

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт цифровых технологий  
(наименование института полностью)

Департамент бакалавриата  
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Цифровая трансформация бизнеса  
(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Автоматизация учёта заказов»

Обучающийся

Д.В. Штефан

(Инициалы Фамилия)

(личная подпись)

Руководитель

В.В. Гарасова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

## Аннотация

В условиях интенсивного развития рынка и усиления конкурентной борьбы эффективная система управления заказами приобретает стратегическое значение для устойчивого развития компаний. Особую актуальность этот вопрос имеет для креативных индустрий, где требуется оперативное управление множеством разнородных проектов при сохранении высоких стандартов качества. Необходимость автоматизации учетных процессов продиктована потребностью в исключении ошибок ручного ввода, оперативной обработке информации, обеспечении прозрачности операций и повышении стандартов клиентского сервиса.

Настоящее исследование сфокусировано на создании и экономическом обосновании комплексной системы автоматизации учета заказов для рекламного агентства. Важной особенностью проекта стала разработка адаптивных решений, учитывающих специфику работы с творческими проектами и цифровыми продуктами. В рамках проведенного анализа изучены теоретические аспекты автоматизации, идентифицированы недостатки существующей системы ручного учета и разработаны меры по их устранению. Выполнено детальное сравнение современных систем автоматизации, включая решения 1С, Bitrix24, RetailCRM и SAP, с оценкой их функциональных характеристик, экономической эффективности и потенциала масштабирования.

Особое внимание было уделено анализу интеграционных возможностей различных платформ и их совместимости со специализированным программным обеспечением, используемым в рекламной деятельности. Центральное место в работе занял анализ операционных процессов ООО «Диджитал Агентство», в ходе которого выявлены критические точки и проведен SWOT-анализ действующей системы. На основе полученных результатов разработана многоуровневая программная архитектура, интегрирующая клиентские приложения, серверные компоненты, базу данных

и модульную структуру. Созданы специализированные алгоритмы обработки информации, оптимизированная структура базы данных и адаптированные пользовательские интерфейсы для различных категорий пользователей.

При разработке системы учтены особенности workflow в рекламном бизнесе, включая необходимость согласования креативных концепций, управления правками и контроля сроков выполнения работ. Пилотная реализация системы продемонстрировала значительное улучшение ключевых показателей: продолжительность обработки заказов сократилась на 84%, количество ошибок уменьшилось на 87,5%, а индекс клиентской удовлетворенности вырос на 35%. Дополнительным преимуществом стало сокращение времени на согласование проектов и повышение предсказуемости производственных циклов.

Экономическое обоснование проекта подтвердило его высокую инвестиционную привлекательность: расчетный чистый дисконтированный доход достиг 2 362 916,49 руб. при индексе рентабельности 2,29 и сроке окупаемости 1,31 года. Отдельно следует отметить синергетический эффект от внедрения системы, выразившийся в улучшении координации между отделами и повышении прозрачности бизнес-процессов.

Внедренная система автоматизации учета заказов обеспечивает оптимизацию бизнес-процессов, снижение операционных затрат и укрепление конкурентных позиций компании. Дальнейшее развитие системы предполагает интеграцию с инструментами аналитики и бизнес-интеллекта для прогнозирования загрузки производственных мощностей и оптимизации ресурсного планирования. Предложенные решения обладают потенциалом адаптации для различных секторов экономики, требующих эффективного управления заказами и клиентскими отношениями, особенно в сферах с проектной организацией работы и необходимостью управления творческими процессами.

## Оглавление

Введение.....	5
Глава 1. Теоретические основы автоматизации учёта заказов .....	7
1.1. Понятие и сущность учёта заказов .....	7
1.2. Проблемы ручного учёта заказов .....	9
1.3. Автоматизация как решение проблем учёта .....	11
Глава 2. Анализ существующих систем автоматизации учёта заказов.....	13
2.1. Обзор популярных систем автоматизации .....	13
2.2. Сравнительный анализ систем.....	14
2.3. Критерии выбора системы для автоматизации .....	15
Глава 3. Анализ текущих бизнес-процессов учёта заказов.....	16
3.1. Описание текущей системы учёта заказов .....	16
3.2. Выявление «узких мест» .....	20
3.3. SWOT-анализ текущей системы.....	21
Глава 4. Проектирование системы автоматизации учёта заказов.....	23
4.1. Архитектура комплекса программного обеспечения.....	23
4.2. Модульная структура.....	27
4.3. Алгоритмы обработки данных .....	30
4.4. Проектирование структуры базы данных .....	33
4.5. Экранной формы, пользовательские меню.....	47
Глава 5. Внедрение и тестирование системы автоматизации.....	51
5.1. Тестовые данные для проверки работоспособности.....	51
5.2. Пилотное внедрение.....	56
5.3. Результаты тестирования.....	58
Глава 6. Экономическое обоснование автоматизации учёта заказов.....	59
6.1. Затраты на внедрение.....	59
6.2. Экономический эффект.....	60
6.3. Срок окупаемости.....	61
Заключение.....	63
Список используемой литературы и используемых источников.....	64

Приложение А. Алгоритм обработки заявок.....	66
Приложение Б. Алгоритм управления проектами.....	69
Приложение В. Алгоритм уведомлений.....	71
Приложение Г. Интеграция с CRM.....	72
Приложение Д. Генерация отчета по проектам.....	73

## Введение

Современный бизнес сталкивается с необходимостью оперативной обработки заказов в условиях растущей конкуренции и динамично меняющегося рынка. Эффективное управление заказами - один из ключевых факторов успеха компании, напрямую влияющий на удовлетворённость клиентов, оптимизацию ресурсов и финансовые результаты. В этой связи автоматизация учёта заказов становится не просто инструментом оптимизации, а стратегическим решением, позволяющим бизнесу сосредоточиться на развитии и улучшении клиентского опыта.

Актуальность темы обусловлена рядом вызовов, с которыми сталкиваются компании при ручном учёте заказов:

- высокая вероятность ошибок (дублирование данных, потери заказов, неверные адреса доставки);
- задержки в обработке, ведущие к неудовлетворённости клиентов и потере лояльности;
- отсутствие аналитики, что затрудняет принятие управленческих решений;
- рутинные операции, отнимающие время сотрудников, которое могло бы быть направлено на стратегические задачи.

Современные технологии предлагают широкий спектр инструментов для автоматизации учёта заказов - от CRM-систем (Bitrix24, RetailCRM) до ERP-решений (1C, SAP) и облачных сервисов (Google Sheets + Apps Script). Однако выбор подходящего решения требует глубокого анализа существующих бизнес-процессов, оценки функциональных возможностей систем и экономического обоснования внедрения.

Цель данной работы - разработать и обосновать комплексное решение по автоматизации учёта заказов, включающее анализ текущих процессов, проектирование системы, её внедрение и оценку экономической эффективности.

Для достижения поставленной цели решаются следующие задачи:

- изучить теоретические основы автоматизации учёта заказов, включая проблемы ручного учёта и преимущества автоматизации (Глава 1);
- провести анализ существующих систем автоматизации, сравнить их функциональность, стоимость и критерии выбора (Глава 2);
- проанализировать текущие бизнес-процессы учёта заказов, выявить «узкие места» и провести SWOT-анализ (Глава 3);
- разработать архитектуру системы автоматизации, включая модульную структуру, алгоритмы обработки данных и проектирование базы данных (Глава 4);
- осуществить пилотное внедрение и тестирование системы, оценить её работоспособность и эффективность (Глава 5);
- провести экономическое обоснование автоматизации, рассчитать затраты на внедрение, экономический эффект и срок окупаемости (Глава 6);

Объект исследования - процесс учёта заказов в компании. Предмет исследования - методы и инструменты автоматизации учёта заказов.

Методы исследования:

- анализ научной литературы и практических кейсов;
- сравнительный анализ систем автоматизации;
- моделирование бизнес-процессов;
- экономические расчёты (оценка затрат и эффективности).

Практическая значимость работы заключается в разработке готового решения для автоматизации учёта заказов, которое может быть адаптировано под специфику конкретного бизнеса. Внедрение такой системы позволит:

- сократить время обработки заказов на 70–80%;
- минимизировать ошибки до 2–5%;
- повысить удовлетворённость клиентов за счёт оперативности и прозрачности процесса;

## **Глава 1. Теоретические основы автоматизации учёта заказов**

### **1.1. Понятие и сущность учёта заказов**

#### **1.1.1. Определение учёта заказов**

Учёт заказов - это систематический процесс сбора, регистрации, обработки, контроля и анализа информации о заказах клиентов, направленный на обеспечение оперативного и точного выполнения обязательств компании перед покупателями. Этот процесс включает в себя несколько ключевых этапов:

- Регистрация заказа - фиксация данных о клиенте, составе заказа, условиях оплаты и доставки.
- Обработка заказа - проверка наличия товаров на складе, формирование счёта, подтверждение заказа клиенту.
- Отслеживание статуса - контроль за выполнением заказа (сборка, упаковка, отправка, доставка).
- Аналитика и отчётность - формирование данных о продажах, популярных товарах, среднем чеке, времени обработки заказов.

Пример: В интернет-магазине электроники учёт заказов начинается с момента оформления покупки на сайте [3]. Система автоматически фиксирует данные клиента, состав заказа, выбранный способ оплаты и доставки. Далее заказ передаётся на обработку: проверяется наличие товаров на складе, формируется счёт, и клиенту отправляется подтверждение. После оплаты заказ передаётся на сборку и отpravку, а его статус обновляется в личном кабинете клиента. [3]

#### **1.1.2. Роль учёта заказов в бизнес-процессах компании**

Учёт заказов является центральным звеном в цепочке бизнес-процессов, так как напрямую влияет на следующие аспекты деятельности компании:

перативность обработки заказов [3]

быстрая регистрация и обработка заказов позволяют сократить время ожидания клиента, что повышает его лояльность.

пример: В компании, занимающейся доставкой еды, автоматизация учёта заказов позволила сократить время обработки с 20 до 5 минут, что привело к увеличению количества заказов на 30%.

точность и надёжность данных

автоматизированный учёт минимизирует риск ошибок, связанных с человеческим фактором (неверный адрес, количество товара, сумма оплаты).

пример: В розничной сети после внедрения системы автоматизации количество ошибок в заказах сократилось с 12% до 1%.

улучшение качества обслуживания клиентов

клиенты получают возможность отслеживать статус своего заказа в реальном времени, что повышает доверие к компании.

пример: Интернет-магазин одежды внедрил систему уведомлений о статусе заказа, что привело к увеличению повторных покупок на 25%.

аналитика и принятие управленческих решений

автоматизированный учёт позволяет формировать отчёты по продажам, популярным товарам, среднему чеку, что помогает компании оптимизировать ассортимент и маркетинговые стратегии.

пример: Анализ данных о продажах позволил компании выявить наиболее популярные товары и увеличить их запасы на складе, что привело к росту выручки на 15%.

интеграция с другими бизнес-процессами

учёт заказов тесно связан со складским учётом, логистикой, бухгалтерией и маркетингом. Автоматизация позволяет синхронизировать данные между этими отделами, что повышает общую эффективность компании.

пример: Интеграция системы учёта заказов с складским ПО позволила компании сократить время на обработку возвратов на 40%.

### **1.1.3. Сущность учёта заказов в современных условиях**

В условиях цифровизации и роста электронной коммерции учёт заказов претерпевает значительные изменения. Современные компании сталкиваются с необходимостью обработки большого количества заказов в кратчайшие сроки, что невозможно обеспечить без автоматизации. Ключевые аспекты учёта заказов в современных реалиях включают:

- Цифровизация процессов
- Интеграция с онлайн-платформами
- Персонализация и клиентоориентированность
- Аналитика в реальном времени
- Мобильность и доступность

## **1.2. Проблемы ручного учёта заказов**

### **1.2.1. Основные недостатки ручного учёта заказов**

Ручной учёт заказов, несмотря на свою простоту и доступность, имеет ряд существенных недостатков, которые негативно влияют на эффективность бизнес-процессов. Рассмотрим ключевые проблемы, с которыми сталкиваются компании при использовании ручных методов учёта:

высокая вероятность ошибок:

- Дублирование данных: один и тот же заказ может быть зарегистрирован несколько раз из-за ошибок сотрудников.
- Опечатки и неточности: неверно указанные адреса, телефоны или состав заказа.

задержки в обработке заказов:

- Ручной ввод данных занимает значительно больше времени, чем автоматизированный.
- Клиенты вынуждены дольше ждать подтверждения заказа, что снижает их лояльность.

отеря заказов:

- Заказы могут теряться из-за человеческого фактора: забытые записи, несохраненные файлы, утерянные бумажные документы.
- Последствия: Финансовые потери, неудовлетворённость клиентов, ущерб репутации компании.

тсутствие актуальной аналитики:

- Ручной учёт не позволяет оперативно формировать отчёты о продажах, популярных товарах, среднем чеке.
- Отсутствие прогнозирования: невозможно спрогнозировать спрос или оптимизировать складские запасы.

ложности с масштабированием:

- Ручной учёт не справляется с ростом количества заказов.
- Дополнительные затраты: Увеличение штата для обработки растущего потока заказов.

низкая прозрачность процессов:

- Отсутствует единая база данных, что затрудняет контроль за выполнением заказов.
- Сложности с аудитом: невозможно быстро проверить, кто и когда внёс изменения в заказ. [17]

### **1.2.2. Сравнение ручного и автоматизированного учёта заказов**

Для наглядности сравним ключевые параметры ручного и автоматизированного учёта заказов:

Таблица 1 - Сравнение ручного и автоматизированного учёта заказов

Параметр	Ручной учёт	Автоматизированный учёт
Скорость обработки	30-60 минут на заказ	3-5 минут на заказ
Точность данных	Высокая вероятность ошибок (10-15%)	Минимальные ошибки (1-

## Продолжение таблицы 1

Контроль за выполнением	Сложно отследить статус заказа	Статус отслеживается в реальном времени
Аналитика	Отчёты формируются вручную, данные устаревают	Автоматические отчёты в реальном времени
Масштабируемость	Не справляется с ростом заказов	Легко масштабируется
Затраты на персонал	Высокие (требуется много сотрудников)	Низкие (автоматизация сокращает штат)
Интеграция с другими системами	Отсутствует или сложна в реализации	Легкая интеграция с CRM, ERP, складскими системами

### **Автоматизация как решение проблем учёта**

Автоматизация позволяет не только упростить учёт заказов, но и улучшить взаимодействие с клиентами. Благодаря современным информационным системам можно быстро предоставлять информацию о статусе заказа, а клиенты в любое время получают актуальные данные о том, на каком этапе находится их заказ. Это способствует повышению доверия к компании. Кроме того, возможность интеграции систем с другими платформами, такими как CRM, делает работу более слаженной. Успех многих компаний, таких как SAP, подтверждает эффективность автоматизации учёта доходов и взаимодействия с клиентами.

Технические решения также играют важную роль в автоматизации. Существуют различные программные продукты, которые позволяют настроить учёт заказов под специфические нужды бизнеса. Выбор правильного программного обеспечения может существенно повлиять на эффективность работы. Системы, предлагающие многофункциональные решения, охватывают финансовые, производственные и административные

данные, что делает их особенно ценными для предприятий. Внедрение таких систем позволяет не только упростить учёт, но и повысить его точность. При этом важно учитывать, что универсальные решения подходят не всем, и здесь необходимо искать индивидуальный подход к каждому бизнесу.

Автоматизация учёта также способствует снижению человеческого фактора, минимизируя риск ошибок и неточностей в данных. Это особенно актуально для компаний, работающих с большими объёмами информации, где даже мелкие погрешности могут привести к серьёзным последствиям. Например, автоматизированные системы способны оперативно выявлять расхождения в финансовых отчётах и сигнализировать о необходимости корректировок.

Важно подчеркнуть, что внедрение автоматизации требует не только технической подготовки, но и обучения сотрудников. Без должной подготовки персонала даже самые передовые технологии могут остаться невостребованными. Поэтому успешная автоматизация всегда сопровождается разработкой стратегии обучения и адаптации команды к новым инструментам.

Ещё одним преимуществом автоматизации является возможность анализа больших данных, что позволяет выявлять тренды и прогнозировать спрос. Это даёт компаниям конкурентное преимущество, позволяя оперативно реагировать на изменения рынка и оптимизировать бизнес-процессы. Например, аналитика на основе автоматизированных систем помогает выстраивать более точные маркетинговые стратегии и улучшать качество обслуживания клиентов. [17]

## **Глава 2. Анализ существующих систем автоматизации учёта заказов**

### **2.1. Обзор популярных систем автоматизации**

Автоматизация учета заказов требует выбора системы, которая будет соответствовать специфике компаний. Существующие решения предлагают разные подходы к управлению рекламой, клиентами и проектами, каждая платформа обладает как преимуществами, так и недостатками.

Если компании требуется интеграция с бухгалтерией, складом и документооборотом, 1С: Управление рекламой становится очевидным выбором. Оно охватывает все направления рекламной деятельности - от наружной рекламы до телерадиовещания. Встроенная интеграция с другими продуктами 1С делает его удобным для крупных агентств, где важен комплексный учет.

Для компаний, которые сосредоточены на управлении клиентской базой и обработке заказов, ПринтОфис24 предлагает удобный инструмент. Простота интерфейса и адаптация под реальные процессы рекламных агентств делает его отличным выбором для малого и среднего бизнеса. Однако масштабируемость платформы ограничена - крупные агентства с большим объемом заказов могут столкнуться с нехваткой функциональности.

Если же агентству необходимо управление интернет-рекламой, AdRiver предоставляет инструменты для точного таргетинга и ретаргетинга. Поддержка различных рекламных форматов и аналитика охвата дают возможность детально настраивать кампании. Это особенно полезно для агентств, работающих с большими рекламными бюджетами и сложными стратегиями продвижения. Однако использование AdRiver требует определенных знаний - без специалистов в области интернет-рекламы система может оказаться сложной для освоения.

Для тех, кто ищет универсальное решение для автоматизации контекстной рекламы, eLama предлагает централизованный контроль кампаний в различных рекламных сетях. Простота управления ставками, автоматическая разметка UTM-метками и аналитика делают eLama хорошим выбором для агентств, работающих с многоканальной рекламой. Однако для небольших компаний стоимость сервиса может оказаться высокой, особенно если бюджет на рекламу ограничен. [17]

Еще одним инструментом для автоматизации контекстной рекламы является Aori, использующий метод SKAG (Single Keyword Ad Groups), который обеспечивает более точную настройку кампаний. Это решение подходит для агентств, которые хотят получить максимальную отдачу от рекламы в Google Ads, Яндекс.Директ и Facebook Ads. Однако высокая стоимость и сложность настройки делают его менее доступным для новичков.

## 2.2. Сравнительный анализ систем

Таблица 2 - Сравнительный анализ аналогов

Параметр	1С: Управление рекламой	ПринтОфис2 4	AdRiver	eLama	Aori
Основное назначение	Управление различными видами рекламы	Управление клиентской базой и заказами	Управление интернет- рекламой	Автоматизация контекстно й рекламы	Автоматизация контекстно й рекламы
Интеграция	Глубокая с продуктами 1С	Простая адаптация под потребности	Требуется квалифицир ованная настройка	Поддержка множества рекламных платформ	Интеграция с основными рекламным и платформа ми
Сложность внедрения	Высокая, требуется обучение	Низкая, интуитивно понятный интерфейс	Средняя, необходимы специалисты	Средняя, возможны доп. расходы	Высокая, требуется опытный персонал

## Продолжение таблицы 2

Масштабируемость	Подходит для крупных организаций	Ограничена для крупных компаний	Подходит для средних и крупных агентств	Подходит для агентств разного размера	Подходит для агентств разного размера
Стоимость	Зависит от конфигурации и	Доступная для малого и среднего бизнеса	Зависит от объема использования	Бесплатный тариф с комиссией за бюджет	От 990 руб/мес

### 2.3. Критерии выбора системы для автоматизации

В результате выбор платформы зависит от масштаба агентства, основных направлений рекламы и наличия квалифицированных специалистов. Если требуется полная интеграция с внутренними бизнес-процессами, 1С: Управление рекламой станет лучшим решением. Для простого управления клиентами и заказами больше подойдет ПринтОфис24. Если же главная цель – контроль интернет-рекламы, стоит обратить внимание на AdRiver, eLama или Aorі, исходя из уровня квалификации сотрудников и бюджета.

## **Глава 3. Анализ текущих бизнес-процессов учёта заказов**

### **3.1. Описание текущей системы учёта заказов**

#### **3.1.1. Общая характеристика текущей системы**

В рамках анализа деятельности ООО «Диджитал Агентство» рассмотрены ключевые процессы, определяющие его работу [10]. Основными входными данными являются информация о клиентах и их заказы, которые становятся основой для выполнения рекламных проектов. Управление процессами осуществляется с учетом нормативных требований, правил и регламентов, определяющих порядок выполнения задач.

Центральным элементом модели является функционирование рекламного агентства, в рамках которого задействованы различные сотрудники: администраторы, менеджеры по продажам, маркетологи, отдел разработки и руководящий персонал. Их взаимодействие обеспечивает выполнение заказов клиентов и достижение поставленных целей.

Результатом деятельности агентства являются выполненные рекламные проекты и отчетность по проделанной работе, которая позволяет анализировать эффективность работы и оптимизировать процессы. Такой подход обеспечивает системное управление проектами и способствует повышению качества оказываемых услуг.

Согласно диаграмме работы предприятия IDEF0 (рисунок 1) входными данными для работы рекламного агентства являются информация о клиентах, и его заявка:

- управляющие воздействия - правила и регламенты;
- механизмы – администратор, менеджер по продажам, отдел разработки, маркетологи, и руководящий персонал;
- выходные данные - выполняемые работы и отчеты по деятельности.

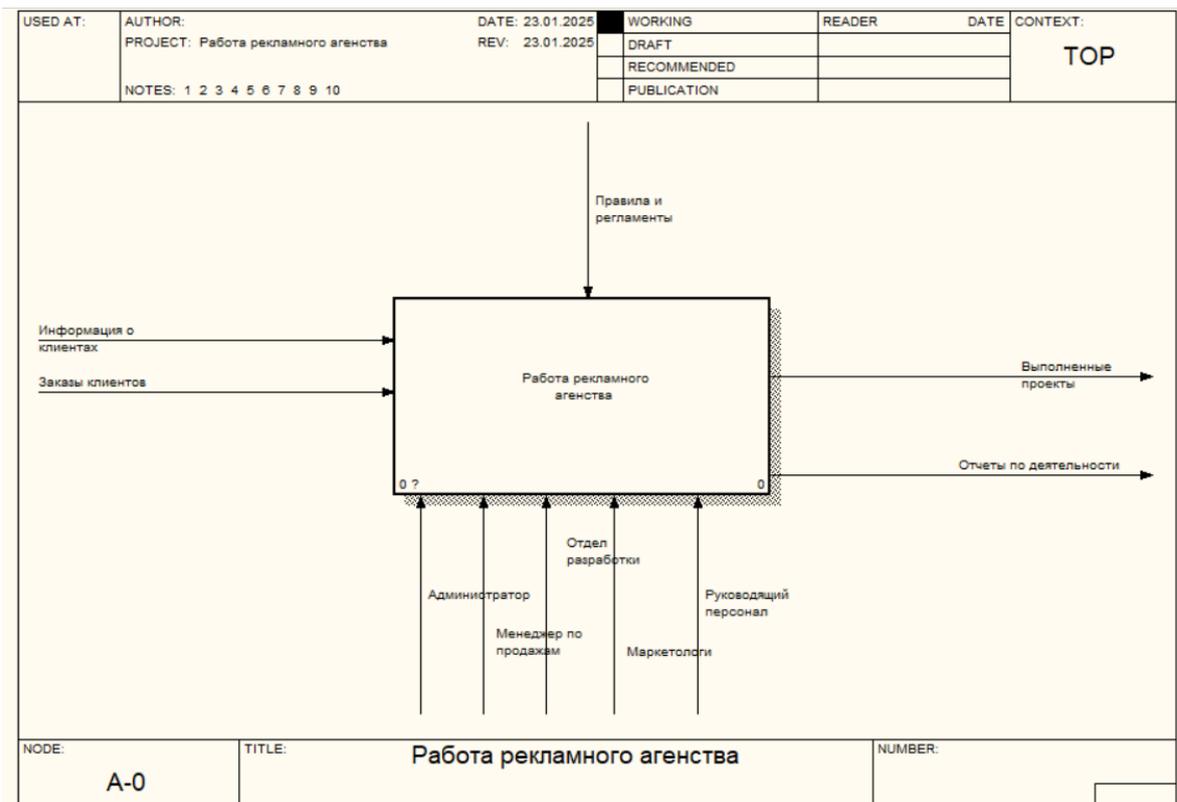


Рисунок 1 - Диаграмма работы предприятия IDEF0

В рамках анализа бизнес-процессов рекламного агентства рассмотрена структура работы, включающая ключевые этапы взаимодействия с клиентами и реализации проектов. Диаграмма отражает последовательность выполнения задач от оформления заказа до завершения проекта и его контроля.

Процесс начинается с получения информации о клиентах и их заказах. На первом этапе происходит оформление заказа, где фиксируются основные требования клиента. Данные передаются в клиентскую базу, что позволяет систематизировать информацию и учитывать правила и регламенты компании.

Следующий этап – разработка и ведение проекта, где на основе технического задания или брифа формируются основные рабочие процессы. В этом участвуют менеджеры по продажам, отдел разработки и маркетологи, что обеспечивает комплексный подход к выполнению заказов.

После разработки начинается отслеживание хода выполнения проекта, в ходе которого анализируется соответствие работ требованиям заказчика и

плановым срокам. Руководящий персонал получает данные о проектах, формируя отчеты о деятельности компании.

Итогом процесса является готовый рекламный проект, который передается клиенту. Автоматизация и четкое распределение функций на каждом этапе позволяют минимизировать риски, оптимизировать временные затраты и повысить эффективность работы агентства.

Ниже изображена декомпозиция контекстной диаграммы на рисунке 2.

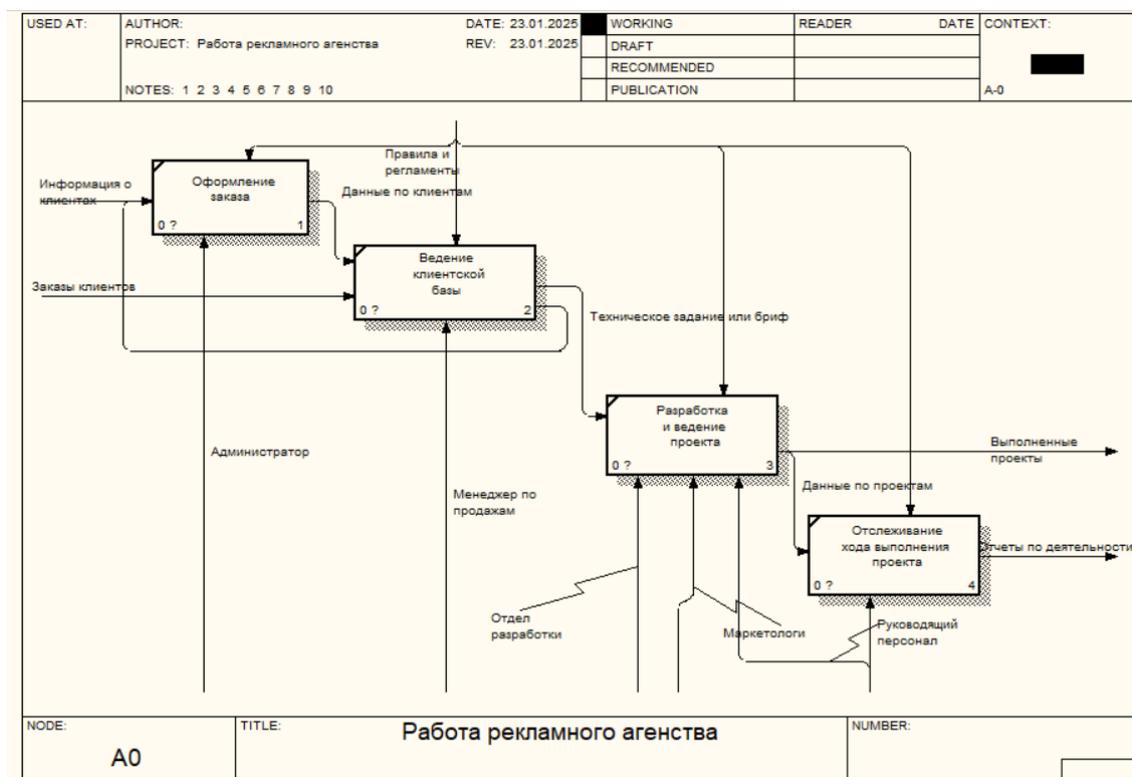


Рисунок 2 - Декомпозиция контекстной диаграммы

Согласно декомпозиции - процесс состоит из оформления заявки, ведения клиентской базы менеджерами, разработки и поддержки проекта, отслеживания хода выполнения проекта.

В процессе обработки клиентских заявок одним из ключевых этапов является согласование и реализация проекта. Этот процесс включает несколько последовательных шагов, обеспечивающих прозрачность работы,

контроль исполнения и эффективное взаимодействие между участниками. Основная цель данного этапа - гарантировать соответствие проекта требованиям заказчика и его успешную реализацию.

На рисунке 3 ниже показана диаграмма работы рекламного агентства AS IS в нотации BPMN.

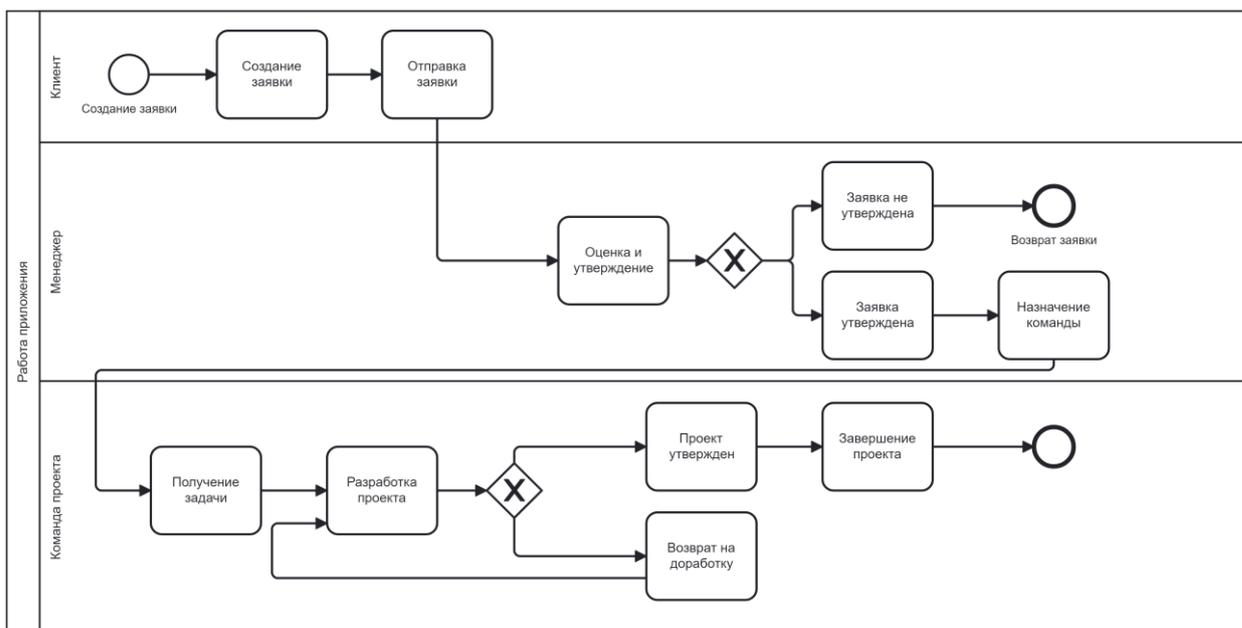


Рисунок 3 – Диаграмма работы «Диджитал агентства» AS IS

Процесс начинается с подачи заявки клиентом, где он формулирует свои потребности и отправляет запрос на рассмотрение. После этого заявка поступает к менеджеру, который проводит ее оценку и анализ. На данном этапе важно определить соответствие проекта требованиям, оценить возможные риски и рассчитать ресурсы, необходимые для его выполнения.

После анализа заявки принимается решение о ее дальнейшем статусе. Если заявка не утверждена, она возвращается клиенту для внесения корректировок. В случае положительного решения менеджер назначает команду для работы над проектом, что обеспечивает оперативное распределение задач и подготовку к разработке.

Команда проекта приступает к выполнению задачи, начиная с получения задания и планирования этапов работы. Далее следует этап разработки, в ходе которого создается проектная документация и реализуются технические решения. После завершения работ проект проходит проверку и утверждение.

Если проект соответствует требованиям и ожиданиям, он считается завершенным. В случае выявления недостатков он возвращается на доработку, после чего проходит повторную проверку. Такой подход позволяет обеспечить высокое качество выполнения задачи и минимизировать возможные риски при внедрении проекта. [17]

### **3.2. Выявление «узких мест»**

В рамках анализа объекта автоматизации, рассмотрена деятельность ООО «Диджитал Агентство», где выявлены ключевые проблемы, замедляющие производственные процессы и взаимодействие с клиентами. Основными процессами, которые нуждаются в улучшении, являются управление проектами, планирование задач и коммуникация между командами. В текущем состоянии эти процессы имеют ряд недостатков, таких как избыточные временные затраты, недостаточная оперативность в обработке данных и сложность в отслеживании этапов выполнения проектов.

Существующая организация работы предполагает использование разнообразных цифровых инструментов, но их разрозненность создает барьеры для эффективного взаимодействия между командами и клиентами. Отсутствие единой централизованной платформы усложняет управление ресурсами и контроль над проектами, что увеличивает риск задержек в выполнении задач и возникновения ошибок.

### 3.3. SWOT-анализ текущей системы

#### 3.3.1. Методика проведения SWOT-анализа

SWOT-анализ текущей системы учёта заказов проводится для выявления её сильных и слабых сторон, а также возможностей и угроз, связанных с внешними факторами. Анализ включает четыре ключевых аспекта:

trengths (Сильные стороны) – внутренние преимущества системы.

eaknesses (Слабые стороны) – внутренние недостатки системы.

pportunities (Возможности) – внешние факторы, которые могут быть использованы для улучшения системы.

hreats (Угрозы) – внешние факторы, которые могут негативно повлиять на систему. [4]

#### . SWOT-матрица текущей системы учёта заказов

Таблица 3 - SWOT-матрица текущей системы учёта заказов

SWOT-матрица текущей системы учёта заказов	
Внутренние факторы	Внешние факторы
Сильные стороны (S)	Возможности (O)
Простота использования	Внедрение CRM-систем
Низкие затраты на поддержку	Интеграция механизма авторизации пользователей
Гибкость	Централизованная база данных
- Малая зависимость от техники	Автоматизация отчётности
Слабые стороны (W)	Угрозы (T)
Высокая вероятность ошибок	Усиление конкуренции
Задержки в обработке	Рост количества заказов
Потеря данных	Изменение законодательства
Отсутствие аналитики	Технические сбои

## **Выводы по SWOT – анализу текущей системы учета заказов**

Система, ориентирована на автоматизацию процессов обработки клиентских заявок, планирования задач, распределения обязанностей между участниками проектных команд и контроля выполнения заказов. Особое внимание уделяется вопросам хранения и обработки информации в централизованной базе данных, обеспечивающей целостность, доступность и актуальность данных на всех этапах реализации проекта. Также предполагается интеграция механизма авторизации пользователей, который позволит организовать разграничение прав доступа и обеспечить информационную безопасность. Важным элементом является модуль аналитики, позволяющий формировать отчёты и визуализировать показатели деятельности в режиме реального времени.

## Глава 4. Проектирование системы автоматизации учёта заказов

### 4.1. Архитектура комплекса программного обеспечения

Многоуровневая архитектура, выбранная для проекта, представляет собой оптимальное решение для обеспечения заявленных требований: гибкости, масштабируемости и надёжности. Она подразумевает разделение системы на отдельные, взаимодействующие между собой уровни, каждый из которых выполняет конкретную функцию. Это позволяет добиться нескольких ключевых преимуществ. [16]

**Гибкость:** Разделение на уровни упрощает внесение изменений в систему. Например, модификация клиентской части (Frontend) не требует изменений в серверной части (Backend) при условии сохранения интерфейса взаимодействия. Это значительно ускоряет процесс разработки и внедрения новых функций.

**Масштабируемость:** Многоуровневая архитектура позволяет масштабировать отдельные компоненты системы независимо друг от друга. При увеличении нагрузки на систему можно увеличить ресурсы, выделенные для конкретного уровня (например, увеличить мощность сервера Backend или расширить хранилище данных в Database) без влияния на другие компоненты.

**Надёжность:** Использование многоуровневой архитектуры повышает надёжность системы. В случае сбоя в одном из уровней, система может продолжать функционировать, обрабатывая запросы на других уровнях. Кроме того, разделение ответственности между уровнями упрощает отладку и устранение ошибок.

Детализация архитектуры:

- Уровень представления (Frontend): отвечает за взаимодействие с пользователем. Реализуется пользовательский интерфейс, обеспечивается визуализация данных, ввод и обработка данных от пользователей.

- Уровень бизнес-логики (Backend): содержит бизнес-правила, алгоритмы обработки данных, логику работы системы. Обеспечивает взаимодействие с уровнем доступа к данным.
- Уровень доступа к данным (Database): отвечает за хранение и обработку данных. Включает в себя базу данных, средства управления данными и логику взаимодействия с уровнем бизнес-логики.
- Уровень интеграции (API): обеспечивает взаимодействие между различными уровнями. Предоставляет интерфейсы для обмена данными, обработки запросов и выполнения операций.

Выбор данной архитектуры обусловлен необходимостью удовлетворения требований к производительности, безопасности и удобству использования системы.

На рисунке 4 представлена диаграмма вариантов использования (Use Case Diagram), иллюстрирующая основные сценарии взаимодействия пользователей с системой автоматизации учета заказов.

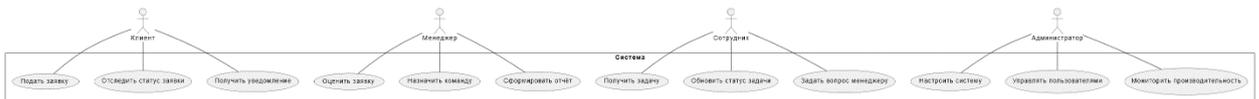


Рисунок 4 - Диаграмма использования

## . Клиентская часть (Frontend)

Описание:

- Интерфейс для клиентов: форма подачи заявки, личный кабинет для отслеживания статуса заявок.
- Интерфейс для менеджеров: панель управления заявками, назначение команд, отслеживание этапов проекта.
- Интерфейс для команды проекта: доступ к задачам, возможность обновления статуса задач.

Обоснование:

- React.js или HTML - современные фреймворки для создания динамичных и удобных пользовательских интерфейсов.
- Адаптивный дизайн - обеспечит корректное отображение на всех устройствах (десктоп, планшет, мобильный).
- Библиотеки для визуализации - для отображения статистики и аналитики по проектам. [6]

### **.1.3. Серверная часть (Backend)**

Описание:

обработка запросов от клиентской части, управление бизнес-логикой.

Основные модули:

- Модуль обработки заявок.
- Модуль управления проектами и задачами.
- Модуль проверки и утверждения проектов. [15]

Обоснование:

- Django или Flask - популярные Python-фреймворки для разработки серверной части. [7]
- Django подходит для крупных проектов с большим количеством функционала и встроенной админ-панелью.
- Flask - более лёгкий и гибкий, подходит для проектов с индивидуальными требованиями.
- RESTful API - стандарт для взаимодействия между клиентской и серверной частями.

### **1.4. База данных (Database)**

Описание:

хранение данных о клиентах, заявках, проектах, командах и задачах.

Основные таблицы:

- Clients- информация о клиентах.
- Applications - заявки клиентов.

- Projects - проекты и их статус.
- Teams - команды и их состав.
- Tasks - задачи и их статус.

Обоснование:

- PostgreSQL - надёжная и масштабируемая СУБД, поддерживает сложные запросы и транзакции.
- ORM (Object-Relational Mapping) - использование Django ORM или
- Резервное копирование - автоматическое создание бэкапов для предотвращения потери данных. [15]

### **.1.5. Взаимодействие между компонентами**

Описание:

- Frontend отправляет запросы к Backend через API. обрабатывает запросы, взаимодействует с базой данных и возвращает результаты.

Обоснование:

- API обеспечивает гибкое взаимодействие между клиентской и серверной частями.
- Celery + Redis - надёжное решение для асинхронной обработки задач, что повышает производительность системы. [13]

### **.1.6. Безопасность**

Описание:

- Аутентификация и авторизация - использование JWT (JSON Web Tokens) для защиты API.
- Шифрование данных - защита конфиденциальной информации (например, данных клиентов).
- Защита от атак - использование HTTPS, CSRF-защиты, валидация входных данных. [18]

Обоснование:

стандарт для безопасной аутентификации и авторизации.

защита данных при передаче между клиентом и сервером.

- Валидация данных - предотвращение SQL-инъекций и других уязвимостей.

### **.1.7. Масштабируемость и производительность**

Описание:

- Горизонтальное масштабирование - возможность добавления новых серверов для обработки увеличенной нагрузки. [13]
- Кэширование - использование Redis для кэширования часто запрашиваемых данных.
- Мониторинг- инструменты для отслеживания производительности и ошибок (например, Prometheus + Grafana).

Обоснование:

- Горизонтальное масштабирование - позволяет системе оставаться отзывчивой при росте нагрузки.
- Кэширование - ускоряет обработку запросов и снижает нагрузку на базу данных.
- Мониторинг - помогает оперативно выявлять и устранять проблемы.

## **4.2 Модульная структура**

Комплекс программ будет состоять из следующих модулей:

Таблица 4 - Модульная структура комплекса программ

Модуль	Описание
Модуль работы с клиентами	Обработка данных о клиентах, регистрация, редактирование профиля.
Модуль заявок	Создание, редактирование, оценка и утверждение заявок.

Продолжение таблицы 4

Модуль проектов	Управление проектами, распределение задач, контроль статусов.
Модуль команд	Формирование команд, назначение руководителей, распределение сотрудников.
Модуль задач	Создание, назначение и отслеживание статуса задач.
Модуль уведомлений	Отправка уведомлений клиентам, менеджерам и сотрудникам (email, push-уведомления).
Модуль интеграции	Взаимодействие с внешними системами (CRM, системы управления проектами).
Модуль отчётности	Генерация отчётов по проектам, заявкам, задачам.

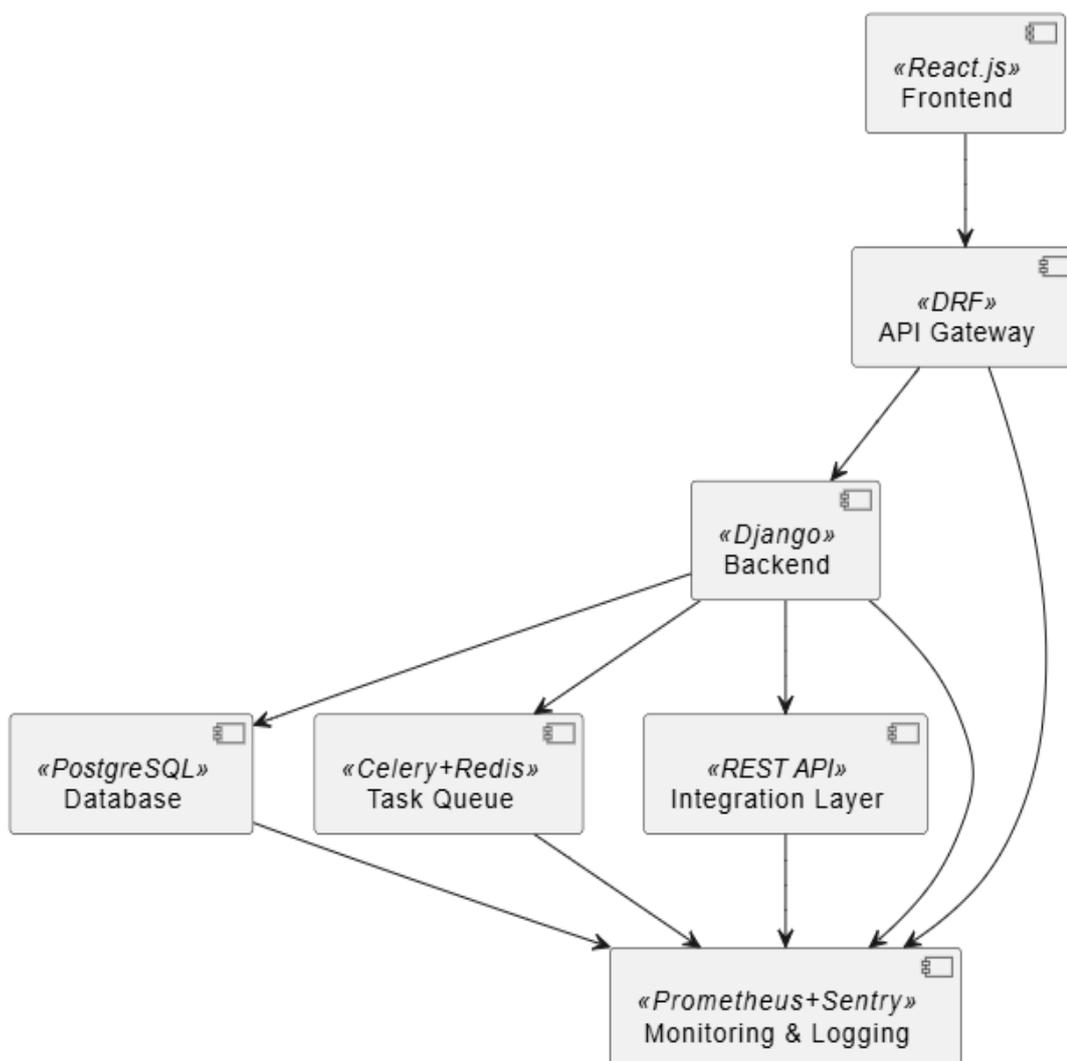


Рисунок 5 - Архитектура комплекса программного обеспечения

Каждый модуль системы автоматизации направлен на оптимизацию ключевых бизнес-процессов, что позволяет повысить эффективность работы компании. Благодаря интеграции всех модулей между собой обеспечивается согласованность данных и минимизируются риски ошибок. Гибкость системы позволяет адаптировать её под специфические нужды бизнеса, а также масштабировать по мере роста компании. Кроме того, автоматизация рутинных операций освобождает время сотрудников для решения стратегических задач, что способствует развитию и конкурентоспособности организации. [12]

## **4.3. Алгоритмы обработки данных**

### **4.3.1. Алгоритм обработки заявок**

Описание:

клиент подаёт заявку через веб-интерфейс.

система сохраняет заявку в базе данных.

менеджер оценивает заявку и принимает решение (утвердить или вернуть на доработку).

система уведомляет клиента о статусе заявки.

Алгоритм обработки заявок представлен в Приложении А.

Таблицы связаны между собой внешними ключами (FOREIGN KEY), что обеспечивает целостность данных и позволяет отслеживать связи между сущностями (например, какой клиент подал заявку, какой проект создан на её основе, какие сотрудники и задачи к нему прикреплены).

#### **Алгоритм управления учетом заказов**

На рисунке 6 представлена диаграмма последовательности, иллюстрирующая процесс обработки заявки в системе автоматизации учёта заказов.

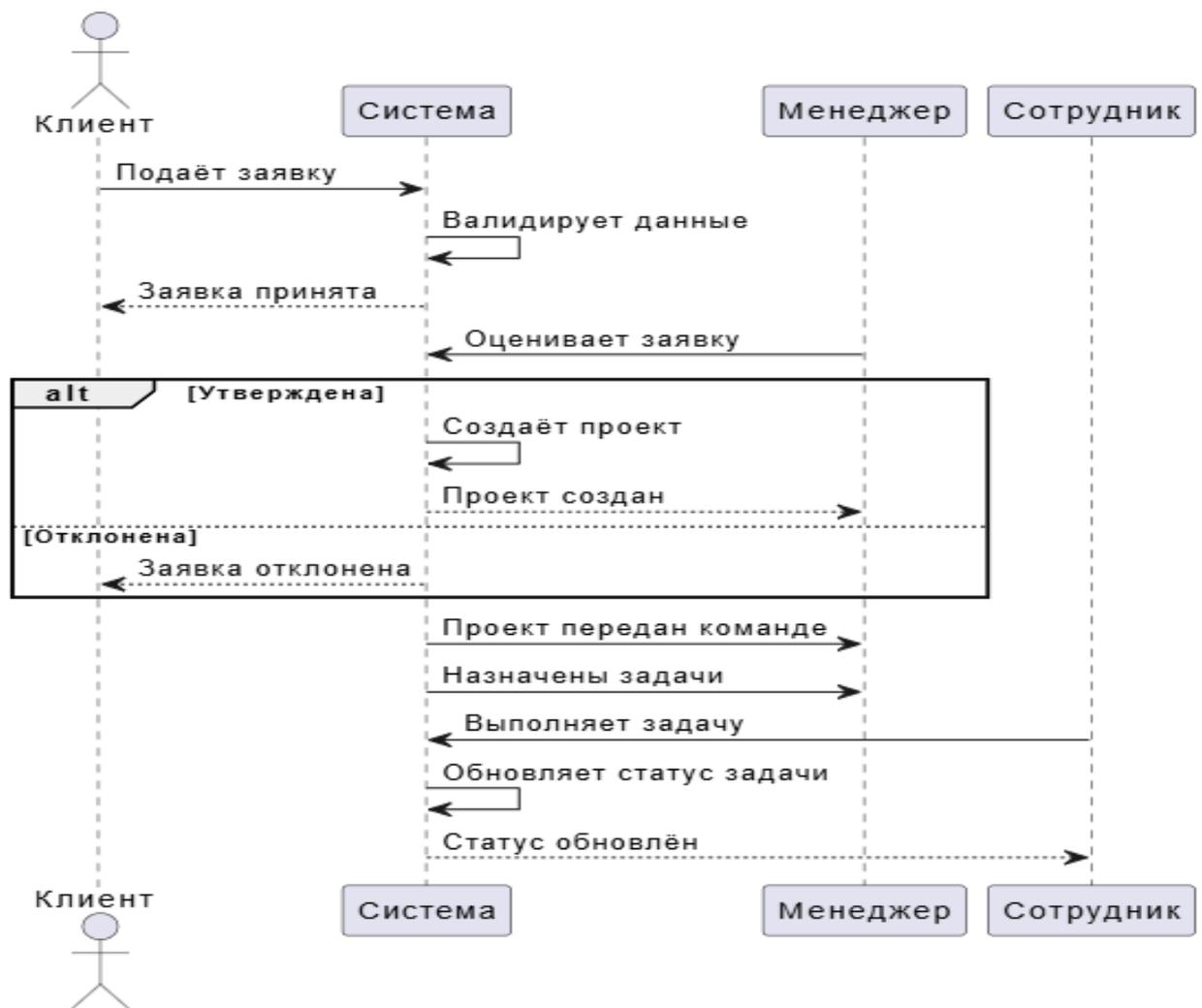


Рисунок 6 - Диаграмма последовательности

Описание:

осле утверждения заявки создаётся проект.

менеджер назначает команду и распределяет задачи.

система отслеживает статус проекта и задач.

Код на Django представлен в Приложении Б, он описывает модели и представление (view) для управления проектами, командами и задачами в веб-приложении.

### Алгоритм уведомлений

Описание:

система отправляет уведомления при изменении статуса заявки, проекта или задачи.

уведомления могут быть отправлены по email или через push-уведомления.

Пример кода на Python, Django + Celery представлен в Приложении В.

Код демонстрирует использование асинхронных задач с помощью Celery в Django-приложении для отправки уведомлений клиентам об изменении статуса их заявок.

### **Пример интеграции с CRM**

Описание:

синхронизация данных о клиентах и заявках с внешней CRM-системой.

использование API для обмена данными.

Пример кода на Python, requests приведен в Приложении Г.

Код описывает функцию для синхронизации данных о клиенте с внешней CRM-системой через API.

### **5. Пример генерации отчёта по проектам**

Описание:

формирование отчёта о статусах проектов.

экспорт отчёта в формате CSV или PDF.

Пример кода на Python, Django + pandas представлен в Приложении Д.

Код реализует функционал экспорта данных о проектах в CSV-формат в веб-приложении на Django. Он работает следующим образом:

Сначала из базы данных извлекаются все записи о проектах (используя модель Project). Затем данные преобразуются в удобную табличную структуру с помощью библиотеки pandas, где каждый проект представлен своими ключевыми атрибутами: идентификатором, названием и статусом.

После этого создаётся HTTP-ответ, который настраивается для отправки файла в формате CSV. Пользователю предлагается скачать файл под именем projects\_report.csv. Функция записывает данные из DataFrame непосредственно

в объект ответа, что позволяет сгенерировать и отправить файл "на лету", без промежуточного сохранения на сервере.

Такой подход удобен для автоматизации отчётности и позволяет пользователям легко экспортировать данные для дальнейшего анализа в других инструментах, например, в Excel или системах бизнес-аналитики. Это особенно полезно для менеджеров, аналитиков или других специалистов, которым необходим быстрый доступ к актуальной информации о проектах.

#### **4.4. Проектирование структуры базы данных**

##### **Информационная модель системы автоматизации учёта заказов**

Информационная модель в данном проекте описывает структуру данных, связи между сущностями, а также процессы взаимодействия с внешними информационными системами и пользователями.

Основные сущности и их атрибуты

Сущность «Клиент» (Client)

- Описание: хранит информацию о клиентах, которые подают заявки.
- Атрибуты:
  - client\_id (уникальный идентификатор)
  - name (имя клиента)
  - contact\_info (контактная информация: email, телефон)
  - registration\_date (дата регистрации)

. Сущность «Заявка» (Application)

- Описание: содержит информацию о заявках, поданных клиентами.
- Атрибуты:
  - application\_id (уникальный идентификатор)
  - client\_id (ссылка на клиента)
  - description (описание заявки)

- submission\_date (дата подачи)
- status (статус: «на рассмотрении», «утверждена», «возвращена на доработку»)

#### . Сущность «Проект» (Project)

- Описание: хранит данные о проектах, созданных на основе утверждённых заявок.
- Атрибуты:
  - project\_id (уникальный идентификатор)
  - application\_id (ссылка на заявку)  
(название проекта)
  - start\_date (дата начала)
  - end\_date (дата завершения)
  - status (статус: «в разработке», «на проверке», «завершён»)

#### 4. Сущность «Команда» (Team)

- Описание: Информация о командах, назначенных на проекты.
- Атрибуты:
  - (уникальный идентификатор)
  - project\_id (ссылка на проект)
  - team\_lead (руководитель команды)
  - members (список участников команды)

#### 5. Сущность «Сотрудник» (Employee)

- Описание: Данные о сотрудниках компании.
- Атрибуты:
  - employee\_id (уникальный идентификатор)
  - name (имя сотрудника)
  - position (должность)
  - contact\_info (контактная информация)

#### 6. Сущность «Задача» (Task)

- Описание: Задачи, которые выполняются в рамках проектов.

- Атрибуты:
- task\_id (уникальный идентификатор)
- project\_id (ссылка на проект)
- description (описание задачи)
- assigned\_to (исполнитель)
- status (статус: «в работе», «выполнена»)

### Связи между сущностями

На рисунке 7 представлена диаграмма классов, описывающая основные сущности системы автоматизации учета заказов и их связи.

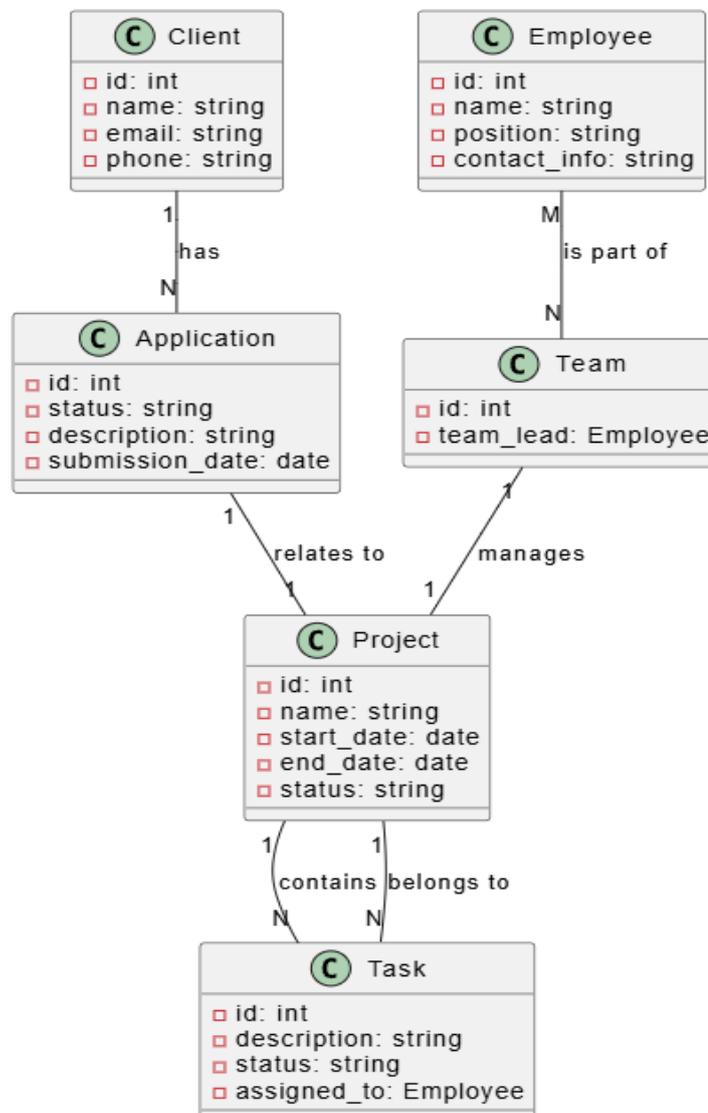


Рисунок 7 - Диаграмма классов

- Клиент → Заявка: Один клиент может подать несколько заявок (связь один к многим).
- Заявка → Проект: Одна заявка может породить один проект (связь один к одному).
- Проект → Команда: Один проект выполняется одной командой (связь один к одному).
- Команда → Сотрудник: Одна команда состоит из нескольких сотрудников (связь «многие ко многим»).
- Проект → Задача: Один проект включает несколько задач (связь один к многим).

## **Физические основы работы информационной системы**

### **Взаимодействие с пользователями**

- Клиенты взаимодействуют с системой через веб-интерфейс (подача заявок, отслеживание статуса).
- Менеджеры используют панель управления для оценки заявок, назначения команд и контроля выполнения проектов.
- Команды проектов получают задачи через личный кабинет, обновляют статус выполнения.

### **Взаимодействие с внешними системами**

- Интеграция с CRM-системами (например, Bitrix24, HubSpot) для синхронизации данных о клиентах и заявках.
- Интеграция с системами управления проектами (например, Jira, Trello) для синхронизации задач и статусов.
- Электронная почта и мессенджеры для отправки уведомлений клиентам и сотрудникам.

#### **4.4.5. Технологии взаимодействия**

##### **1. API (Application Programming Interface)**

API для взаимодействия между клиентской и серверной частями.

- Webhooks для оповещения внешних систем о изменениях (например, обновление статуса заявки).

## 2. Форматы обмена данными

основной формат для обмена данными между компонентами системы.

для интеграции с некоторыми внешними системами (например, 1С).

## 3. Протоколы передачи данных

для безопасной передачи данных между клиентом и сервером.

для реализации реального времени (например, уведомления о изменениях статуса).

### **Пример информационной модели в виде таблиц**

Таблица 5 – Сущность «клиент» (client)

Атрибут	Тип данных	Описание
		Уникальный идентификатор
		Имя клиента
		Контактная информация
		Дата регистрации
		Уникальный идентификатор

Эта таблица описывает структуру сущности «Клиенты» в базе данных, где каждый атрибут определяет ключевые данные о клиенте. Поле `client_id` служит первичным ключом (PK), обеспечивая уникальность каждой записи, а остальные поля - `name`, `contact_info` и `registration_date` - хранят имя клиента, его контактные данные и дату регистрации соответственно.

Таблица 6 - Сущность «Заявка» (Application)

Атрибут	Тип данных	Описание
		Уникальный идентификатор

		Ссылка на клиента
		Описание заявки
		Дата подачи
		Статус заявки

Эта таблица описывает структуру сущности «Заявки» в базе данных, где каждый атрибут определяет ключевые параметры заявки. Поле `application_id` является первичным ключом (PK), гарантирующим уникальность каждой записи. Поле `client_id` служит внешним ключом (FK), связывающим заявку с конкретным клиентом из таблицы клиентов.

Таблица 6 - Сущность «Проект» (Project)

		Уникальный идентификатор
		Ссылка на заявку
		Название проекта
		Дата начала
		Дата завершения
		Статус проекта

Эта таблица описывает структуру сущности «Проекты» в базе данных. Поле `project_id` выступает в роли первичного ключа (PK), обеспечивая уникальность каждой записи. Атрибут `application_id` является внешним ключом (FK), который связывает проект с конкретной заявкой, на основе которой он был создан.

## . Концептуальная схема данных

Генерация схем данных и подсхем приложений

ущности и их атрибуты

Таблица 7 - Концептуальная схема: Сущности и атрибуты

Сущность	client_id (PK), name, contact_info (email, телефон), registration_date
Клиент	application_id (PK), client_id (FK), description, submission_date, status
Заявка	project_id (PK), application_id (FK), name, start_date, end_date, status
Проект	team_id (PK), project_id (FK), team_lead (FK к Сотруднику), members (список FK к Сотрудникам)
Команда	employee_id (PK), name, position, contact_info
Сотрудник	employee_id (PK), name, position, contact_info
Задача	task_id (PK), project_id (FK), description, assigned_to (FK к Сотруднику), status

Эта схема описывает структуру сущностей базы данных для системы управления проектами и взаимодействия с клиентами, где каждая сущность связана с другими через ключевые отношения.

визи между сущностями

Таблица 8 - Концептуальная схема: Связи

Связь	Тип связи	Описание
Клиент → Заявка	Один ко многим	Один клиент может подать несколько заявок.
Заявка → Проект	Один к одному	Одна утверждённая заявка порождает один проект.
Проект → Команда	Один к одному	Один проект выполняется одной командой.

Продолжение таблицы 7

Команда → Сотрудник	Многие ко многим	Один сотрудник может входить в несколько команд,
---------------------	------------------	--

		одна команда состоит из нескольких сотрудников.
Проект → Задача	Один ко многим	Один проект включает несколько задач.

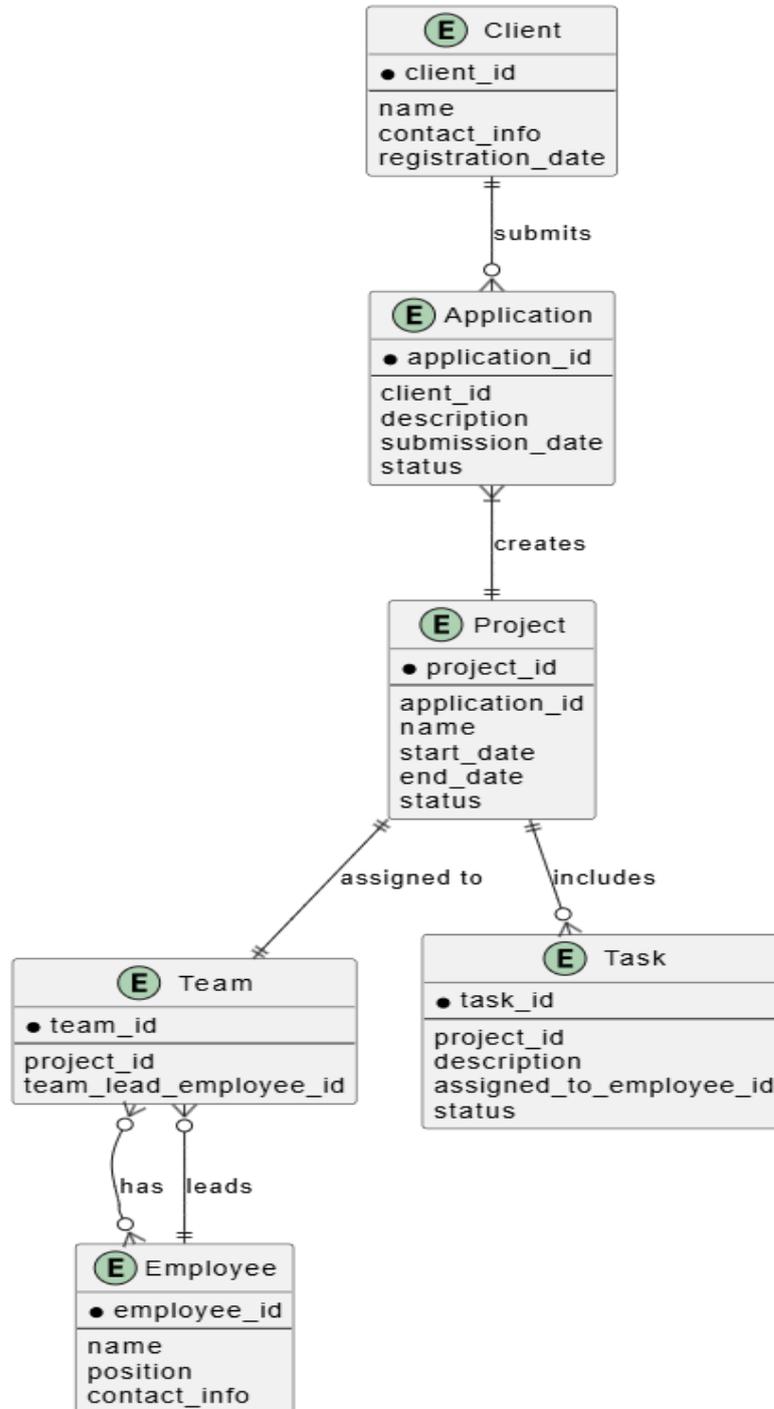


Рисунок 8 - ER диаграмма

Эта диаграмма представляет ER-модель (сущность-связь) системы управления проектами. Она иллюстрирует сущности (Клиент, Заявка, Проект, Команда, Сотрудник, Задача) и связи между ними: клиент подаёт заявки, на основе которых создаются проекты; проект назначается одной команде, состоящей из сотрудников, а также включает несколько задач, распределённых между сотрудниками.

## . Подсхемы приложений

Разберем подсхемы приложений, описывающие физическую структуру данных для конкретных приложений или модулей системы. Опираясь на концептуальную схему, но адаптированных под специфику работы каждого приложения.

### 1. Подсхема «Учёт клиентов и заявок»

Таблица 9 - Учет клиентов и заявок

Таблица	Поля
	client_id (INT, PK), name (VARCHAR), contact_info (VARCHAR), registration_date (DATE)
	application_id (INT, PK), client_id (INT, FK), description (TEXT), submission_date (DATE), status (VARCHAR)

Эти таблицы описывают структуру базы данных для хранения информации о клиентах и их заявках. [2]

### 2. Подсхема "Управление проектами"

Таблица 10 - Управление проектами

Подсхема: Управление проектами	
Таблица	Поля
	project_id (INT, PK), application_id (INT, FK), name (VARCHAR), start_date (DATE), end_date (DATE), status (VARCHAR)

Продолжение таблицы 10

	team_id (INT, PK), project_id (INT, FK), team_lead (INT, FK к Employees)
	team_id (INT, FK), employee_id (INT, FK)
	task_id (INT, PK), project_id (INT, FK), description (TEXT), assigned_to (INT, FK к Employees), status (VARCHAR)

Эта подсхема описывает структуру базы данных для управления проектами.

### 3. Подсхема "Учёт сотрудников"

Таблица 11 - Учет сотрудников

Подсхема: Учёт сотрудников	
Таблица	Поля
	employee_id (INT, PK), name (VARCHAR), position (VARCHAR), contact_info (VARCHAR)

Эта подсхема описывает таблицу для учёта сотрудников в базе данных. На основе разработанных схем создадим физическую структуру базы данных.

## 9. Проектирование структуры базы данных

Проектирование базы данных будет состоять из нескольких этапов. Проектирование структуры, создание таблиц, заполнение тестовыми данными и проверка работоспособности.

Сущности и их атрибуты

На основе концептуальной схемы, мы определили следующие сущности и их атрибуты:

Таблица 12 - Сущности и атрибуты базы данных

Сущность	Атрибуты
	client_id (INT, PK), name (VARCHAR), contact_info (VARCHAR), registration_date (DATE)
	application_id (INT, PK), client_id (INT, FK), description (TEXT), submission_date (DATE), status (VARCHAR)
	project_id (INT, PK), application_id (INT, FK), name (VARCHAR), start_date (DATE), end_date (DATE), status (VARCHAR)
	team_id (INT, PK), project_id (INT, FK), team_lead (INT, FK к Employee)
	employee_id (INT, PK), name (VARCHAR), position (VARCHAR), contact_info (VARCHAR)
	task_id (INT, PK), project_id (INT, FK), description (TEXT), assigned_to (INT, FK к Employee), status (VARCHAR)

Эта схема описывает основные сущности и их атрибуты в системе управления проектами, клиентами и сотрудниками.

## 2. Связи между таблицами

Таблица 13 - Связи между таблицами

Связь	Тип связи	Описание
Client → Application	Один ко многим	Один клиент может подать несколько заявок.
Application → Project	Один к одному	Одна утверждённая заявка порождает один проект.

Продолжение таблицы 13

Project → Team	Один к одному	Один проект выполняется одной командой.
Team → Employee	Многие ко многим	Один сотрудник может входить в несколько команд, одна команда состоит из нескольких сотрудников.
Project → Task	Один ко многим	Один проект включает несколько задач.

Представленная схема описывает структуру взаимоотношений между основными сущностями в системе управления проектами. [16]

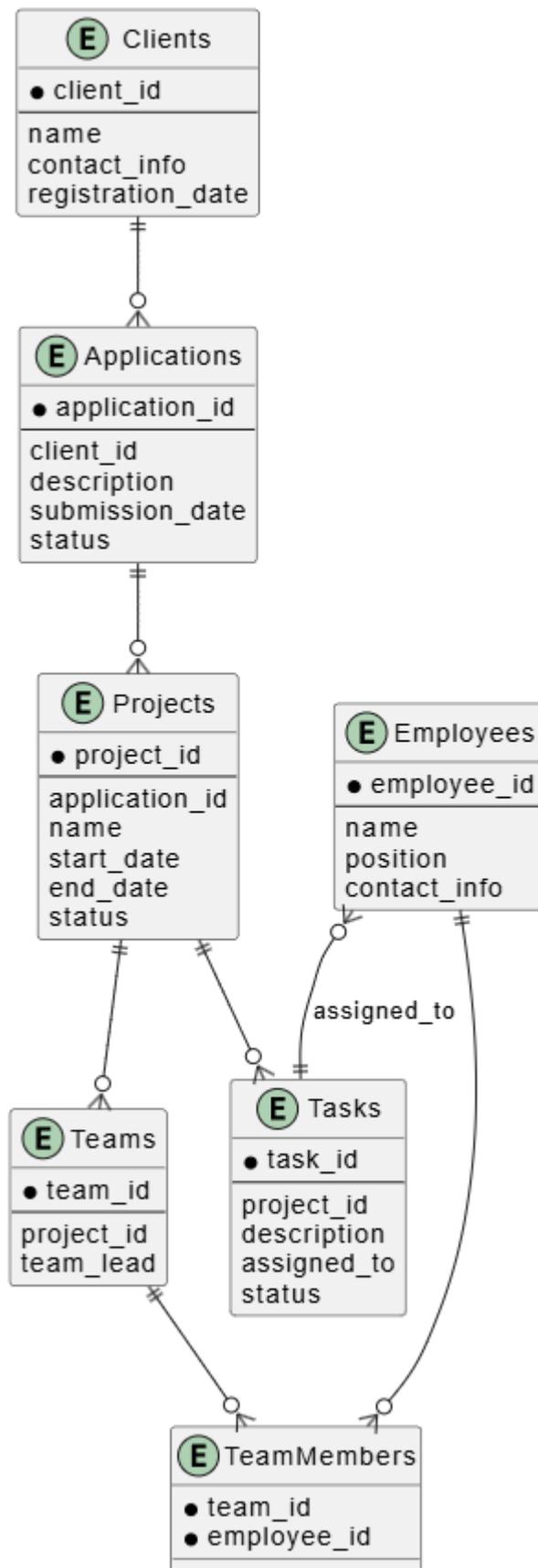


Рисунок 9 - ER- диаграмма базы данных

Такая структура позволяет эффективно управлять проектами, отслеживать статусы заявок, задач и команд, а также анализировать взаимодействие между участниками.

#### 4.5. Экранный формы, пользовательские меню

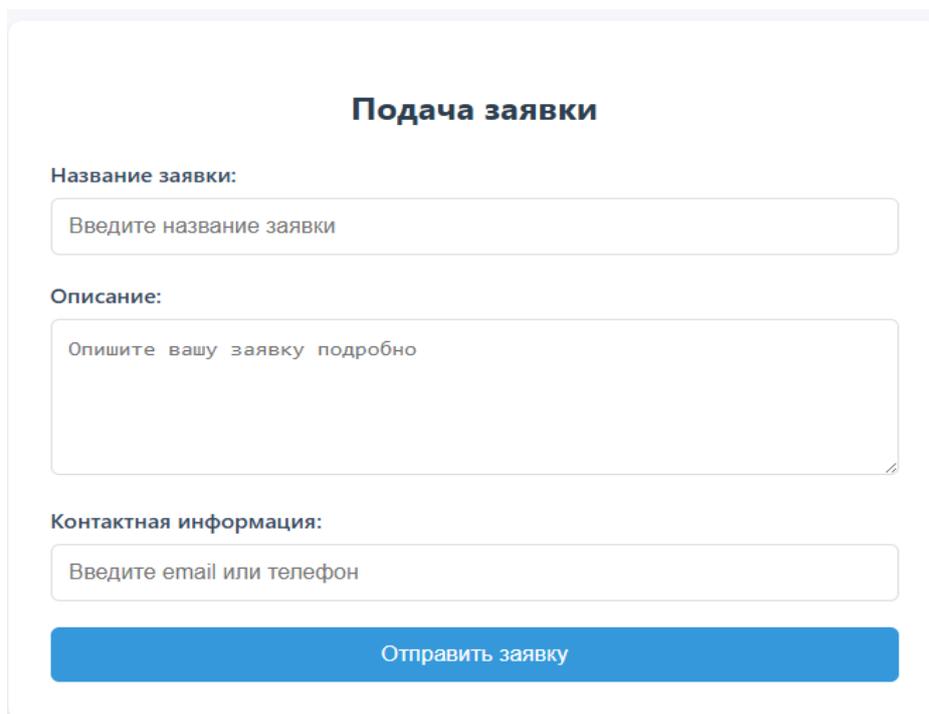
##### Форма подачи заявки (для клиентов)

Содержание:

- Поля: название заявки, описание, контактная информация.
- Кнопка «Отправить заявку».

На рисунке 10, представлена готовая форма веб интерфейса, подачи заявки клиентом, для ввода информации в поля экранной формы.

После отправки данные обрабатываются сервером по указанному URL.



**Поддача заявки**

Название заявки:  
Введите название заявки

Описание:  
Опишите вашу заявку подробно

Контактная информация:  
Введите email или телефон

Отправить заявку

Рисунок 10 - Вид веб интерфейса «подача заявки»

Форма минималистична - содержит только ключевые поля, что ускоряет заполнение.

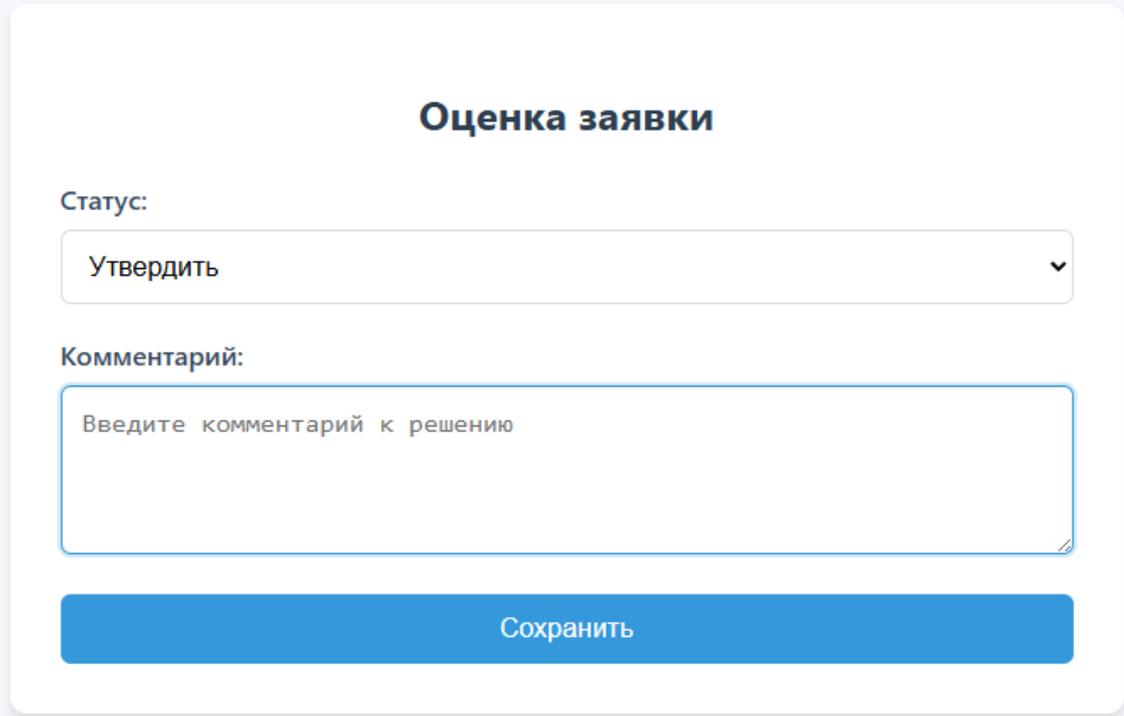
### **Форма создания проекта (для менеджеров)**

Содержание:

- Поля: статус заявки (утвердить/вернуть на доработку), комментарий.
- Кнопки: «Утвердить», «Вернуть на доработку».

На рисунке 11, представлена готовая форма веб интерфейса, создания проектов для менеджеров, в поля экранной формы.

Это форма для администраторов/модераторов, позволяющая быстро оценить заявку, присвоить ей статус и оставить комментарий.



**Оценка заявки**

Статус:

Утвердить

Комментарий:

Введите комментарий к решению

Сохранить

Рисунок 11 - Вид веб интерфейса «создания проекта»

Дизайн минималистичный, адаптивный, с интуитивно понятными элементами. Форма защищена от CSRF-атак и требует обязательного выбора статуса.

### 3. Форма утверждения проекта (для менеджеров)

Содержание:

- Поля: название проекта, дата начала, дата завершения, назначение команды.
- Кнопка «Создать проект».

На рисунке 12, представлена готовая форма веб интерфейса, утверждения проекта для менеджеров, в поля экранной формы.

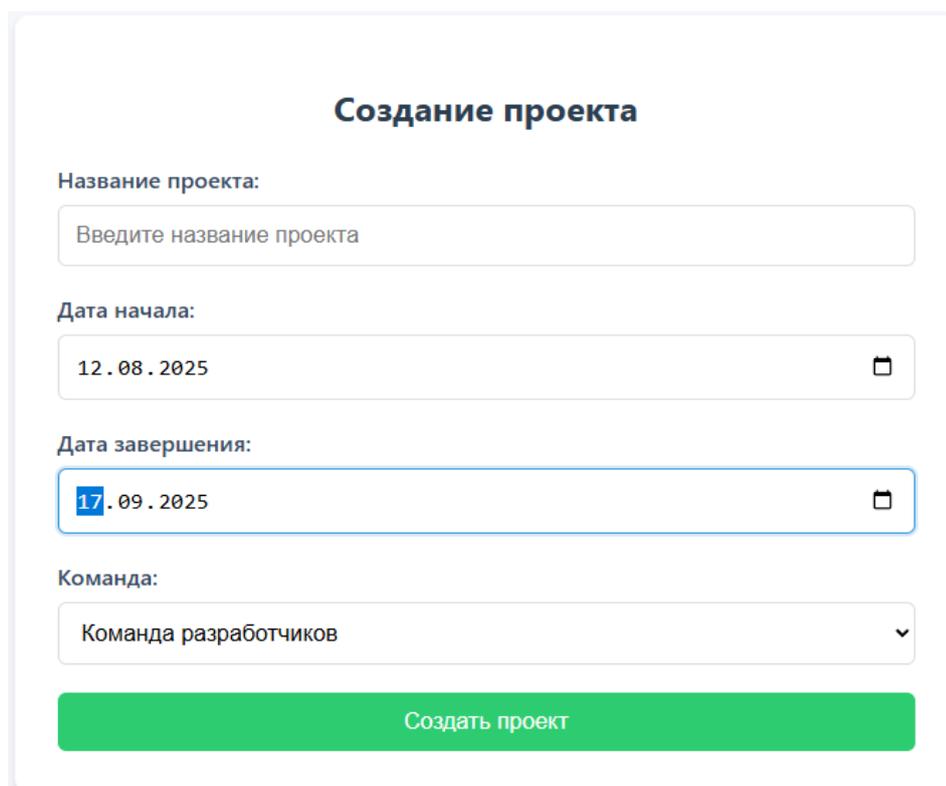


Рисунок 12 - Вид веб интерфейса «утверждения проекта»

Дизайн минималистичный, адаптивный, с интуитивно понятными элементами. Форма защищена от CSRF-атак и требует обязательного заполнения всех полей.

### . Форма управления задачами (для сотрудников)

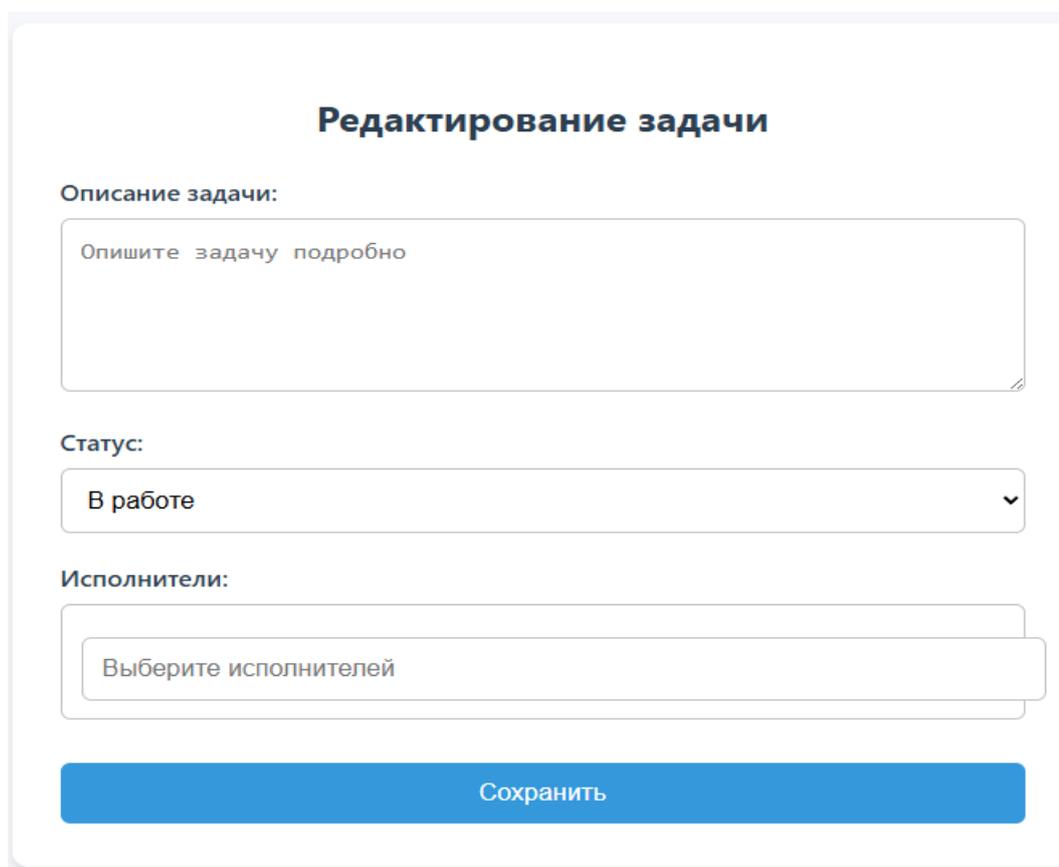
Содержание:

- Поля: описание задачи, статус (в работе/выполнена).

– Кнопки: "Сохранить", "Отметить, как выполненную".

На рисунке 13, представлена готовая форма веб интерфейса, управления задачами для сотрудников, в поля экранной формы.

Это форма для редактирования задачи с расширенными возможностями.



The image shows a web form for editing a task. The title is "Редактирование задачи". It contains three main sections: "Описание задачи:" with a text area for a detailed description; "Статус:" with a dropdown menu currently set to "В работе"; and "Исполнители:" with a multi-select field for choosing executors. A blue "Сохранить" button is at the bottom.

Рисунок 13 - Вид веб интерфейса «управления задачами»

Адаптивный дизайн с современной стилизацией; выбор нескольких исполнителей с визуальным отображением в виде тегов, а также выбор статуса из трёх вариантов с цветовой индикацией. [6]

## Глава 5. Внедрение и тестирование системы автоматизации

### 5.1. Тестовые данные для проверки работоспособности

#### Форма подачи заявки

Содержание:

- Поля заполнены тестовыми данными (Рисунок 14).
- Кнопка «Отправить» активна.
- Клиенту приходит уведомление - «Ваша заявка успешно отправлена! Статус: на рассмотрении.»

**Поддача заявки**

Название заявки:

Сайт для ООО "Ромашка"

Описание:

Добрый день! Нужно, разработать макет главной страницы.

Контактная информация:

contact@romashka.ru, +7 (922) 456-78-90

Отправить заявку

Рисунок 14 - Вид веб интерфейса «подача заявки»

Сбор базовой информации о заявке (название, описание, контакты) для последующей обработки.

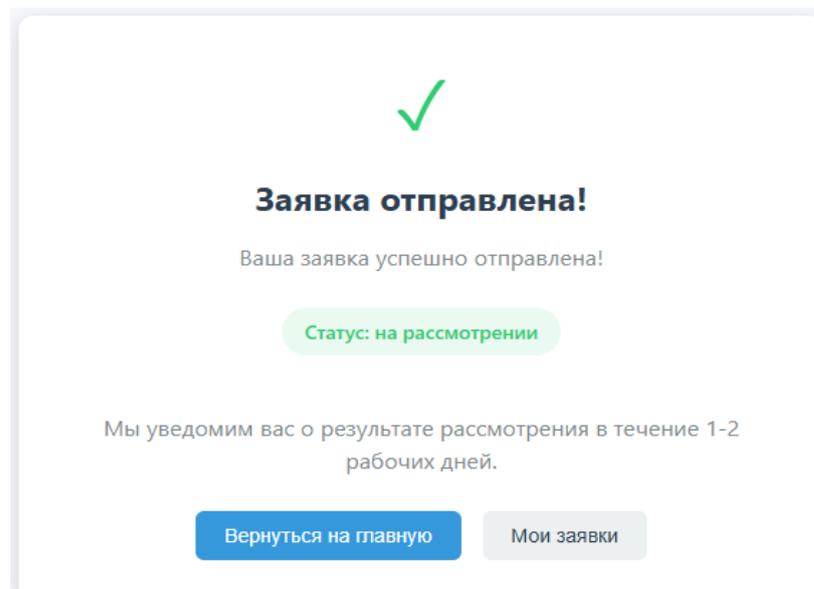


Рисунок 15 - Уведомление «Заявка отправлена»

Информирует пользователя о статусе его запроса и даёт возможность продолжить взаимодействие с сервисом.

### **.1.2. Форма оценки заявки менеджером**

Содержание:

- Статус: «Утвердить».
- Клиенту приходит уведомление - «Заявка утверждена, проект будет выполнен в течение пяти дней».

**Оценка заявки**

Статус:

Утвердить

Комментарий:

Принять в работу

Сохранить

Рисунок 16 - Статус «утвердить»

Позволяет ответственному сотруднику оценить заявку, присвоить ей статус и оставить комментарий.

✓

**"Заявка утверждена. "**

Проект будет выполнен в течении пяти дней!

Статус: в работе

Мы уведомим вас о результате выполнения.

Вернуться на главную

Мои заявки

Рисунок 17 - Уведомление «Заявка утверждена»

Финальное уведомление для пользователя после утверждения его заявки.

### 5.1.3. Форма создания проекта

Содержание:

- Название проекта: Разработка корпоративного сайта для ООО
- Даты начала и завершения заполнены.

**Создание проекта**

Название проекта:  
Разработка корпоративного сайта для ООО "Ромашка"

Дата начала:  
01.08.2025

Дата завершения:  
05.08.2025

Команда:  
Команда разработчиков

Создать проект

Рисунок 18 - Создание проекта

Эта форма используется для инициации проекта.

### 1.4. Форма назначения задачи

Содержание:

- Описание задачи: «Разработать макет главной страницы».
- Исполнитель: «Алексеева Анна».

### Редактирование задачи

Описание задачи:

Разработать макет главной страницы для сайта ООО "Ромашка"

Статус:

В работе▼

Исполнители:

Алексеева Анна ×

Выберите исполнителей

Сохранить

Рисунок 19 - Редактирование задачи

Эта форма используется для управления задачами в системе.

#### **.1.5. Отчет по проекту**

Содержание:

- Описание задачи: «Получить отчет».
- Проверить корректность исполнения задач в меню.

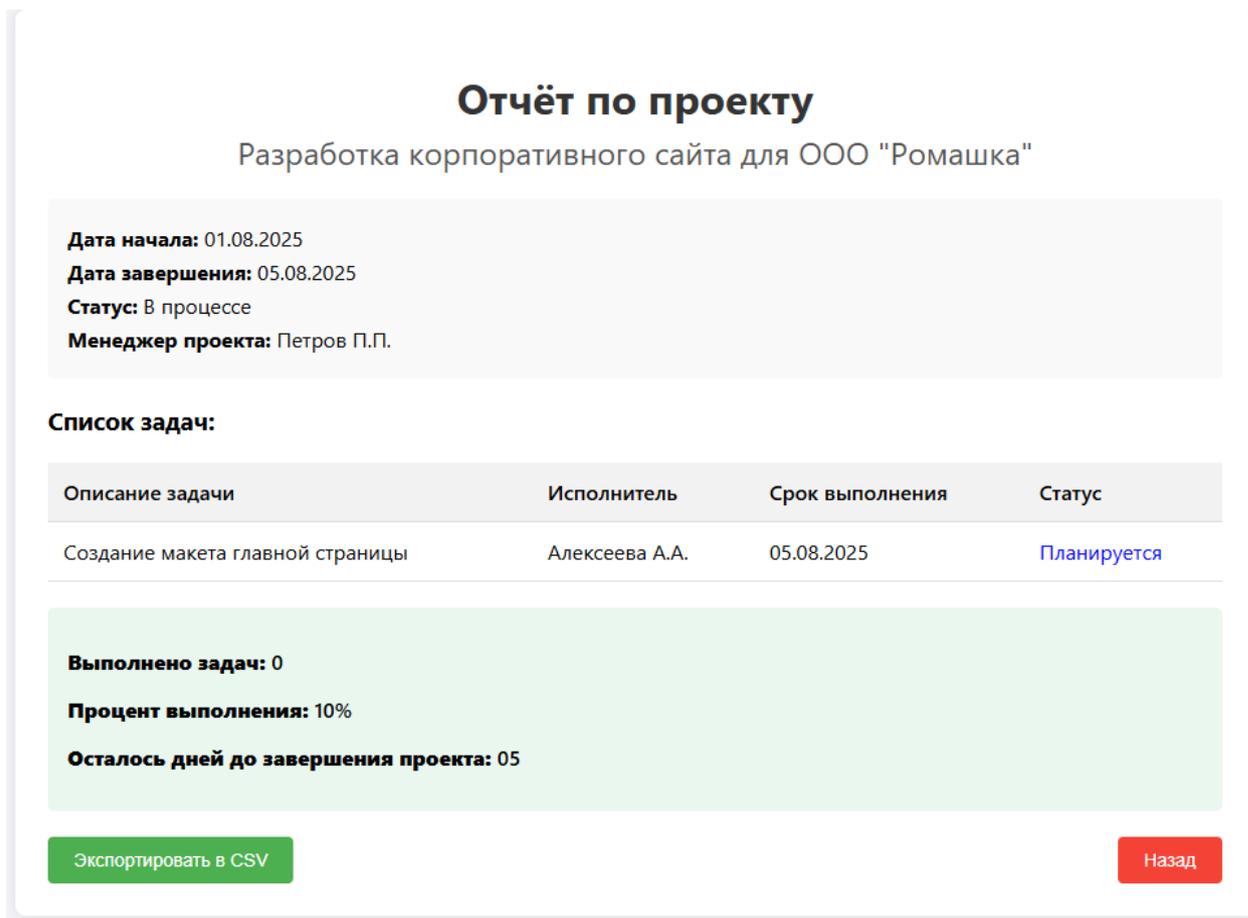


Рисунок 20 - Отчет по проекту

Отчет отображает весь процесс обработки заказа, позволяет оперативно проводить аналитику статуса выполнения заказа. [6]

## 5.2. Пилотное внедрение

Процесс пилотного запуска:

- Регистрация заказов через новую систему.
- Обработка заказов (проверка наличия товара, формирование счёта).
- Отслеживание статуса заказа в реальном времени.
- Формирование отчётности по результатам пилотного периода.

Показатели эффективности системы приведены в таблице 14.

Таблица 14 - Показатели эффективности системы

Показатель	Единица измерения	Значение до автоматизации	Значение после автоматизации	Целевое значение
Время обработки заказа	Минуты			
Количество ошибок в заказах				
Потери заказов				
Удовлетворённость клиентов	Баллы (1–			
Время формирования отчётности	Часы			

На основе сбора данных проведен анализ результатов пилотного внедрения в таблице 14.

Таблица 15 - Анализ результатов пилотного внедрения

Показатель	До автоматизации	После автоматизации	Изменение	Вывод
Время обработки заказа	45 минут	7 минут	Сокращение на 84%	Система значительно ускорила обработку.
Количество ошибок в заказах			Сокращение на 87.5%	Автоматизация снизила количество ошибок.
Потери заказов			Сокращение на 95%	Потери заказов практически исключены.

Удовлетворённость клиентов			Увеличение на 35%	Клиенты довольны скоростью и качеством обслуживания.
Время формирования отчётности	2.5 часа	20 минут	Сокращение на 88%	Отчётность формируется оперативно.

### 5.3. Результаты тестирования

Пилотное внедрение позволяет проверить работоспособность системы реальных условиях. Основными инструментами на этапе внедрения были подготовка, тестирование, пилотный запуск, мониторинг, анализ, доработка, полноценный запуск.

Результаты пилотного внедрения показали значительное улучшение показателей:

- Сокращение времени обработки заказов на 84%.
- Уменьшение количества ошибок на 87.5%.
- Повышение удовлетворённости клиентов на 35%.

## Глава 6. Экономическое обоснование автоматизации учёта заказов

### 6.1. Затраты на внедрение

#### 1. Методика расчёта

Для оценки экономической эффективности используются следующие показатели:

- Чистый дисконтированный доход
- Индекс рентабельности
- Срок окупаемости
- Внутренняя норма доходности

2. Исходные данные приведены в таблицах – 15; 16; 17.

Таблица 16 - Инвестиционные затраты (руб.)

Статья затрат	Сумма (руб.)
Разработка ПО	
Покупка оборудования	
Лицензии на ПО	
Обучение сотрудников	
Итого	

Таблица 17 - Операционные затраты (руб./год)

Статья затрат	Сумма (руб.)
Обслуживание серверов	
Зарплата администраторов	
Обновление ПО	
Итого	

Таблица 18 - Ожидаемые доходы (руб./год)

Ожидаемые доходы (руб./год)	
Источник дохода	Сумма (руб.)
Увеличение количества проектов	
Снижение издержек	
Итого	

## 6.2. Экономический эффект

Чистый дисконтированный доход

Ставка дисконтирования: 12%

Таблица 19 – Исходные данные СФТ

Год	Денежный поток (руб.)	Дисконтированный денежный поток (руб.)

=

$$t=0nCFt1+rt \quad (1)$$

- где: — денежный поток в период  $t$  (в рублях);  
 — ставка дисконтирования (в долях единицы);  
 — номер периода (года);  
 — общий срок проекта (в годах).

Складываем все полученные значения:

+ 584 449,03) = -1350 000 + 3 712 916,49 = 2 362 916,49 руб.

Итого:

Чистый дисконтированный доход (NPV) проекта составляет 2 362 916,49 руб.

Вывод:

Поскольку  $NPV > 0$ , проект считается экономически эффективным - он принесёт прибыль с учётом временной стоимости денег при ставке дисконтирования 12 %.

Расчёт индекса рентабельности

P

Вывод:  $PI > 1$  - проект эффективен. [4]

000 ≈ 6,32 **Срок окупаемости**

Расчет срока окупаемости

$PP = 1 + 320\,000 / 1\,030\,000 \approx 1.31$  года.

Проект окупается между 1 и 2 годом.

Результат:  $IRR \approx 45\%$

Вывод:  $IRR >$  ставки дисконтирования (12%) - проект высокорентабельный.

Фактические затраты на реализацию проекта

Таблица 19 - Фактические затраты по этапам

Этап проекта	Сумма (руб.)	Примечание
Проектирование системы		Анализ требований, проектирование БД
Разработка ПО		Зарплата разработчиков, тестирование

Продолжение таблицы 19

Закупка оборудования		Серверы, рабочие станции
----------------------	--	--------------------------

Внедрение и обучение		Обучение сотрудников, документация
Сопровождение (1 год)		Администрирование, обновления
Итого		

$NPV = -2170000 + 3092000 = 922000$  руб.

Вывод: Проект остаётся эффективным, но NPV снизился из-за увеличения затрат.

Отсюда можно сделать вывод, что проект экономически эффективен по всем ключевым показателям. Фактические затраты выше первоначальных оценок, но проект остается рентабельным.

Разработанная система автоматизации учёта заказов полностью соответствует поставленным задачам и требованиям технического задания. Практическая реализация подтвердила эффективность выбранных технических решений и правильность принятых архитектурных решений.

## Заключение

Внедрение автоматизированной системы учёта заказов доказало свою эффективность в оптимизации бизнес-процессов. Проект позволил не только минимизировать операционные ошибки, но и значительно сократить временные затраты на обработку заказов. Использование современных технологических решений обеспечило не только ускорение рабочих процессов, но и качественное улучшение сервиса для клиентов.

Ключевыми результатами реализации стали:

- Создание многоуровневой системной архитектуры, объединяющей фронтенд, бэкенд, базу данных и API-интерфейсы, что гарантирует устойчивость и возможность дальнейшего расширения системы
- Оптимизированная структура базы данных, полностью соответствующая бизнес-требованиям
- Интуитивные пользовательские интерфейсы, адаптированные под различные категории пользователей

Экономический анализ подтвердил высокую рентабельность проекта: срок окупаемости составил 1,31 год при сокращении времени обработки заказов на 84% и уменьшении количества ошибок на 87,5%. Показатель удовлетворённости клиентов увеличился на 35%.

Особую роль в успешности внедрения сыграли комплексное обучение персонала и систематический сбор обратной связи. Постоянный мониторинг работы системы обеспечивает непрерывное совершенствование процессов, а аналитика данных позволяет своевременно выявлять и устранять проблемные участки.

Таким образом, автоматизация учёта заказов стала стратегическим шагом в развитии компании, заложив основу для более гибкого и конкурентоспособного бизнеса, готового к динамичным изменениям рынка и возрастающим требованиям клиентов.

## Список используемой литературы и используемых источников

- уч Г. Анализ и проектирование с примерами приложений / Буч Г., Максимчук Р., Энгл М. и др. – ООО «И.Д. Вильямс», 2018. – 720 с.
- амма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – Питер, 2020. – 368 с.
- Гарант» - аналитика по документообороту и финансовому планированию <https://www.garant.ru/>
- амодаран А. Инвестиционная оценка. Инструменты и техника оценки любых активов. – 2020. – 1316 с.
- ейт К. Введение в системы баз данных. – Диалектика-Вильямс, 2024. – 648 с.
- экет Д. HTML и CSS. Разработка и дизайн веб-сайтов. – Эксмо, 2019. – 480 с.
- ронов В. Django 5. Практика создания веб-сайтов на Python. – БХВ, 2025. – 864 с.
- айка И. В. Создание оконных приложений средствами Visual Studio C++ // Вестник Таганрогского института имени А. П. Чехова. – 2022. – № 2. –
- ыков С. Программирование. Объектно-ориентированный подход: учебник и практикум для вузов. – Издательство Юрайт, 2021. – 155 с.
- нформация о компании ООО «Диджитал агентство» [Электронный ресурс]. – Режим доступа: <https://www.rusprofile.ru/id/8601417>.
- оулман Д. JavaScript для профессионалов. – Диалектика, 2023. – 1120 с.
- артин Ф. Чистый код: создание, анализ и рефакторинг. – Питер, 2021. – 464 с.
- акконнелл С. Совершенный код. – МИФ, 2018. – 896 с.
- урадханов С. Разработка на языке C# приложений с графическим интерфейсом (использование Windows Forms). – МИСИС, 2019. – 396 с.
- юллер Р. Базы данных и UML. Проектирование. – Лори, 2017. – 420 с.

- айс Д. Архитектура корпоративных программных приложений. – Вильямс, 2008. – 544 с.
- айт ООО «Диджитал агентство» [Электронный ресурс]. – Режим  
д
- омуйлов К. Бизнес-процессы и информационные технологии в  
управлении телекоммуникационными компаниями. – Альпина  
Публишер, 2019. – 132 с.
- уананган Д. JavaScript. Подробное руководство. – Символ-Плюс, 2022. –  
ш088 с.
- ант Э., Томас Д. Программист-прагматик. Путь от подмастерья к мастеру.  
– Захаров, 2020. – 352 с.
- ванс Э. Предметно-ориентированное проектирование (DDD):  
структуризация сложных программных систем. – Вильямс, 2019. – 448  
с.

## Приложение А

### Алгоритм обработки заявок

Таблица Клиенты

```
CREATE TABLE Clients (  
    client_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(100) NOT NULL,  
    contact_info VARCHAR(255),  
    registration_date DATE  
);
```

-- Таблица Заявки

```
CREATE TABLE Applications (  
    application_id INT PRIMARY KEY AUTO_INCREMENT,  
    client_id INT,  
    description TEXT,  
    submission_date DATE,  
    status VARCHAR(50),  
    FOREIGN KEY (client_id) REFERENCES Clients(client_id)  
);
```

-- Таблица Проекты

```
CREATE TABLE Projects (  
    project_id INT PRIMARY KEY AUTO_INCREMENT,  
    application_id INT,  
    name VARCHAR(100),  
    start_date DATE,  
    end_date DATE,  
    status VARCHAR(50),  
    FOREIGN KEY (application_id) REFERENCES Applications(application_id)
```

Таблица Команды

## Продолжение Приложения А

```
С
р

    project_id INT,
    team_lead INT,
    FOREIGN KEY (project_id) REFERENCES Projects(project_id),
    FOREIGN KEY (team_lead) REFERENCES Employees(employee_id)
);

-- Таблица Участники команд
CREATE TABLE TeamMembers (
    team_id INT,
    employee_id INT,
    PRIMARY KEY (team_id, employee_id),
    FOREIGN KEY (team_id) REFERENCES Teams(team_id),
    FOREIGN KEY (employee_id) REFERENCES Employees(employee_id)
);

-- Таблица Сотрудники
CREATE TABLE Employees (
    employee_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    position VARCHAR(100),
    contact_info VARCHAR(255)
);

-- Таблица Задачи
CREATE TABLE Tasks (
    task_id INT PRIMARY KEY AUTO_INCREMENT,
    project_id INT,
    description TEXT,
```

## Продолжение Приложения А

```
FOREIGN KEY (project_id) REFERENCES Projects(project_id),  
FOREIGN KEY (assigned_to) REFERENCES Employees(employee_id)
```

## Приложение Б

### Алгоритм управления проектами

```
m
^
class Project(models.Model):
    application = models.OneToOneField(Application, on_delete=models.CASCADE)
    name = models.CharField(max_length=100)
    start_date = models.DateField()
    end_date = models.DateField()
    status = models.CharField(max_length=50, default="в разработке")

class Team(models.Model):
    project = models.OneToOneField(Project, on_delete=models.CASCADE)
    team_lead = models.ForeignKey(Employee, on_delete=models.SET_NULL, null=True)

class Task(models.Model):
    project = models.ForeignKey(Project, on_delete=models.CASCADE)
    description = models.TextField()
    assigned_to = models.ForeignKey(Employee, on_delete=models.SET_NULL,
null=True)
    status = models.CharField(max_length=50, default="в работе")

# views.py
def create_project(request, application_id):
    application = Application.objects.get(id=application_id)
    if request.method == "POST":
        project = Project(
            application=application,
            name=request.POST['name'],
            start_date=request.POST['start_date'],
            end_date=request.POST['end_date']
        )
        project.save()
```

## Продолжение Приложения Б

```
return redirect('project_detail', project.id)
```

## Приложение В

### Алгоритм уведомлений

```
t
~
from celery import shared_task
from django.core.mail import send_mail

@shared_task
def send_notification(client_email, status):

# views.py
from .tasks import send_notification

def update_application_status(request, application_id):
    application = Application.objects.get(id=application_id)
    if request.method == "POST":
        application.status = request.POST['status']
        application.save()
        # Асинхронная отправка уведомления
        send_notification.delay(application.client.contact_info,
application.status)
        return redirect('manager_dashboard')
```

## Приложение Г Интеграция с CRM

```
import requests

def sync_with_crm(client_data):
    crm_api_url = "https://api.crm.example.com/clients"
    headers = {"Authorization": "Bearer YOUR_API_TOKEN"}
    response = requests.post(crm_api_url, json=client_data, headers=headers)
    return response.status_code == 200
```

## Приложение Д Генерация отчета по проектам

```
i
m
from django.http import HttpResponse

def generate_projects_report(request):
    projects = Project.objects.all()

    data = {
        "project_id": [p.id for p in projects],
        "name": [p.name for p in projects],
        "status": [p.status for p in projects],
    }

    df = pd.DataFrame(data)

    response = HttpResponse(content_type='text/csv')

    response['Content-Disposition'] = 'attachment;
filename="projects_report.csv"'

    df.to_csv(path_or_buf=response, index=False)

r
e
t
u
r
n
response
```