i) => {
  tmp.push({
    id: i.Ref_Key,
    providerId: i.Owner_Key,
    code: i.Code,
    name: i.Description,
    role: i.Роль_Key,
    address: i.Адрес,
    phone: i.ТелефонМобильный,
    phoneHome: i.ТелефонРабочий,
    email: i.Email,
    other: i.Другое,
  })
});
setContactsProvider(tmp);
}, [ref.contactsProvider]);

useEffect(() => {
  let tmp = [];
  ref.providerRoles.map((i) => {
    tmp.push({
      id: i.Ref_Key,
      code: i.Code,
      name: i.Description,
    })
  });
  setProviderRoles(tmp);
}, [ref.providerRoles]);

useEffect(() => {
  let data = [];
  let index = 0;

  orders.items.map((d) => {
    d.RecordSet.map((i) => {
      data.push({
        key: index,
        orderId: i.Заявка_Key,
        date: i.Period,
        numOrder: i.НомерЗаказа,
        problem: i.ПричинаОбращения,
        clientId: i.Клиент_Key,
        managerId: i.Менеджер_Key,
        masterId: i.Мастер_Key,
        dateOrder: i.ДатаЗаказа,
        deviceId: i.Устройство_Key,
        status: i.Статус,
        price: i.СтоимостьЗаказа,
        comment: i.Комментарии,
        result: i.Результат,
        check: i.СчетНаОплату,
      });
      index++;
    });
  });

  data = data.filter((item) => item.managerId === userInfo.idUser);

  let dataComparison = data;
  let dataSave = [];

  data.map((d) => {

```

```

let newData = [];
let flag = false;

dataComparison.map((c) => {
  if (d.deviceId === c.deviceId && d.numOrder === c.numOrder) {

    dataSave.length > 0 && dataSave.map((save) => {
      if (save !== undefined) {
        if (save.deviceId === d.deviceId && save.numOrder === d.numOrder)
        {
          flag = true;
        }
      }
    });

    !flag && newData.push(c);
  }
});

newData.length > 0 && newData.sort();
newData.length > 0 && dataSave.push(newData[newData.length - 1]);
});

dataSave.sort((a, b) => b.numOrder - a.numOrder);
setOrdersByUser(dataSave);
}, [orders.items]);

useEffect(() => {
  let data = [];
  let dataProps = [];
  let titleProps = items.titleProps;
  let allValuesProps = items.allValuesProps;

  items.titleProps.map((i) => {
    titleProps.push({
      id: i.Ref_Key,
      name: i.Description,
    });
  });

  items.allValuesProps.map((i) => {
    allValuesProps.push({
      id: i.Ref_Key,
      name: i.Description,
    });
  });

  items.valuesProps.map((i) => {
    dataProps.push({
      deviceId: i.Номенклатура_Key,
      typeProp: getNameById(i.ВидСвойства_Key, titleProps),
      valueProp: getNameById(i.Значение, allValuesProps),
    });
  });

  let index = -1;
  items.data.map((i) => {
    !i.IsFolder && data.push({
      key: index++,
      id: i.Ref_Key,
      code: i.Code,
      name: i.Description,
      type: i.ВидНоменклатуры,
    });
  });
});

```

```

        ndsId: i.ОсновнаяСтавкаНДС_Key,
        article: i.Артикул,
    });
});

setDevices(data);
setPropsItem(dataProps);
}, [items.data]);

useEffect(() => {
    let device = [];
    let status = [];
    ordersByUser.map((d) => {
        device.push({
            text: getNameById(d.deviceId, devices),
            value: d.deviceId,
        })
        status.push({
            text: textStatus(d.status),
            value: d.status,
        })
    })

    setFilters(device);
    setFiltersStatus(status);
}, [ordersByUser]);

useEffect(() => {
    let data = [];
    staff.posts.map((i) => {
        data.push({
            id: i.Ref_Key,
            code: i.Code,
            name: i.Description,
        });
    });

    setPosts(data);
    staff.getStaff();
}, [staff.posts]);

useEffect(() => {
    let data = [];
    let masterId = postsIdByName("Мастер");
    staff.items.map((i) => {
        i.Должность_Key === masterId && data.push({
            id: i.Ref_Key,
            code: i.Code,
            fullname: i.Description,
        });
    });

    setMasters(data);
}, [staff.items]);

useEffect(() => {
    let data = [];
    let managerId = postsIdByName("Менеджер");
    staff.items.map((i) => {
        i.Должность_Key === managerId && data.push({
            id: i.Ref_Key,
            code: i.Code,
            fullname: i.Description,

```

```

    });
  });

  setManagers(data);
}, [staff.items]);

useEffect(() => {
  let tmp = [];
  register.price.map((i) => {
    tmp.push({
      itemId: i.Номенклатура_Key,
      price: i.Цена,
    })
  });
  setPrice(tmp);
}, [register.price]);

useEffect(() => {
  let data = [];
  let staffId = postsIdByName("Кладовщик");
  staff.items.map((i) => {
    i.Должность_Key === staffId && data.push({
      id: i.Ref_Key,
      code: i.Code,
      fullname: i.Description,
    });
  });
});

  setStockers(data);
}, [staff.items]);

useEffect(() => {
  let data = [];
  let index = -1;
  let dataProps = [];
  let titleProps = clients.titleProps;
  let allValuesProps = clients.allValuesProps;

  clients.titleProps.map((i) => {
    titleProps.push({
      id: i.Ref_Key,
      name: i.Description,
    });
  });

  clients.valuesProps.map((i) => {
    dataProps.push({
      clientId: i.Контрагент_Key,
      typeProp: getNameById(i.ВидСвойства_Key, titleProps),
      valueProp: i.Значение,
    });
  });

  clients.items.map((i) => {
    !i.IsFolder && data.push({
      key: index++,
      id: i.Ref_Key,
      code: i.Code,
      name: i.Description,
      fullname: i.ПолноеНаименование,
      type: i.ВидКонтрагента,
    });
  });
});

```

```

    setPropsProvider(dataProps);
    setClient(data);
  }, [clients.items]);

const getNameById = (id, arr) => {
  let rez = arr.filter((item) => item.id === id);
  let name = '';
  rez.map((r) => {
    name = r.name;
  });
  return name;
};

const getRoleByContact = (id) => {
  let rez = contactsProvider.filter((item) => item.id === id);
  let role = providerRoles.filter((item) => item.id === rez[0].role);
  let name = '';
  role.map((r) => {
    name = r.name;
  });
  return name;
};

const getNdsById = (id) => {
  let rez = nds.filter((item) => item.id === id);
  let bid = '';
  rez.map((r) => {
    bid = r.bid;
  });
  return bid;
};

const handleChange = (event, newValue) => {
  setValue(newValue);
};

const nameProp = (id, type) => {
  let rez = propsItem.filter((item) => item.deviceId === id);
  rez = rez.filter((item) => item.typeProp === type);
  let prop = '';

  rez.map((r) => {
    prop = r.valueProp;
  });

  return prop;
};

const namePropProvider = (id, type) => {
  let rez = propsProvider.filter((item) => item.clientId === id);
  rez = rez.filter((item) => item.typeProp === type);
  let prop = '';

  rez.map((r) => {
    prop = r.valueProp;
  });

  return prop;
};

const columns = [
  {

```

```

    title: "Заказ",
    key: 'order',
    sorter: (a, b) => a.numOrder - b.numOrder,
    render: order => <>
      <div style={{fontSize: '12pt', fontWeight:
'600'}}>{order.numOrder}</div>
      <div>{moment(order.dateOrder).format('DD.MM.YYYY')}</div>
    </>
  },
  {
    title: 'Статус',
    key: 'status',
    filters: filtersStatus,
    onFilter: (value, record) => record.status.startsWith(value),
    render: order => <>
      <Tag color={colorTag(order.status)}
key={order.status}>{textStatus(order.status)}</Tag>
      {order.status === "ГотовКВыдаче" || order.status ===
"ГотовКВыдачеБезРемонта" ?
        <div style={{marginTop: '10px'}}><ReturnClient order={order}/></div>
          : order.status !== "Получен" ? <div style={{marginTop:
'10px'}}><Space direction={"vertical"}>
            <Button onClick={() => setStatusSelect(true)}>
              Изменить статус
            </Button>
          </Space></div> : null}
    </>
  ),
},
{
  title: 'Устройство',
  dataIndex: 'deviceId',
  key: 'deviceId',
  render: deviceId => <>
    <div style={{fontSize: '11pt', fontWeight: '600', marginBottom:
'7px'}}>{getNameById(deviceId, devices)}</div>
    <div style={{fontSize: '9pt'}}>{nameProp(deviceId, "Тип устройства")
? ("Тип устройства: " + nameProp(deviceId, "Тип устройства"))
: null}</div>
    <div style={{fontSize: '9pt'}}>{nameProp(deviceId, "Производитель")
? ("Производитель: " + nameProp(deviceId, "Производитель"))
: null}</div>
  </>,
},
{
  title: 'Причина обращения',
  dataIndex: 'problem',
  key: 'problem',
},
{
  title: 'Мастер',
  key: 'masterId',
  render: order => <>
    {order.masterId === "000000000-0000-0000-0000-000000000000"
? <SetMaster
  masters={masters}
  order={order}/>
  : masterNameById(order.masterId)}
  </>,
},
{
  title: 'Клиент',

```

```

    dataIndex: 'clientId',
    key: 'clientId',
    render: clientId => getNameById(clientId, client),
  },
  {
    title: 'Документы',
    key: 'doc',
    render: order => <>
      {order.check ?
        <div><a style={{color: '#215e9a'}} target="_blank"
          href={'http://localhost:8080/checks/' + order.check +
            '.pdf'}>Чек</a></div> : null}
        <div><a style={{color: '#215e9a'}} target="_blank"
          href={'http://localhost:8080/receipts/' + order.numOrder +
            '.pdf'}>Квитанция</a></div>
      </>
    },
  ],
];

return (<>
  <div className='page'>
    <Segmented
      block
      width="100%"
      size="large"
      onChange={(e) => {
        setValueSegm(e)
      }}
      options=[
        {
          label: 'Заказы',
          value: 'orders',
          icon: <AppstoreOutlined/>,
        },
        {
          label: 'Документы',
          value: 'docs',
          icon: <AuditOutlined/>,
        },
        {
          label: 'Номенклатура',
          value: 'itemsList',
          icon: <ProfileOutlined/>,
        },
        {
          label: 'Контрагенты',
          value: 'kontragenti',
          icon: <ContactsOutlined/>,
        },
        {
          label: 'Пользователи',
          value: 'usersList',
          icon: <TeamOutlined/>,
        },
      ],
    </>
  </div>

  <TabContext value={valueSegm}>
    <TabPanel value="usersList" style={{margin: '-20px -20px 0 -20px'}}>
      <UsersPage usersAll={usersAll} getNameById={getNameById}/>
    </TabPanel>
    <TabPanel value="kontragenti" style={{margin: '-20px -20px 0 -20px'}}>

```

```

        <ClientsPage namePropProvider={namePropProvider}
getNameById={getNameById} client={client}
            nameProp={nameProp} contactsProvider={contactsProvider}
            getRoleByContact={getRoleByContact}
getFullnameById={getFullnameById}/>
    </TabPanel>
    <TabPanel value="itemsList" style={{margin: '-20px -20px 0 -20px'}}>
        <ItemsPage getNameById={getNameById} nameProp={nameProp}
devices={devices} getNdsById={getNdsById}/>
    </TabPanel>
    <TabPanel value="docs" style={{margin: '-20px -20px 0 -20px'}}>
        <DocsPage masterNameById={masterNameById} userInfo={userInfo}
devices={devices} nameProp={nameProp}
            client={client} getNameById={getNameById}
filtersStatus={filtersStatus} colorTag={colorTag}
            textStatus={textStatus} managerNameById={managerNameById}
getFullnameById={getFullnameById}
            getNdsById={getNdsById} contactsProvider={contactsProvider}
getRoleByContact={getRoleByContact}
            stockers={stockers} price={price} masters={masters}
statusList={statusList}
            managerId={userInfo.idUser} contacts={contactsProvider}/>
    </TabPanel>
    <TabPanel value="orders" style={{margin: '-20px -20px 0 -20px'}}>
        <div style={{padding: '10px 15px 10px 15px', width: '100%', display:
'grid', gridTemplateColumns: '1fr 25%'}}>
            <div>
                <Typography xs={12} variant="h6" component="h6">
Заказы <Button style={{marginLeft: '15px'}}
                    key="addOrder" icon={}<PlusOutlined/> onClick={()
=> setOrderAdd(true)}>
Добавить
                </Button>
            </Typography>
        </div>
        <div style={{textAlign: 'right', alignItems: 'right'}}>
            <Input
                placeholder="Поиск по номеру заказа"
                prefix={}<SearchOutlined/>
                style={{margin: 0}}
            />
        </div>
    </div>
    <Box sx={{width: '100%', padding: '0 5px 0 5px'}}>
        <Box sx={{bgcolor: '#fff'}}>
            <AntTabs value={value} onChange={handleChange} aria-label="ant
example">
                <AntTab label="Все" value="1"/>
                <AntTab label="В работе" value="2"/>
                <AntTab label="Готовые" value="3"/>
                <AntTab label="Ждут запчастей" value="4"/>
                <AntTab label="На согласовании" value="5"/>
                <AntTab label="Выполненные" value="6"/>
            </AntTabs>
        </Box>

        <TabContext value={value}>
            <TabPanel value="1">
                <Table columns={columns}
                    style={{margin: '-5px -10px -10px -5px'}}
                    expandable={{
                        expandedRowRender: (record) => (<
                            {record.comment} ? (<

```

```

margin: '7px 0 7px 50px',}}>
    <div style={{fontSize: '12pt', fontWeight: '600',
        Комментарии специалистов
    </div>
    <p style={{margin: '0 0 0 50px'}}>
        {record.comment}
    </p>
</>) : null}
{record.result ? (<
    <div style={{fontSize: '12pt', fontWeight: '600',
        marginBottom: '7px',: '7px 0 7px 50px',}}>
        Результаты работы
    </div>
    <p style={{margin: '0 0 0 50px'}}>
        {record.result}
    </p>
</>) : null}
</>),
    rowExpandable: (record) => record.result ||
record.comment, }} dataSource={ordersByUser}/>
</TabPanel>
</TabContext>
</Box>
</Box>
</TabPanel>
</TabContext>

    {orderAdd && <AddOrder modalVisible={orderAdd} devices={devices}
setModalVisible={setOrderAdd} client={client}
        stocks={stocks}/>}
    {statusSelect && <SetStatus modalVisible={statusSelect}
setModalVisible={setStatusSelect} />}
    </div>
    </>);
});

export default Manager;

```

ПРИЛОЖЕНИЕ Д
РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ
(рекомендуемое)

Восстановление пароля

На указанный адрес будет отправлено письмо с кодом подтверждения

Email
kr.belyakova19@gmail.ru

ПОЛУЧИТЬ КОД ПОДТВЕРЖДЕНИЯ

 В системе не зарегистрирована учетная запись, связанная с kr.belyakova19@gmail.ru

Рисунок Д.1 – Ошибка при получении кода подтверждения

Запрос на сброс пароля Входящие x



Сервисный центр <kr.belyakova@mail.ru>
кому: мне ▾

Восстановление пароля

Здравствуйте! Забыли свой пароль от сервисного центра?
Если вы хотите сбросить пароль для kr.belyakova19@gmail.com,
Скопируйте код подтверждения в форму восстановления на сайте.

Код подтверждения: e2e91c

Если вы не запрашивали восстановление пароля, проигнорируйте это письмо.

Рисунок Д.2 – Письмо с кодом подтверждения

Восстановление пароля

Пожалуйста, проверьте электронную почту. Вы получите код подтверждения, который нужно будет ввести здесь, чтобы сбросить пароль

Email
kr.belyakova19@gmail.com

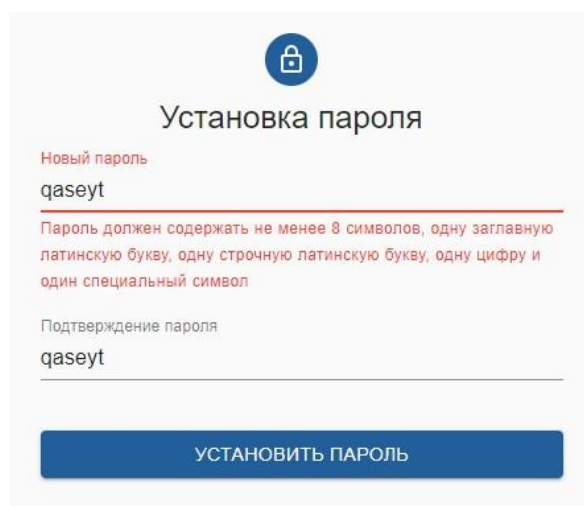
Код подтверждения
qwerty

ПОДТВЕРДИТЬ

ПОВТОРИТЬ ОТПРАВКУ

 Неправильный код подтверждения

Рисунок Д.3 – Неверный код подтверждения



Установка пароля

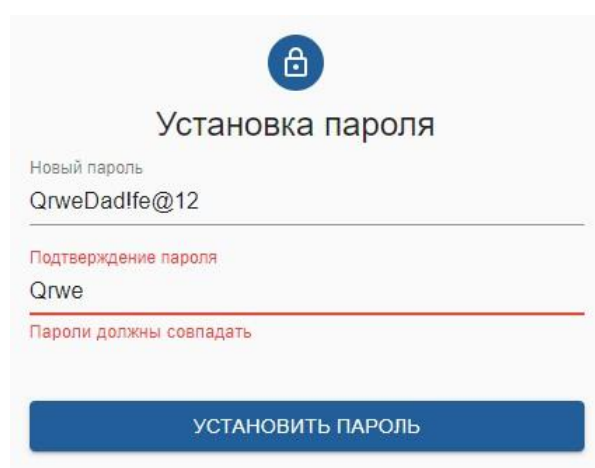
Новый пароль
qaseyt

Пароль должен содержать не менее 8 символов, одну заглавную латинскую букву, одну строчную латинскую букву, одну цифру и один специальный символ

Подтверждение пароля
qaseyt

УСТАНОВИТЬ ПАРОЛЬ

Рисунок Д.4 – Ошибка установки пароля



Установка пароля

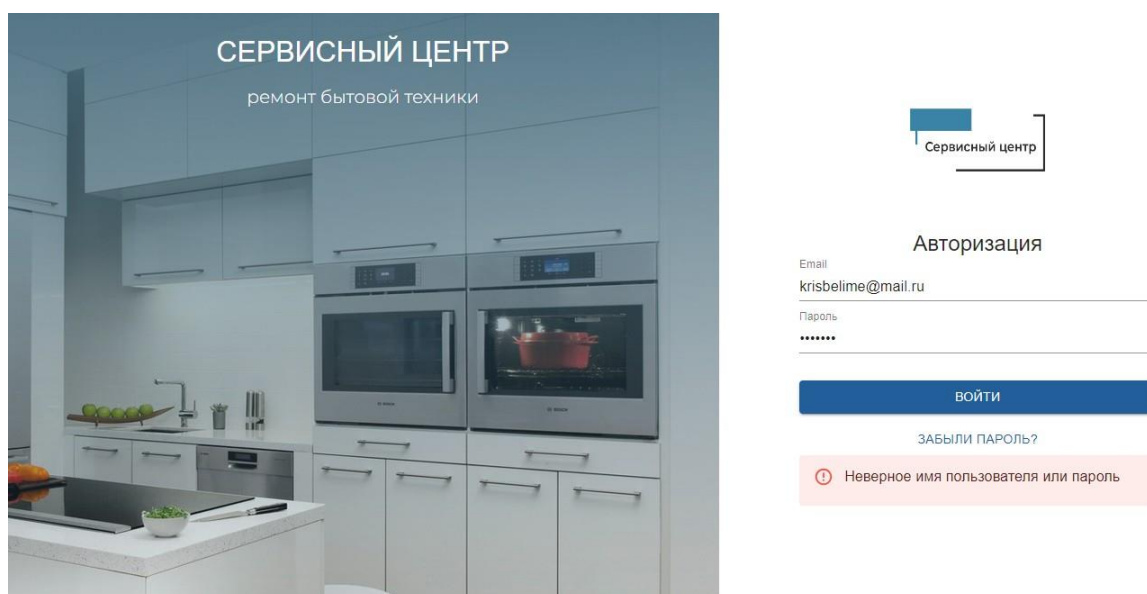
Новый пароль
QrweDadlfe@12

Подтверждение пароля
Qrwe

Пароли должны совпадать

УСТАНОВИТЬ ПАРОЛЬ

Рисунок Д.5 – Сообщение о том, что пароли должны совпадать



СЕРВИСНЫЙ ЦЕНТР
ремонт бытовой техники

Сервисный центр

Авторизация

Email
krisbelime@mail.ru

Пароль

ВОЙТИ

ЗАБЫЛИ ПАРОЛЬ?

Неверное имя пользователя или пароль

Рисунок Д.6 – Попытка авторизации в системе при вводе неверного пароля

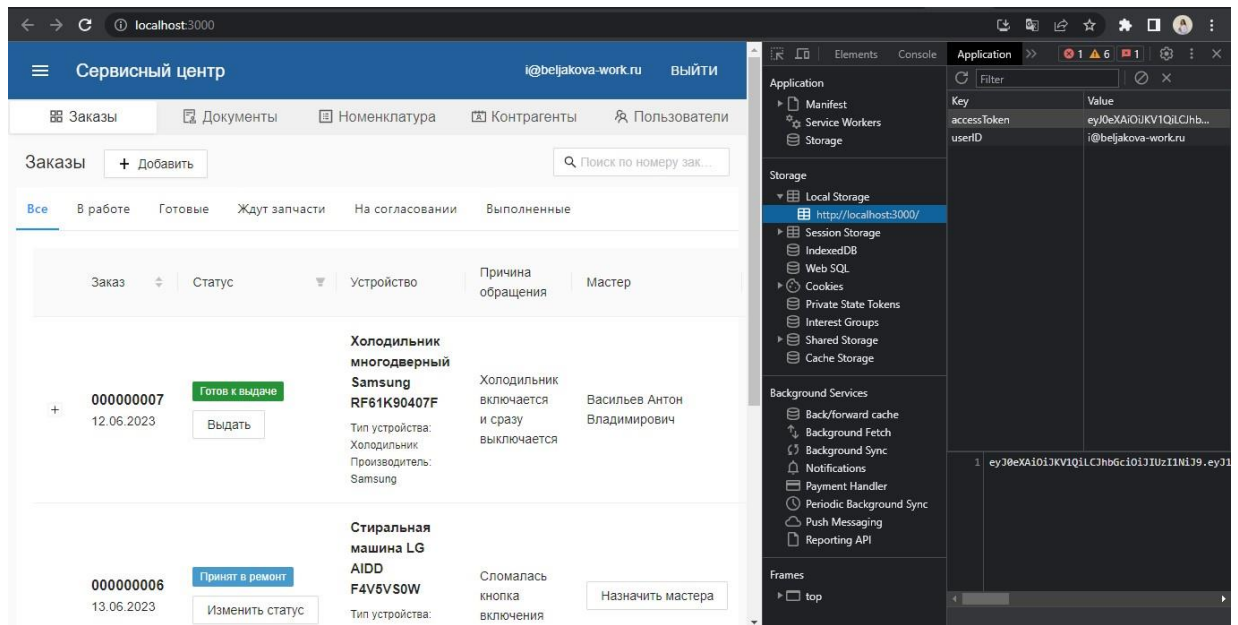


Рисунок Д.7 – Local Storage

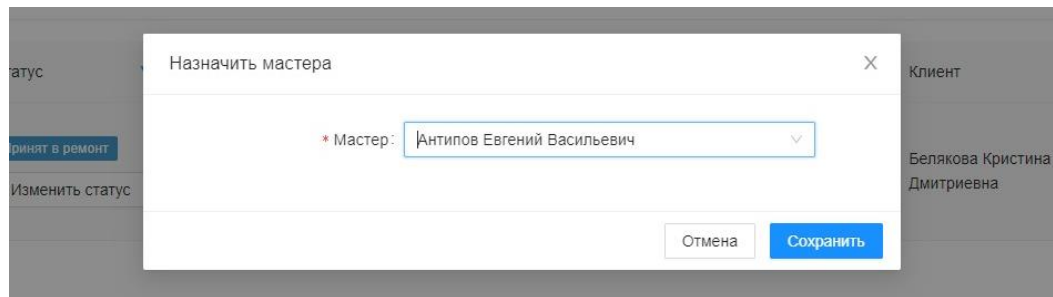


Рисунок Д.8 – Выбор мастера

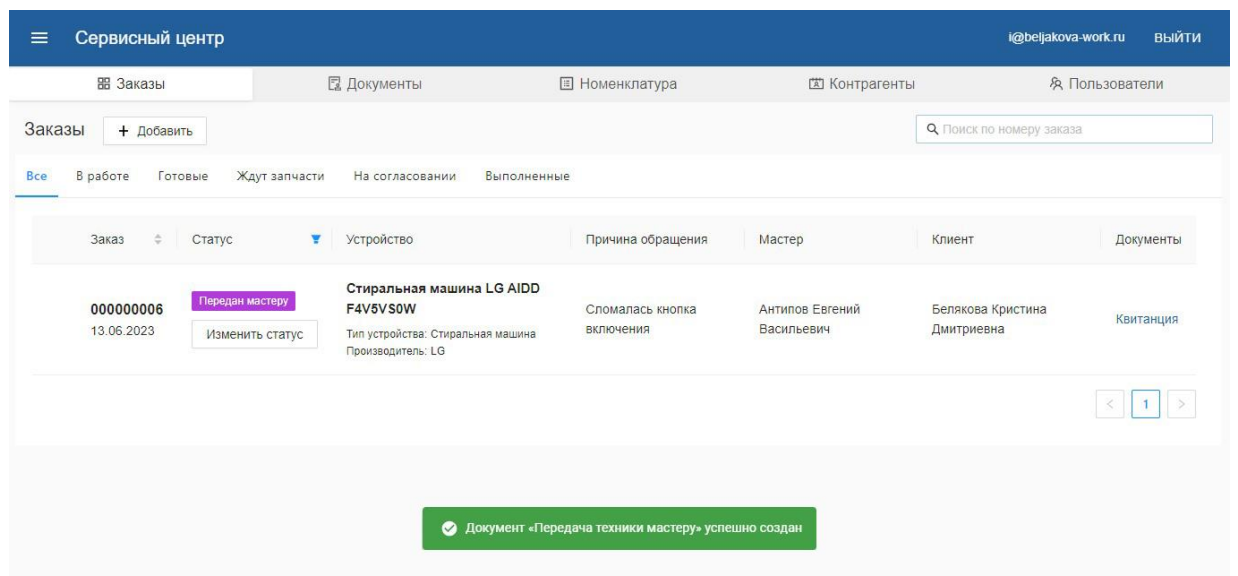


Рисунок Д.9 – Сообщение о том, что документ «Передача техники мастеру» успешно создан

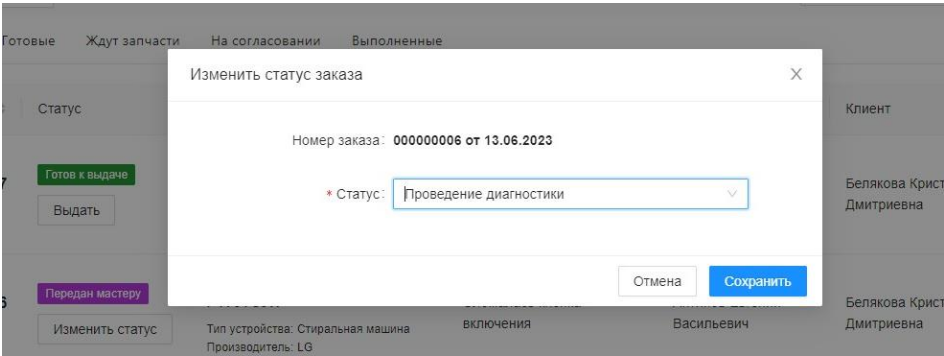


Рисунок Д.10 – Форма изменения статуса заказа

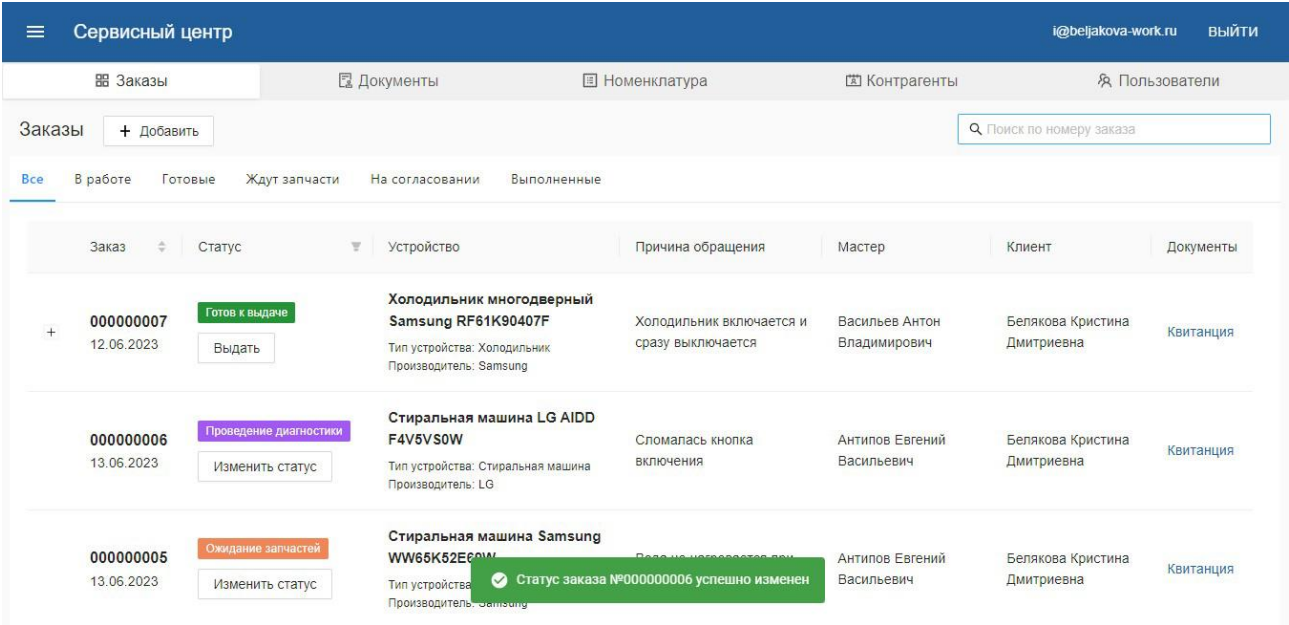


Рисунок Д.11 – Сообщение о том, что статус заказа успешно изменен

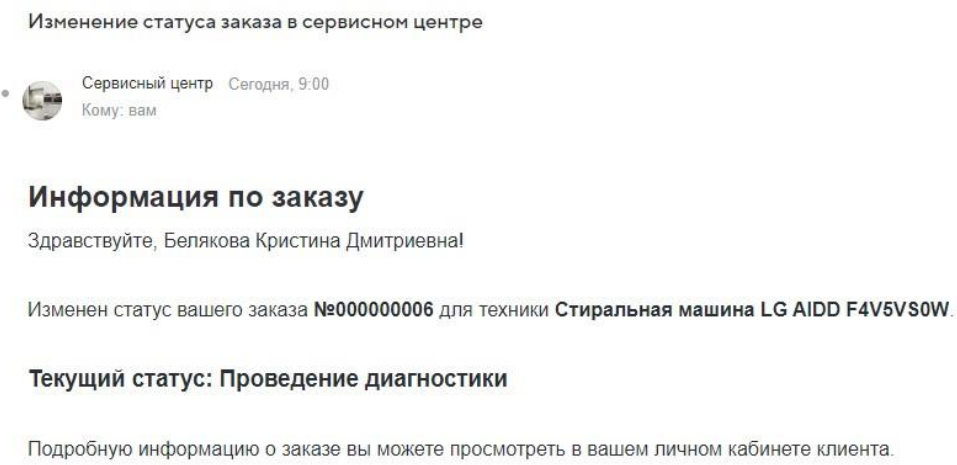


Рисунок Д.12 – Отправленное письмо клиенту о том, что статус заказа изменен

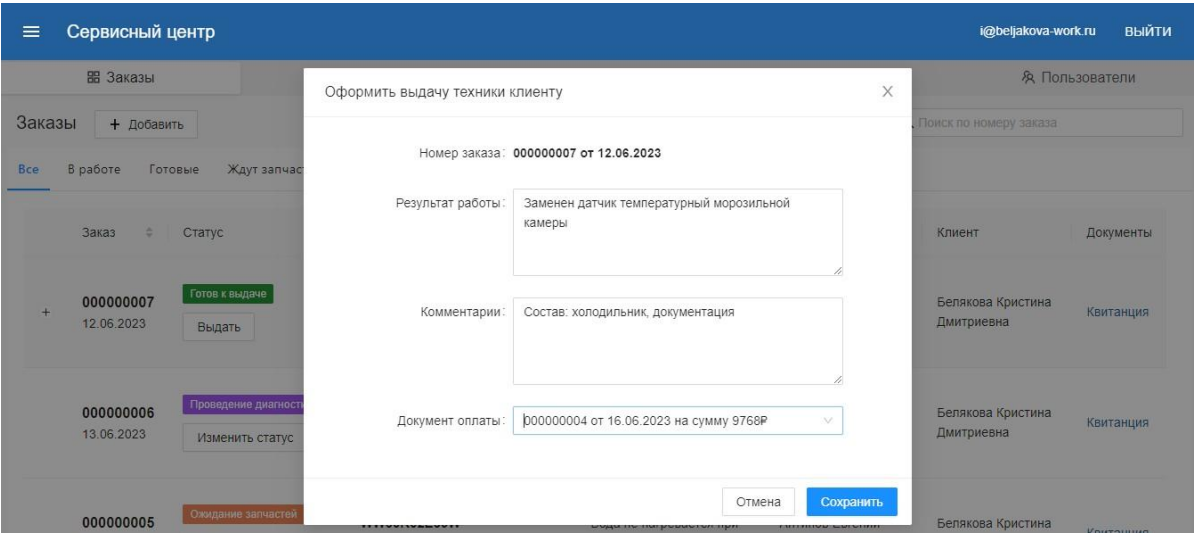


Рисунок Д.13 – Форма документа «Возврат техники контрагенту»

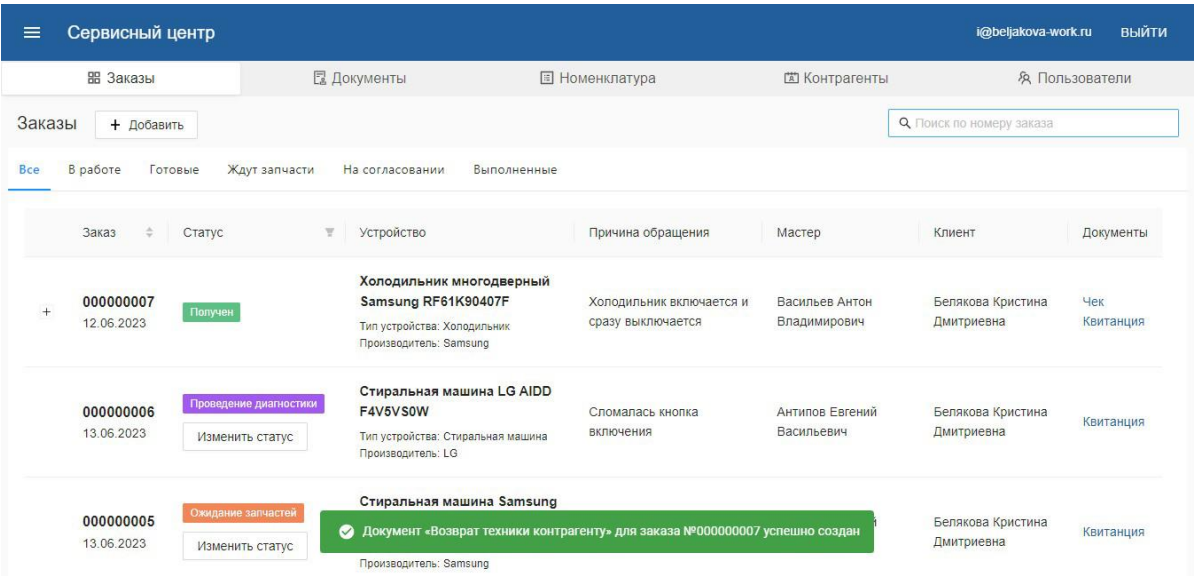


Рисунок Д.14 – Сообщение о том, что документ успешно создан



Рисунок Д.15 – Печатная форма документа «Реализация материалов»

Реализация материалов

Дата: 22.06.2023 09:24:40

* Клиент:

Фролов Артём Андреевич

* Склад:

Основной

ПОДБОР НОМЕНКЛАТУРЫ

Номенклатура	Количество	Цена	Сумма
Ручка двери для холодильника LG GA-B509CCIL	1	6300Р	6300Р
Диагностика	1	500Р	500Р
Замена ручки двери для холодильника	1	2550Р	2550Р

Отмена

Сохранить

Рисунок Д.16 – Создание документа «Реализация материалов»

Заказы

Документы

Номенклатура

Контрагенты

Пользователи

Документы

+ Добавить

Поиск по номеру документа

Поступление техники от клиента

Передача техники мастеру

Возврат техники клиенту

Реализация материалов

Поступление материалов

Цены на номенклатуру

Документ	Склад	Менеджер	Клиент	Стоимость	Чек
<div>000000007</div> <div>22.06.2023</div>	Основной	Иванов Иван Иванович	Фролов Артём Андреевич	9950Р	<div>Открыть</div> <div></div> <div></div>

Номенклатура	Количество	Цена	Сумма	Ставка НДС	Сумма НДС	Всего
Ручка двери для холодильника LG GA-B509CCIL	1	5250Р	5250Р	20%	1050Р	6300Р
Диагностика	1	500Р	500Р	18%	90Р	590Р
Замена ручки двери для холодильника	1	2550Р	2550Р	20%	510Р	3060Р

Итого по документу: 9950Р

Документ «Реализация материалов» успешно создан

Рисунок Д.17 – Сообщение о том, что документ создан

Поступление материалов

Дата: 22.06.2023 10:35:00

* Поставщик:

ООО "САМСУНГ ЭЛЕКТРОНИКС РУС КОМПАНИ"

* Контактное лицо поставщика:

Михайлова Марина Алексеевна

* Склад:

Основной

ПОДБОР НОМЕНКЛАТУРЫ

Материал	Количество	Цена	Сумма
Амортизаторы для стиральной машины Samsung WW65K62E69W	10	1680	16800Р

Отмена

Сохранить

Рисунок Д.18 – Создание документа «Поступление материалов»

The screenshot shows the 'Сервисный центр' (Service Center) web application interface. The top navigation bar includes a menu icon, the title 'Сервисный центр', the user email 'l@beljakova-work.ru', and a 'ВЫЙТИ' (Logout) button. Below the navigation bar are tabs for 'Заказы' (Orders), 'Документы' (Documents), 'Номенклатура' (Nomenclature), 'Контрагенты' (Counterparties), and 'Пользователи' (Users). The 'Документы' tab is active, showing a list of documents. A search bar is present with the text 'Поиск по номеру документа'. Below the search bar are tabs for different document types: 'Поступление техники от клиента', 'Передача техники мастеру', 'Возврат техники клиенту', 'Реализация материалов', 'Поступление материалов' (selected), and 'Цены на номенклатуру'. The main table displays document details for document number '000000004' dated '22.06.2023'. The supplier is 'ООО "САМСУНГ ЭЛЕКТРОНИКС РУС КОМПАНИ"', and the contact is 'Михайлова Марина Алексеевна'. The warehouse is 'Основной', and the warehouse manager is 'Ларин Сергей Владимирович'. The total cost is '16800Р'. Below the document details is a table for the materials received, showing 'Амортизаторы для стиральной машины Samsung WW65K52E69W' with a quantity of '10' and a price of '1680Р', totaling '16800Р'. At the bottom, a green message box states: 'Документ «Поступление материалов» успешно создан' (Document 'Materials Receipt' successfully created).

Рисунок Д.19 – Сообщение о том, что документ «Поступление материалов» успешно создан

The screenshot shows the 'Пользователи (создание)' (Users (creation)) form. The form has tabs for 'Основное' (Main), 'Ключи авторизации пользователей' (User authorization keys), 'Коды подтверждения пользователей' (User confirmation codes), and 'Еще...' (More). The 'Основное' tab is active. The form contains the following fields: 'Почта:' (Email) with the value 'kr.belyakova@yandex.ru', 'Наименование:' (Name) with the value 'Акимова Анастасия Владимировна', 'Ссылка на пользователя:' (User link) with the value 'Акимова Анастасия Владимировна', 'Роль:' (Role) with the value 'Менеджер', and a checkbox for 'Пользователь заблокирован:' (User blocked) which is currently unchecked. There are buttons for 'Записать и закрыть' (Save and close), 'Записать' (Save), 'Отправить приглашение' (Send invitation), and 'Еще...' (More). A help icon (?) is also present.

Рисунок Д.20 – Создание нового пользователя системы

Доступ в систему <http://localhost:3000/>



Сервисный центр kr.belyakova@mail.ru Сегодня в 14:45

Я >

Папка: Входящие

Здравствуйте, Акимова Анастасия Владимировна
Вам предоставлен доступ в систему Сервисный центр
Для начальной установки пароля перейдите по ссылке ниже
<http://localhost:3000/password-change/40d7a830-1573-4730-99e7-ad0406a923fb>

Рисунок Д.21 – Письмо с ссылкой на установку пароля

Ведомость продаж по контрагентам

Сформировать | Выбрать вариант... | Настройки... | Еще ▾

Период: ☒ 01.01.2023 - 30.06.2023

Ведомость продаж по контрагентам				
Параметры: Период: 01.01.2023 - 30.06.2023				
Контрагент	Количество	Себестоимость	Выручка	Прибыль
Номенклатура				
Клиенты	11,000	17 708,00	35 416,00	17 708
Агапов Алексей Владимирович	1,000	2 442,00	4 884,00	2 442
Датчик температурный морозильной камеры холодильника Samsung RF61K90407F	1,000	2 442,00	4 884,00	2 442
Белякова Кристина Дмитриевна	8,000	11 630,00	23 260,00	11 630
Ручка двери для холодильника LG GA-B509CCIL	2,000	5 250,00	10 500,00	5 250
Датчик температурный морозильной камеры холодильника Samsung RF61K90407F	2,000	4 884,00	9 768,00	4 884
Насос (помпа) для стиральной машины LG AIDD F4V5VS0W	4,000	1 496,00	2 992,00	1 496
Фролов Артём Андреевич	2,000	3 636,00	7 272,00	3 636
Ручка двери для холодильника LG GA-B509CCIL	1,000	3 150,00	6 300,00	3 150
Амортизаторы для стиральной машины Samsung WW65K52E69W	1,000	486,00	972,00	486
Итого	11,000	17 708,00	35 416,00	17 708

Рисунок Д.22 – Отчет «Ведомость продаж по контрагентам»

Движение материалов

Сформировать | Выбрать вариант... | Настройки... | Еще ▾

Период: ☒ 01.01.2023 - 30.06.2023

Движения материалов				
Параметры: Склад: Период: 01.01.2023 - 30.06.2023				
Склад	Материал	Начальный остаток	Приход	Расход
Основной				Конечный остаток
	Амортизаторы для стиральной машины Samsung WW65K52E69W	10,000	10,000	1,000
	Двигатель (мотор) вентилятора для холодильника LG GA-B509CCIL	10,000		
	Опорно цепной двигатель в сборе Samsung RF61K90407F	10,000		
	Ручка двери для холодильника LG GA-B509CCIL	10,000		3,000
	Датчик температурный морозильной камеры холодильника Samsung RF61K90407F	10,000		3,000
	Насос (помпа) для стиральной машины LG AIDD F4V5VS0W	10,000		4,000
	Тэн (нагревательный элемент) для стиральной машины Samsung WW65K52E69W	5,000		
	Тэн (нагревательный элемент) для стиральной машины LG AIDD F4V5VS0W	4,000		
Склад №1				
	Насос (помпа) для стиральной машины LG AIDD F4V5VS0W	5,000		
	Опорно цепной двигатель в сборе Samsung RF61K90407F	3,000		

Рисунок Д.23 – Отчет «Движения материалов»

Остатки техники на складе

Сформировать | Выбрать вариант... | Настройки... | Еще ▾

Период: ☒ 01.01.2023 - 30.06.2023

Остатки техники на складе	
Параметры: Период: 01.01.2023 - 30.06.2023	
Кладовщик	Количество
Техника	
Кладовщики	2,000
Анисимов Никита Васильевич	1,000
Стиральная машина LG AIDD F4V5VS0W	1,000
Ларин Сергей Владимирович	1,000
Холодильник многодверный Samsung RF61K90407F	1,000
Итого	2,000

Рисунок Д.24 – Отчет «Остатки техники на складе»