



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Экономики и управления на предприятии природопользования»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(бакалаврская работа)
по направлению подготовки 09.03.03 Прикладная информатика
(квалификация – бакалавр)

На тему «Разработка веб - приложения «Мини - CRM для малого бизнеса»

Исполнитель Кешишян Григорий Мисакович

Руководитель к.т.н., Сафонова Татьяна Владимировна

«К защите допускаю»

Руководитель кафедры

кандидат экономических наук

Майборода Евгений Викторович

«22» 06 2026 г.

Филиал Российского государственного гидрометеорологического университета в г. Туапсе	
НОРМОКОНТРОЛЬ ПРОЙДЕН	
«22» 06 2026 г.	
 ПОДПИСЬ	РАСШИФРОВКА ПОДПИСИ

Туапсе
2026

ОГЛАВЛЕНИЕ

Введение.....	4
1 Аналитическая часть.....	8
1.1 Анализ предметной области. Теоретические основы CRM-систем	8
1.2 Характеристика объекта исследования	12
1.3 Обоснование выбора задачи. Анализ существующих CRM-систем	16
1.4 Экономико-информационная сущность задачи.....	19
2 Проектная часть.....	24
2.1 Информационное обеспечение задачи.....	24
2.2 Технологическое обеспечение	28
2.2.1 Обоснование проектных решений по технологическому обеспечению	28
2.2.2 Технологические операции	32
2.3 Техническое обеспечение.....	35
2.3.1 Обоснование проектных решений по техническому обеспечению....	35
2.3.2 Комплекс технических средств	37
2.4 Программное обеспечение задачи.....	37
2.4.1 Обоснование проектных решений по программному обеспечению ..	37
2.4.2 Архитектура программного обеспечения.....	40
2.4.3 Описание программных модулей	42
2.5 Руководство пользователя.....	44
2.5.1 Описание интерфейса	44
2.5.2 Порядок работы с системой	47
3 Обоснование экономической эффективности результатов ВКР.....	50
3.1 Выбор и обоснование методики расчёта экономической эффективности.....	50
3.2 Расчёт показателей экономической эффективности	52
3.2.1 Расчёт себестоимости разработки программного средства.....	52
3.2.2 Расчёт экономического эффекта от внедрения.	56

Заключение	60
Список использованных источников	63
Приложение	66

Введение

Цифровая трансформация бизнес-процессов является одной из ключевых тенденций современной экономики. Автоматизация рутинных операций, централизация данных и обеспечение прозрачности процессов становятся обязательными условиями конкурентоспособности даже для небольших компаний. В этом контексте системы управления взаимоотношениями с клиентами (CRM) перестают быть привилегией крупного бизнеса и превращаются в инструмент первой необходимости для любой организации, работающей с клиентским потоком.

По данным исследования компании Salesforce «State of Sales» (2023), 74% менеджеров по продажам, использующих CRM-системы, считают, что они помогают им лучше понимать потребности клиентов. Аналитики платформы HubSpot фиксируют, что внедрение CRM повышает продуктивность отдела продаж в среднем на 34%, а точность прогнозирования выручки — на 42%. Российский рынок в данном отношении демонстрирует отставание: по данным аналитического агентства CNews, в 2023–2024 годах лишь 18–22% отечественных малых предприятий применяли специализированные CRM-системы.

Однако специфика российского рынка CRM в 2022–2025 годах существенно изменилась. Уход крупных западных вендоров (Salesforce, Microsoft Dynamics), ограничения на использование зарубежных облачных платформ для хранения персональных данных российских граждан, а также требования Федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных» [1] к локализации данных сформировали устойчивый спрос на отечественные self-hosted решения. При этом существующие российские платформы — Битрикс24, amoCRM — либо финансово избыточны для малого бизнеса, либо содержат функционал, невостребованный небольшими командами.

Таким образом, сложилась ситуация рыночной ниши: малые

предприятия, включая небольшие подразделения ИТ-компаний, нуждаются в простом, бесплатном, полностью русскоязычном и развёртываемом на собственном сервере инструменте управления клиентскими взаимодействиями. Данная выпускная квалификационная работа направлена на закрытие этой потребности путём разработки специализированного веб-приложения «Мини-CRM для малого бизнеса».

Актуальность темы подтверждена в ходе преддипломной практики на базе туапсинского филиала ООО ИК «СИБИНТЕК» — крупнейшего ИТ-аутсорсера России, дочернего предприятия группы «Роснефть». В ходе практики установлено, что, несмотря на наличие корпоративной системы Service Desk, сотрудники регионального подразделения фактически ведут оперативный учёт клиентских обращений в разрозненных таблицах Microsoft Excel и электронной почте. Это приводит к потерям обращений, значительным временным затратам на поиск истории взаимодействий и низкой прозрачности текущей загрузки.

Объектом исследования является туапсинский филиал ООО ИК «СИБИНТЕК» — информационно-технологическая компания в сфере ИТ-аутсорсинга.

Предметом исследования выступает процесс управления взаимодействием с клиентами в условиях небольшого регионального подразделения.

Цель работы — разработка веб-приложения «Мини-CRM для малого бизнеса», автоматизирующего процессы ведения клиентской базы, управления сделками и задачами, формирования оперативной отчётности.

Для достижения поставленной цели необходимо решить следующие задачи:

- исследовать теоретические основы систем управления взаимоотношениями с клиентами, изучить их классификацию и особенности применения в малом бизнесе;

- изучить нормативно-правовую базу Российской Федерации в области обработки и защиты персональных данных в информационных системах;
- провести анализ объекта исследования: описать специфику деятельности туапсинского филиала ООО ИК «СИБИНТЕК», его организационную структуру и текущую степень автоматизации;
- выявить и количественно оценить недостатки существующей технологии управления клиентскими взаимодействиями;
- провести сравнительный анализ существующих CRM-решений на российском рынке и обосновать необходимость собственной разработки;
- разработать функциональные модели бизнес-процессов в нотациях IDEF0 (AS-IS) и BPMN 2.0 (TO-BE);
- спроектировать реляционную базу данных (ER-диаграмма) и описать сценарии взаимодействия пользователей с системой (UML Use-Case);
- реализовать веб-приложение в соответствии с выявленными функциональными и нефункциональными требованиями;
- разработать руководство пользователя с описанием интерфейса и порядком работы;
- рассчитать технико-экономические показатели и срок окупаемости разработанной системы по нормативной методике.

Теоретической основой исследования послужили труды отечественных и зарубежных авторов в области информационных систем, управления взаимоотношениями с клиентами и веб-разработки [2–6, 11, 21–23].

Нормативной основой являются: Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных» [1]; Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации» [6]; Постановление Правительства РФ от 01.11.2012 № 1119 [24].

Методологической базой послужили: метод структурного функционального моделирования (IDEF0); нотация описания бизнес-процессов BPMN 2.0; методология объектно-ориентированного анализа (UML Use-Case, ER-диаграмма); метод сравнительного анализа программных продуктов;

нормативная методика расчёта экономической эффективности разработки программных средств.

Практическая значимость работы состоит в том, что разработанное приложение представляет собой готовое к развёртыванию программное решение, не требующее лицензионных затрат. Возможность self-hosted развёртывания на отечественном VPS-сервере гарантирует соответствие требованиям ФЗ-152 о локализации персональных данных. Система не привязана к специфике конкретного предприятия и может быть тиражирована для широкого круга малых организаций.

Структура выпускной квалификационной работы включает: введение; первый раздел (аналитическая часть); второй раздел (проектная часть); третий раздел (обоснование экономической эффективности); заключение; список использованных источников (30 наименований); приложения (А—Е).

1 Аналитическая часть

1.1 Анализ предметной области. Теоретические основы CRM-систем

Понятие и эволюция CRM. Термин CRM (Customer Relationship Management — управление взаимоотношениями с клиентами) трактуется в двух взаимосвязанных аспектах. В широком смысле CRM обозначает бизнес-стратегию, направленную на построение и поддержание долгосрочных взаимовыгодных отношений с клиентами. В узком, прикладном смысле под CRM понимается класс информационных систем, автоматизирующих процессы управления клиентской базой, продажами, обращениями и взаимодействиями [2].

Концепция управления взаимоотношениями с клиентами уходит корнями в теорию маркетинга отношений (Relationship Marketing), разработанную Леонардом Берри в 1983 году. В 1990-х годах, с развитием корпоративных информационных систем, концепция трансформировалась в отдельный класс программного обеспечения. Первые системы автоматизации продаж (SFA — Sales Force Automation) постепенно расширили функциональность и к 2000-м годам превратились в полноценные CRM-платформы, охватывающие маркетинг, продажи и сервисное обслуживание.

Аналитики компании Gartner, впервые применившие термин «CRM» в современном значении в конце 1990-х годов, определяют его как «бизнес-стратегию, ориентированную на привлечение и удержание клиентов с максимальной выгодой для организации». При этом технологическая составляющая является инструментом реализации стратегии, а не самоцелью.

Функциональность CRM-систем. С функциональной точки зрения современные CRM-системы решают следующие задачи: централизованное хранение и структурирование информации о клиентах; автоматизация процессов продаж (управление воронкой, сделками, коммерческими предложениями); управление задачами и поручениями; ведение полной истории взаимодействий с клиентом (письма, звонки, встречи); аналитика

ключевых показателей деятельности (KPI); интеграция с внешними каналами коммуникации (e-mail, мессенджеры, телефония) [3].

Классификация CRM-систем. По функциональному назначению принято выделять три основных типа CRM-систем.

Операционные CRM автоматизируют повседневные операции: регистрацию обращений, управление контактами и сделками, назначение задач, ведение истории. Этот тип CRM ориентирован на поддержку конкретных бизнес-процессов отдела продаж и клиентского сервиса. Именно к этому классу относится разрабатываемое в данной работе приложение.

Аналитические CRM ориентированы на глубокий анализ накопленных данных о клиентах: сегментацию клиентской базы, прогнозирование поведения, оценку жизненного цикла клиента (LTV — Lifetime Value), анализ эффективности маркетинговых кампаний. Аналитические CRM, как правило, используют технологии Business Intelligence и Data Mining.

Коллаборативные CRM обеспечивают взаимодействие с клиентами через различные каналы и координацию между подразделениями компании. Особенностью этого типа является омниканальность: клиент обслуживается единообразно вне зависимости от канала обращения (телефон, e-mail, мессенджер, чат на сайте) [4].

По способу развёртывания CRM-системы классифицируются на облачные (SaaS) и локальные (on-premise, self-hosted). Облачные решения предоставляются по модели «программное обеспечение как услуга»: организация получает доступ к системе через браузер, не заботясь об инфраструктуре. Однако данные хранятся на серверах вендора, что при работе с персональными данными российских граждан может вступать в противоречие с требованиями ФЗ-152 о локализации. Локальные решения развёртываются на собственном сервере организации, что обеспечивает полный контроль над данными, но требует технических компетенций для поддержки.

По масштабу целевой аудитории CRM-системы делятся на: корпоративные (Salesforce, SAP CRM, Microsoft Dynamics 365),

ориентированные на предприятия с тысячами сотрудников и сложными процессами; платформы для среднего бизнеса (Битрикс24, amoCRM), рассчитанные на десятки и сотни пользователей; решения для малого бизнеса и индивидуальных предпринимателей — класс, представленный значительно скромнее [4].

Специфика применения CRM в малом бизнесе существенно отличается от корпоративного сектора. Малый бизнес, как правило, располагает командой не более 15–20 человек, ограниченным ИТ-бюджетом (или его полным отсутствием), отсутствием выделённого системного администратора и низким порогом терпимости к сложности внедрения. Исследования показывают, что 60% неудачных внедрений CRM в малом бизнесе связаны не с техническими проблемами, а с избыточной сложностью системы и сопротивлением персонала [5].

Это означает, что CRM-система для малого бизнеса должна удовлетворять следующим критериям: минимальное время освоения без специального обучения; работа через браузер без установки дополнительного ПО; отсутствие или минимальная стоимость лицензии; интуитивно понятный русскоязычный интерфейс; возможность развёртывания на доступном по цене сервере; минимальный набор функций, точно соответствующий потребностям (без «раздутого» функционала).

Нормативно-правовое регулирование. Разработка информационной системы, осуществляющей сбор и обработку персональных данных физических лиц, неизбежно затрагивает область правового регулирования.

Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных» [1] является основополагающим нормативным актом в данной сфере. Статья 3 определяет персональные данные как «любую информацию, относящуюся к прямо или косвенно определённому или определяемому физическому лицу». В контексте CRM-системы к персональным данным относятся: фамилия, имя, отчество контактного лица; номер телефона; адрес электронной почты;

почтовый адрес. Следовательно, система является оператором персональных данных и обязана соответствовать требованиям ФЗ-152.

Статья 18.1 ФЗ-152, введённая Федеральным законом от 21.07.2014 № 242-ФЗ, установила требование локализации: обязанность операторов обеспечивать хранение и обработку персональных данных российских граждан с использованием баз данных, физически расположенных на территории Российской Федерации. Это требование непосредственно влияет на выбор инфраструктуры: облачные сервисы зарубежных провайдеров (AWS, Google Cloud) не обеспечивают размещение данных исключительно в российских дата-центрах, тогда как self-hosted развёртывание на отечественном VPS-хостинге полностью исключает соответствующие риски.

Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации» [6] устанавливает общие требования к защите информации в информационных системах. Статья 16 обязывает применять правовые, организационные и технические меры защиты. К техническим мерам, реализованным в разрабатываемой системе, относятся: хэширование паролей алгоритмом bcrypt с коэффициентом сложности 12; аутентификация на основе JWT-токенов с ограниченным сроком действия; ролевая модель разграничения доступа (администратор / менеджер / наблюдатель); журнал аудита всех действий пользователей.

Постановление Правительства РФ от 01.11.2012 № 1119 [24] классифицирует информационные системы персональных данных по уровням защищённости (УЗ-1—УЗ-4). Разрабатываемая система, обрабатывающая данные контактных лиц клиентов без специальных категорий, относится к уровню защищённости УЗ-4 с минимальным набором требований: обеспечение режима безопасности помещения, где размещена система; использование средств защиты, прошедших оценку соответствия; назначение ответственного за безопасность.

Таким образом, теоретические основы и нормативная база формируют конкретные требования к технической реализации системы, которые полностью

учтены в архитектуре разрабатываемого приложения. В частности, выбор self-hosted развёртывания обусловлен не только стоимостными факторами, но и требованиями законодательства о локализации персональных данных.

1.2 Характеристика объекта исследования

Общие сведения о предприятии. Объектом исследования является туапсинский филиал ООО ИК «СИБИНТЕК» (Сибирская Интернет Компания). Компания основана 26 августа 1999 года и зарегистрирована в г. Москве по адресу: 117638, г. Москва, Севастопольский проспект, 28, стр. 1. Основной государственный регистрационный номер (ОГРН): 1027700270023. ИНН: 7724249557. Уставный капитал составляет 244 963 000 рублей. Учредителями являются ООО «РН-Трейд» (51%) и ООО «РН-Актив» (49%) — дочерние структуры группы компаний «Роснефть» [7].

Деятельность компании. Основной вид деятельности по ОКВЭД — 63.11.1 «Деятельность по созданию и использованию баз данных и информационных ресурсов». Кроме того, ООО ИК «СИБИНТЕК» осуществляет деятельность по более чем 30 направлениям, включая: предоставление услуг связи и передачи данных; телематические услуги; монтаж и обслуживание средств охранной и пожарной безопасности; разработку шифровальных средств защиты информации; техническую защиту конфиденциальной информации; деятельность в области информационных технологий.

По масштабу деятельности ООО ИК «СИБИНТЕК» входит в число крупнейших ИТ-аутсорсеров России: компания обслуживает информационно-технологическую инфраструктуру предприятий нефтяной отрасли. Общество располагает сетью из семи региональных филиалов, расположенных в ключевых нефтедобывающих и нефтеперерабатывающих регионах России.

Туапсинский филиал. Туапсинский филиал ООО ИК «СИБИНТЕК» расположен в г. Туапсе Краснодарского края и обслуживает корпоративных

клиентов региона. Основным и крупнейшим заказчиком является ООО «РН-Туапсенефтепродукт» — оператор Туапсинского нефтеперерабатывающего завода, входящего в производственный контур НК «Роснефть». Нефтеперерабатывающий завод в Туапсе является одним из старейших в России (основан в 1929 году) и перерабатывает западносибирские нефти, поступающие по трубопроводу.

Деятельность туапсинского филиала охватывает четыре основных направления. Сопровождение ИТ-инфраструктуры заказчиков включает мониторинг работоспособности серверного оборудования, сетевых устройств и телекоммуникационных систем, проведение регламентных работ и плановых обновлений. Техническая поддержка пользователей реализована по двухуровневой модели: специалисты первой линии принимают входящие обращения и решают типовые инциденты; нестандартные ситуации эскалируются на вторую линию поддержки с участием более квалифицированных специалистов. Администрирование корпоративных систем охватывает поддержку Active Directory, корпоративных приложений и баз данных заказчиков. Организация телекоммуникационных решений включает проектирование, монтаж и обслуживание структурированных кабельных систем, ВОЛС и беспроводных сетей.

Специфика экономической области. Объект исследования функционирует на рынке корпоративных ИТ-услуг и ИТ-аутсорсинга. Данный рынок в России демонстрирует устойчивый рост: по данным РАЭК, объем рынка ИТ-аутсорсинга вырос с 180 млрд рублей в 2021 году до 260 млрд рублей в 2023 году. Нефтяная отрасль является традиционно крупнейшим потребителем ИТ-услуг в России.

Организационная структура туапсинского филиала носит линейно-функциональный характер. Линейно-функциональная структура предполагает, что линейные руководители принимают решения и отдают распоряжения, а функциональные подразделения обеспечивают их информационной и методической поддержкой. Данный тип структуры оптимален для организаций

численностью 15–50 человек с устойчивыми, повторяющимися процессами.

Структура включает четыре подразделения (рисунок 1.1):

- отдел технической поддержки пользователей — включает специалистов первой линии поддержки (приём обращений, решение типовых инцидентов) и специалистов второй линии поддержки (решение нестандартных инцидентов, эскалация к разработчикам);
- отдел сетевой инфраструктуры — сетевые инженеры и системные администраторы, ответственные за работоспособность сетей и серверного оборудования клиентов;
- отдел по работе с клиентами — координирует взаимодействие с заказчиками на уровне договоров, SLA и крупных проектов;
- административный персонал — обеспечивает управленческие, кадровые и бухгалтерские функции.



Линейно-функциональная структура - 4 подразделения - ~25–30 сотрудников

Рисунок 1.1 - Организационная структура туапсинского филиала ООО ИК «СИБИНТЕК»

Численность персонала туапсинского филиала составляет около 25–30 сотрудников. Непосредственно с клиентскими обращениями работают сотрудники отдела технической поддержки (8–10 человек) и отдела по работе с клиентами (3–4 человека).

Степень автоматизации деятельности. Рабочие станции всех сотрудников подключены к корпоративной локальной сети с централизованным

управлением через Active Directory. Сетевая инфраструктура включает: файловый сервер (Windows Server 2019) для хранения корпоративных документов; корпоративный почтовый сервер (Microsoft Exchange); принтеры и МФУ с сетевым подключением.

Для управления инцидентами клиентов в структуре СИБИНТЕК существует корпоративная система Service Desk класса ITSM (IT Service Management). Данная система поддерживает полноценный жизненный цикл инцидента в соответствии с методологией ITIL (IT Infrastructure Library): регистрацию, категоризацию, приоритизацию, назначение исполнителя, эскалацию, разрешение и закрытие. Система интегрирована с корпоративной базой знаний и обеспечивает сбор метрик SLA.

Тем не менее данная система используется в основном для взаимодействия с головным офисом и крупным заказчиком — ТНЗ. Для оперативного управления небольшими заявками регионального масштаба сотрудники не используют корпоративную ITSM-систему по следующим причинам: избыточность форм и процессов для коротких типовых заявок; необходимость соблюдения корпоративных процедур согласования, требующих участия нескольких уровней управления; техническая сложность создания новых категорий и шаблонов для специфических задач филиала.

Система электронного документооборота в туапсинском филиале не внедрена: документы оформляются в MS Word, согласовываются по электронной почте и хранятся на сетевых папках файлового сервера. Корпоративная база знаний практически не заполняется сотрудниками филиала: типовые решения хранятся либо в личных заметках специалистов, либо в виде переписки электронной почты.

Таким образом, несмотря на высокую общую степень ИТ-автоматизации деятельности туапсинского филиала (что закономерно для ИТ-компании), оперативное управление клиентскими взаимодействиями на уровне регионального подразделения осуществляется без специализированной системы. Это создаёт предпосылки для внедрения лёгкого инструмента, не

конкурирующего с корпоративными системами, а дополняющего их на уровне ежедневной оперативной деятельности.

1.3 Обоснование выбора задачи. Анализ существующих CRM-систем

Перед проектированием собственной системы проведён детальный анализ существующих CRM-решений, представленных на российском рынке. Анализ включает как специализированные CRM-платформы, так и смежные инструменты, применяемые малым бизнесом в качестве замены CRM.

Битрикс24. Российская платформа производства компании «1С-Битрикс», запущенная в 2012 году. Представляет собой универсальный корпоративный портал, объединяющий CRM, систему управления задачами и проектами, корпоративный мессенджер, облачное хранилище, конструктор сайтов и интернет-магазинов. По данным самой компании, платформой пользуются более 15 миллионов организаций в 187 странах.

Преимущества Битрикс24: богатый функционал в едином решении; хорошая интеграция с российскими телефонными операторами; наличие коробочной версии для самостоятельного хостинга; русскоязычный интерфейс и поддержка. Недостатки с точки зрения целевой аудитории: крайняя перегруженность интерфейса, требующая многонедельного обучения; бесплатный тариф ограничен 5 гигабайтами хранилища и базовым CRM-функционалом; платные тарифы начинаются от 1 990 рублей в месяц за организацию, при этом расширенный CRM-функционал доступен от тарифа «CRM+» стоимостью 4 790 рублей в месяц [8].

amoCRM (с 2023 года — КОМО). Российская CRM-система производства компании «QSOFТ», специализированная для управления продажами и воронкой сделок. Позиционируется как «самая продаваемая CRM в России» для отделов продаж.

Преимущества amoCRM: современный и удобный интерфейс; хорошая воронка продаж; богатый API и экосистема интеграций; мобильное

приложение. Недостатки: полное отсутствие бесплатного тарифа (минимальная стоимость 599 рублей за пользователя в месяц, то есть для команды из 5 человек — не менее 2 995 рублей); ориентированность на продажный сценарий означает избыточность инструментария для сервисного учёта обращений; данные хранятся исключительно в облаке amoCRM, отсутствие self-hosted варианта [9].

HubSpot CRM. Американская платформа производства компании HubSpot Inc. (NYSE: HUBS), основанной в 2006 году. Входит в тройку крупнейших CRM-платформ мира наряду с Salesforce и Microsoft Dynamics.

Преимущества HubSpot: функциональный бесплатный тариф с неограниченным числом пользователей; современный и интуитивный интерфейс; мощная аналитика; хорошая документация. Недостатки: интерфейс и документация преимущественно на английском языке; платные тарифы начинаются от \$15 за пользователя в месяц и экспоненциально растут с расширением функционала; данные хранятся на серверах в США, что прямо нарушает требования ФЗ-152 о локализации персональных данных российских граждан [10].

Open-source решения. На рынке представлено несколько зрелых CRM-систем с открытым исходным кодом. SuiteCRM — форк популярной SugarCRM с богатым функционалом. EspoCRM — современная система с чистым REST API. Vtiger CRM — платформа с функционалом автоматизации маркетинга. Все перечисленные системы доступны бесплатно и поддерживают self-hosted развёртывание. Их основные недостатки: отсутствие официальной русификации (частичные переводы сообщества); интерфейс уровня 2012–2016 годов, значительно уступающий современным SaaS-платформам; необходимость технических компетенций для развёртывания и настройки, недоступных малому бизнесу без ИТ-специалиста в штате.

Инструменты-заменители. Практика использования Notion, Airtable и Google Sheets в качестве импровизированных CRM широко распространена среди малых команд. Эти инструменты позволяют создать базу клиентов на

основе гибких таблиц или баз данных, однако в них отсутствуют: встроенные статусные воронки с визуализацией; история изменений на уровне записи; разграничение прав по ролям; аналитический дашборд; механизм задач и напоминаний, привязанных к клиенту или сделке. Кроме того, Notion и Airtable являются зарубежными облачными сервисами и имеют те же проблемы с локализацией данных, что и HubSpot.

Сравнительный анализ рассмотренных систем по ключевым критериям приведён в таблице 1.1.

Таблица 1.1 - Сравнительный анализ CRM-систем

Критерий	Битрикс24	amoCRM	HubSpot	EspoCRM	МиниCRM (разраб.)
Бесплатный тариф	Да (ограничен)	Нет	Да (базов.)	Да (self-h.)	Да (полный)
Язык интерфейса	Русский	Русский	Английский	Частично рус.	Русский
Сложность освоения	Высокая	Средняя	Средняя	Высокая	Низкая
Мин. стоимость	~2 000 р./мес.	599 р./польз.	\$15/польз.	0 р.	0 р.
Self-hosted вариант	Нет (Коробка)	Нет	Нет	Да	Да
Данные в РФ	Частично	Нет	Нет	Да (при self-h.)	Да (гарантир.)
Актуальный UI	Да	Да	Да	Частично	Да
Генератор КП	Нет (в платн.)	Нет	Нет	Нет	Да
Индикатор риска сделки	Нет	Нет	Нет	Нет	Да
Русская документация	Да	Да	Нет	Частично	Да

Анализ рынка (таблица 1.1) подтверждает: ни одно из рассмотренных решений не является оптимальным для небольшой команды с ограниченным бюджетом, которой нужен простой русскоязычный инструмент с гарантированной локализацией данных в РФ, современным интерфейсом и

нулевой стоимостью. Это обосновывает целесообразность собственной разработки.

Конкурентные преимущества разрабатываемой системы «Мини-CRM» перед существующими аналогами:

- полная бесплатность — отсутствие лицензионных платежей принципиально для малого бизнеса с ограниченным ИТ-бюджетом;
- self-hosted развёртывание на отечественном сервере — гарантия соответствия ФЗ-152 и полный контроль над данными;
- встроенный шаблонный генератор коммерческих предложений — функция, отсутствующая в базовых тарифах всех рассмотренных конкурентов;
- индикатор риска потери сделки на основе анализа активности — инструмент проактивного управления, которого нет ни у одного из конкурентов в бесплатном варианте;
- минимальный порог освоения — система проектируется специально для небольших команд без ИТ-специалиста в штате;
- современный тёмный интерфейс — сопоставим с ведущими коммерческими платформами.

1.4 Экономико-информационная сущность задачи

Описание существующего бизнес-процесса (AS-IS). Для описания существующей технологии управления клиентскими взаимодействиями в туапсинском филиале ООО ИК «СИБИНТЕК» применяется методология структурного функционального моделирования IDEF0 (Integrated DEFinition for FunctionModeling). IDEF0 является государственным стандартом США (FIPS 183) и широко используется в отечественной практике проектирования информационных систем. В нотации IDEF0 каждая функция описывается четырьмя группами элементов: входы (Input), управляющие воздействия (Control), механизмы (Mechanism), выходы (Output) — так называемая ICOM-

модель.

Контекстная диаграмма (уровень A0) описывает процесс «Управлять клиентскими взаимодействиями» в целом. На входе — клиентское обращение (по e-mail или телефону) и данные о клиенте. Управляющими воздействиями выступают регламент работы с клиентами и корпоративные стандарты СИБИНТЕК. Механизмы — сотрудники филиала и доступные им инструменты (электронная почта, Microsoft Excel, мессенджеры). На выходе — ответ клиенту, выполненная задача и ежемесячная отчётность. Контекстная диаграмма приведена в приложении А (рисунок А.1).

Диаграмма декомпозиции (уровень A1) детализирует контекстный процесс на три подфункции. Приведена в приложении А (рисунок А.2).

Функция A1 «Приём и первичная регистрация обращения» включает: приём заявки по электронной почте или телефону; идентификацию клиента (поиск предыдущей переписки в почте); первичную оценку срочности и типа обращения; фиксацию в личной таблице Microsoft Excel сотрудника. На выходе — зарегистрированная задача в личной таблице.

Функция A2 «Обработка и исполнение задачи» охватывает: назначение исполнителя (устно или по электронной почте); выполнение работ; согласование результата с клиентом; при необходимости — передачу задачи другому специалисту с пересылкой цепочки писем. На выходе — выполненная задача и уведомление клиента.

Функция A3 «Закрытие и формирование отчётности» включает: обновление статуса в личной таблице Excel; ежемесячную выгрузку данных каждым сотрудником; ручную консолидацию в сводный файл; отправку отчётности в головной офис. На выходе — ежемесячный отчёт.

Анализ информационных потоков выявил следующие проблемы. Клиент формирует обращение произвольной формы, без стандартизированного шаблона. Данные о клиенте распределены между почтовым клиентом (история переписки), личной таблицей Excel сотрудника (текущие задачи) и телефонной

книгой (контакты). Единой точки хранения клиентской информации не существует.

Ежемесячная отчётность формируется вручную: каждый сотрудник выгружает данные из своей таблицы Excel, после чего один из сотрудников сводит всё в единый документ. По оценкам сотрудников, этот процесс занимает от трёх до четырёх рабочих часов и нередко сопровождается расхождениями в данных из-за несовместимости форматов индивидуальных таблиц.

Выявленные недостатки существующей технологии:

- потеря обращений — при большом потоке входящих писем часть обращений рискует остаться без ответа; без единого реестра заявок контролировать это невозможно; по оценке сотрудников, не менее двух обращений в месяц «зависают» без ответа более трёх рабочих дней;

- отсутствие единой истории взаимодействий — при повторном обращении клиента сотруднику необходимо вручную искать предыдущую переписку в почте; среднее время поиска составляет 10–20 минут; при отсутствии исполнителя (отпуск, болезнь) информация о текущих задачах становится недоступной для коллег;

- сложность передачи задач — делегирование задачи другому сотруднику требует пересылки цепочки писем или устного объяснения контекста; при этом часть контекста неизбежно теряется;

- отсутствие оперативного контроля — руководитель не может быстро оценить загруженность сотрудников и статус открытых задач без индивидуального опроса каждого исполнителя;

- трудоёмкость подготовки отчётности — ежемесячная консолидация данных из личных таблиц занимает 3–4 рабочих часа и сопровождается ошибками; данные за прошлые периоды практически недоступны для анализа.

Количественная оценка потерь. В ходе преддипломной практики были проведены беседы с сотрудниками отдела технической поддержки и отдела по работе с клиентами. Полученные оценки:

- среднее время поиска истории взаимодействий с конкретным клиентом

в почтовой переписке — 10–20 минут на запрос;

- время подготовки ежемесячной отчётности — 3–4 рабочих часа (1 раз в месяц);

- число обращений, не получивших ответа в течение трёх рабочих дней, — не менее двух в месяц;

- время, затрачиваемое на повторное «вхождение в контекст» при возвращении к задаче после перерыва, — 5–10 минут на задачу.

Суммарно неэффективные операции, поддающиеся автоматизации, занимают порядка 8–10 рабочих часов в месяц у каждого сотрудника, непосредственно работающего с клиентами. При команде из 4 таких сотрудников совокупные потери составляют 32–40 рабочих часов в месяц.

Описание предлагаемого бизнес-процесса (ТО-BE). Предлагаемая технология описана в нотации BPMN 2.0 (Business Process Model and Notation) — международном стандарте описания бизнес-процессов ISO 19510:2013. Нотация BPMN ориентирована на описание потоков управления между участниками (пулами и дорожками) и обеспечивает наглядное отображение взаимодействия сотрудников и системы.

Диаграмма ТО-BE содержит три дорожки (swimlanes): «Клиент», «Сотрудник» и «Система Мини-CRM». Диаграмма приведена в приложении Б (рисунок Б.1).

В предлагаемой технологии система «Мини-CRM» берёт на себя все рутинные операции по хранению информации, уведомлению и формированию отчётности: клиент направляет обращение любым удобным способом; сотрудник фиксирует его в системе через стандартизированную форму; система автоматически создаёт запись в базе данных, назначает задачу ответственному и отображает её в интерфейсе; сотрудник выполняет задачу и закрывает её в системе; система фиксирует запись в истории взаимодействий и обновляет статус на дашборде.

Сравнение существующей и предлагаемой технологий по ключевым характеристикам приведено в таблице 1.2.

Таблица 1.2 - Сравнение технологий AS-IS и TO-BE

Характеристика	AS-IS (существующая)	TO-BE (предлагаемая)
Хранение данных о клиентах	Разрозненно (e-mail, Excel)	Централизованно в реляционной БД
Регистрация обращений	Вручную, без единого формата	Стандартизированная форма в системе
История взаимодействий	В переписке или отсутствует	Автоматически, мгновенный поиск
Передача задач между сотрудниками	По e-mail или устно	Переназначение ответственного в один клик
Контроль статусов задач	Только у исполнителя	Канбан-доска: виден всей команде
Оперативный контроль руководителя	Требует опроса сотрудников	Дашборд с KPI в реальном времени
Подготовка ежемесячной отчётности	Ручная консолидация, 3–4 ч	Автоматически, экспорт по кнопке
Поиск истории по клиенту	10–20 минут (поиск в почте)	Мгновенно в карточке клиента
Потеря обращений	~2 в месяц	Исключена системой учёта
Прогнозирование рисков по сделкам	Отсутствует	Автоматический расчёт risk_score

Как видно из таблицы 1.2, предлагаемая технология системно устраняет все выявленные в ходе анализа недостатки существующего процесса. Особо следует отметить, что ряд функций — индикатор риска потери сделки, генератор коммерческих предложений — не просто автоматизируют существующие операции, но создают качественно новые управленческие возможности, которых не было в системе до внедрения.

2 Проектная часть

2.1 Информационное обеспечение задачи

Информационное обеспечение системы включает структуру базы данных, входные и выходные информационные потоки, а также описание сущностей, их атрибутов и связей.

Проектирование базы данных. Для хранения данных разрабатываемой системы спроектирована реляционная база данных на основе СУБД PostgreSQL 15. Реляционная модель была выбрана в противовес документно-ориентированным (MongoDB) и графовым (Neo4j) СУБД ввиду чётко структурированной предметной области с устойчивыми связями между сущностями. Структура базы данных описывается ER-диаграммой (Entity-RelationshipDiagram, диаграмма «сущность-связь») в нотации Чена. ER-диаграмма приведена в приложении В (рисунок В.1).

База данных приведена к третьей нормальной форме (3НФ). Первая нормальная форма (1НФ) обеспечена атомарностью всех атрибутов: каждое поле хранит единственное неделимое значение. Вторая нормальная форма (2НФ) обеспечена устранением частичных зависимостей: все неключевые атрибуты зависят от первичного ключа целиком, а не от его части. Третья нормальная форма (3НФ) обеспечена устранением транзитивных зависимостей: ни один неключевой атрибут не зависит от другого неключевого атрибута. Приведение к 3НФ исключает аномалии обновления, вставки и удаления данных.

В базе данных выделены шесть сущностей.

Сущность `users` (пользователи системы). Хранит учётные записи сотрудников. Атрибуты: `id` (INTEGER, первичный ключ, автоинкремент); `full_name` (VARCHAR(255), полное имя, NOT NULL); `email` (VARCHAR(255), уникальный адрес электронной почты, UNIQUE NOT NULL); `password_hash` (VARCHAR(255), хэш пароля bcrypt, NOT NULL); `role` (ENUM('admin', 'manager', 'viewer'), роль пользователя, DEFAULT 'manager'); `is_active`

(BOOLEAN, признак активности учётной записи, DEFAULT TRUE); last_login (TIMESTAMP, дата последнего входа); created_at (TIMESTAMP, дата создания записи, DEFAULT NOW()); updated_at (TIMESTAMP, дата изменения, DEFAULT NOW()).

Сущность clients (клиенты). Содержит карточки клиентских организаций и контактных лиц. Атрибуты: id (INTEGER, PK); full_name (VARCHAR(255), наименование клиента, NOT NULL); phone (VARCHAR(30), номер телефона); email (VARCHAR(255), адрес электронной почты клиента); address (TEXT, почтовый адрес); source (ENUM('yandex', 'vk', 'referral', 'repeat', 'cold_call', 'site', 'other'), источник обращения); status (ENUM('new', 'active', 'hot', 'cold', 'lost'), статус клиента, DEFAULT 'new'); notes (TEXT, произвольные примечания); created_by (INTEGER, FK → users.id, кто создал запись); created_at, updated_at.

Сущность deal_stages (стадии воронки продаж). Описывает этапы, которые проходит сделка в процессе работы. Атрибуты: id (INTEGER, PK); name (VARCHAR(100), наименование стадии, NOT NULL); order (INTEGER, порядок отображения в воронке); color (VARCHAR(20), цвет стадии в интерфейсе канбан-доски); is_final (BOOLEAN, признак завершающей стадии); is_won (BOOLEAN, признак успешного закрытия).

Сущность deals (сделки). Является центральной в модели. Атрибуты: id (INTEGER, PK); title (VARCHAR(255), наименование сделки, NOT NULL); amount (DECIMAL(14,2), сумма сделки, DEFAULT 0); client_id (INTEGER, FK → clients.id, NOT NULL); stage_id (INTEGER, FK → deal_stages.id, NOT NULL); responsible_user_id (INTEGER, FK → users.id, ответственный менеджер); closed_at (TIMESTAMP, дата закрытия сделки); loss_reason (VARCHAR(255), причина отказа при проигрыше); risk_score (INTEGER, расчётный показатель риска потери, DEFAULT 0); last_activity_at (TIMESTAMP, дата последней активности, DEFAULT NOW()); created_at, updated_at.

Сущность tasks (задачи). Хранит задачи и поручения, привязанные к сделкам и клиентам. Атрибуты: id (INTEGER, PK); title (VARCHAR(255), NOT NULL); description (TEXT, расширенное описание); deal_id (INTEGER, FK →

deals.id, опциональная привязка к сделке); client_id (INTEGER, FK → clients.id, опциональная привязка к клиенту); assigned_to (INTEGER, FK → users.id, ответственный исполнитель); created_by (INTEGER, FK → users.id, автор задачи); priority (ENUM('low', 'medium', 'high'), приоритет, DEFAULT 'medium'); status (ENUM('open', 'done', 'cancelled'), статус задачи, DEFAULT 'open'); due_date (TIMESTAMP, срок выполнения); completed_at (TIMESTAMP, фактическая дата выполнения); created_at, updated_at.

Сущность audit_logs (журнал аудита). Фиксирует все значимые действия пользователей системы. Атрибуты: id (INTEGER, PK); user_id (INTEGER, FK → users.id); action (VARCHAR(50), тип действия: login, create, update, delete); entity_type (VARCHAR(50), тип объекта: client, deal, task); entity_id (INTEGER, идентификатор изменённого объекта); changes (JSONB, JSON-объект с изменениями — поля до и после); ip_address (VARCHAR(45), IP-адрес пользователя); created_at (TIMESTAMP, NOT NULL).

Связи между сущностями. Один клиент может иметь множество сделок (1:N, clients → deals). Одна стадия может содержать множество сделок (1:N, deal_stages → deals). Один пользователь может быть ответственным за множество сделок (1:N, users → deals). Одна сделка может иметь множество задач (1:N, deals → tasks). Один клиент может иметь множество задач (1:N, clients → tasks). Один пользователь может быть исполнителем множества задач (1:N, users → tasks). Один пользователь порождает множество записей в журнале аудита (1:N, users → audit_logs).

Индексирование базы данных. Для оптимизации производительности запросов созданы индексы: составной индекс (client_id, stage_id) для запросов канбан-доски; индекс (assigned_to) для фильтрации задач по ответственному; индекс (status) для фильтрации задач и сделок; индекс (due_date) для сортировки задач по сроку; индекс (created_at DESC) для сортировки в ленте активности.

Входные данные системы: данные клиента (ФИО / наименование, телефон, e-mail, адрес, источник заявки, примечания); данные сделки

(наименование, описание работ, сумма, стадия, ответственный менеджер); данные задачи (заголовок, описание, приоритет, срок выполнения, ответственный исполнитель); учётные данные пользователя (e-mail, пароль) при авторизации.

Выходные данные системы: сводный дашборд с четырьмя KPI-карточками (выручка за месяц, новые клиенты, конверсия воронки, активные сделки); канбан-доска сделок с автоматическими индикаторами риска; таблица клиентов с историей взаимодействий; список задач с фильтрацией по статусу и приоритету; аналитические отчёты по источникам заявок и эффективности менеджеров; персонализированный текст коммерческого предложения.

Функциональные требования к системе (сформулированы по результатам анализа бизнес-процессов):

- ведение базы клиентов — создание, просмотр, редактирование карточек; поиск по ФИО, телефону, e-mail; фильтрация по статусу и источнику обращения;
- управление сделками — создание, редактирование, архивирование; привязка к клиенту; назначение ответственного менеджера;
- канбан-доска с настраиваемыми стадиями воронки продаж;
- управление задачами — создание, назначение ответственного, установка дедлайна, смена статуса;
- автоматический расчёт и отображение риска потери сделки на основе активности;
- автоматическая фиксация истории всех действий пользователей (журнал аудита);
- дашборд с ключевыми показателями и лентой последних событий;
- аналитика по источникам заявок, конверсии воронки, выручке по менеджерам;
- шаблонный генератор коммерческих предложений с тремя стилями;
- ролевая модель: администратор — полный доступ; менеджер — работа со своими клиентами; наблюдатель — только просмотр;

- JWT-аутентификация с автоматическим обновлением токена.

Нефункциональные требования:

- работа через браузер без установки дополнительного ПО на рабочих станциях;
- поддержка одновременной работы до 20 пользователей;
- время отклика интерфейса не более 2 секунд при типовых операциях;
- хранение персональных данных в СУБД PostgreSQL на сервере в РФ (соответствие Ф3-152);
- хэширование паролей алгоритмом bcrypt (saltrounds = 12);
- передача данных по HTTPS в производственной среде;
- доступность системы не менее 99% рабочего времени.

Use-Case диаграмма, описывающая сценарии взаимодействия двух актёров (менеджер и администратор) с системой, приведена в приложении Г (рисунок Г.1). В системе определены два типа актёров. Менеджер выполняет основные операционные функции: авторизацию, управление карточками клиентов, создание и обработку сделок и задач, просмотр дашборда и аналитики, генерацию коммерческих предложений. Администратор обладает всеми правами менеджера и дополнительно управляет учётными записями пользователей, настраивает стадии воронки и имеет доступ к журналу аудита.

2.2 Технологическое обеспечение

2.2.1 Обоснование проектных решений по технологическому обеспечению

Общая архитектура. Приложение построено по трёхуровневой клиент-серверной архитектуре. Первый уровень — уровень представления (Presentation Layer) — реализован как SPA (Single Page Application) на базе Vue.js 3. Второй уровень — уровень бизнес-логики (Business Logic Layer) — представлен REST API сервером на Node.js/Express.js. Третий уровень — уровень данных (Data Layer) — реализован в виде реляционной СУБД PostgreSQL 15.

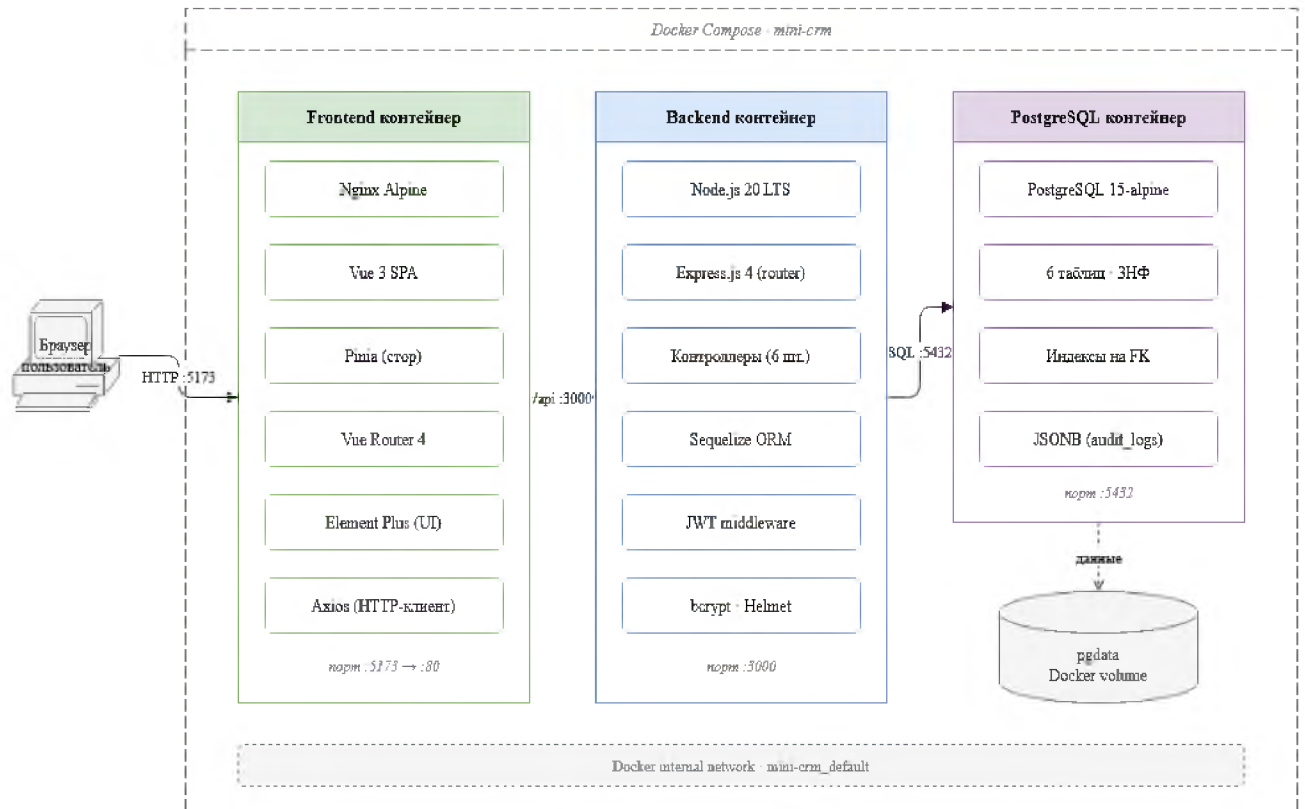


Рисунок 2.1 - Компонентная диаграмма системы «Мини-CRM»

Трёхуровневая архитектура обеспечивает чёткое разделение ответственности между компонентами системы: уровень представления не содержит бизнес-логики; уровень бизнес-логики не зависит от способа хранения данных; уровень данных не зависит от интерфейса. Это позволяет независимо разрабатывать, тестировать и масштабировать каждый компонент [11].

Взаимодействие между уровнями организовано через REST API (Representational State Transfer). Клиент отправляет HTTP-запросы методами GET (получение данных), POST (создание), PUT (полное обновление), PATCH (частичное обновление) и DELETE (удаление). Сервер обрабатывает запросы через Sequelize-модели и возвращает ответы в формате JSON. Каждый защищённый маршрут требует наличия валидного JWT-токена в заголовке Authorization.

Контейнеризация. Для обеспечения воспроизводимого и изолированного запуска всех сервисов используется DockerCompose. Контейнеризация

позволяет развернуть всё приложение одной командой `docker-compose up -d` на любом сервере с установленным Docker Engine. Конфигурация `docker-compose.yml` описывает три сервиса: `postgres` (СУБД), `backend` (Node.js API) и `frontend` (Vue SPA + Nginx). Это исключает проблему «работает на моём компьютере» и гарантирует одинаковое поведение системы в любой среде [12].

Обоснование выбора Vue.js 3. Для разработки фронтенда выбран Vue.js 3 — прогрессивный JavaScript-фреймворк с открытым исходным кодом. Версия 3 (выпущена в 2020 году) использует Composition API, обеспечивающий лучшую переиспользуемость логики и интеграцию с TypeScript. Vue.js реализует компонентный подход: каждый элемент интерфейса (карточка клиента, форма сделки, таблица задач, канбан-колонка) представляет собой изолированный компонент со своим шаблоном, логикой и стилями.

При выборе фреймворка рассматривались три основных претендента: Vue.js 3, React 18 и Angular 17. Сравнение приведено в таблице 2.1.

Таблица 2.1 - Сравнение JavaScript-фреймворков для фронтенда

Критерий	Vue.js 3	React 18	Angular 17
Порог вхождения	Низкий	Средний	Высокий
Встроенный роутинг	VueRouter (офиц.)	Сторонний	Встроен
Управление состоянием	Pinia (офиц.)	Redux/Zustand	NgRx
Размер bundle (минимальный)	~33 КБ	~42 КБ	~130 КБ
Документация на русском	Хорошая	Частичная	Частичная
Подходит для CRM-масштаба	Да	Да	Избыточно

Vue.js 3 предпочтён React по критериям низкого порога вхождения, наличия официального управления состоянием (Pinia) и достаточности для данного масштаба приложения. Angular избыточен: его *opinionated*-архитектура добавляет сложность, неоправданную для CRM-системы из 6 страниц.

Технологический стек. Итоговый выбор технологий для каждого уровня системы обоснован следующим образом. На уровне представления: Vue.js 3.4 — компонентный фреймворк; Pinia 2.1 — управление глобальным состоянием;

VueRouter 4.2 — клиентская маршрутизация; Element Plus 2.4 — UI-библиотека компонентов на русском языке; Axios 1.6 — HTTP-клиент; Vite 5.0 — инструмент сборки. На уровне бизнес-логики: Node.js 20 LTS — серверная платформа; Express.js 4.18 — веб-фреймворк; Sequelize 6.35 — ORM для PostgreSQL; jsonwebtoken 9.0 — JWT-аутентификация; bcrypt 5.1 — хэширование паролей. На уровне данных: PostgreSQL 15 — реляционная СУБД.

Полный технологический стек приведён в таблице 2.2.

Таблица 2.2 - Технологический стек разработки

Компонент	Технология	Версия	Назначение
Фронтенд-фреймворк	Vue.js	3.4	SPA-приложение, компонентный подход
Управление состоянием	Pinia	2.1	Глобальный стор (auth, данные)
Маршрутизация	VueRouter	4.2	Навигация без перезагрузки страниц
UI-компоненты	Element Plus	2.4	Готовые RU-компоненты интерфейса
HTTP-клиент	Axios	1.6	Запросы к REST API, интерцепторы
Сборщик	Vite	5.0	Быстрая сборка, HMR в разработке
Бэкенд-платформа	Node.js	20 LTS	Серверная среда выполнения JS
Бэкенд-фреймворк	Express.js	4.18	HTTP-сервер, роутинг, middleware
ORM	Sequelize	6.35	Абстракция над PostgreSQL
СУБД	PostgreSQL	15	Реляционное хранилище данных
Аутентификация	jsonwebtoken	9.0	Генерация и верификация JWT
Хэширование паролей	bcrypt	5.1	Безопасное хранение паролей
Контейнеризация	Docker + Compose	24+	Изолированный запуск сервисов
Веб-сервер	Nginx	Alpine	Раздача SPA, проксирование API

2.2.2 Технологические операции

Технологический процесс функционирования системы включает несколько основных операций, каждая из которых реализована в соответствующем модуле системы.

Операция 1: Аутентификация пользователя. Пользователь открывает страницу авторизации и вводит e-mail и пароль. Клиент отправляет POST-запрос на `/api/auth/login`. Сервер извлекает запись пользователя из таблицы `users` по e-mail. Функция `bcrypt.compare()` сравнивает введенный пароль с хэшем из базы данных. При совпадении сервер генерирует две JSON Web Token: access-токен с полезной нагрузкой `{id, role}` и сроком действия 15 минут; refresh-токен со сроком действия 7 дней. Оба токена возвращаются клиенту. Access-токен сохраняется в `localStorage` и добавляется к заголовку `Authorization: Bearer<token>` каждого последующего запроса. При истечении access-токена Axios-интерцептор автоматически отправляет refresh-токен на `/api/auth/refresh` и получает новый access-токен без прерывания работы пользователя [13].

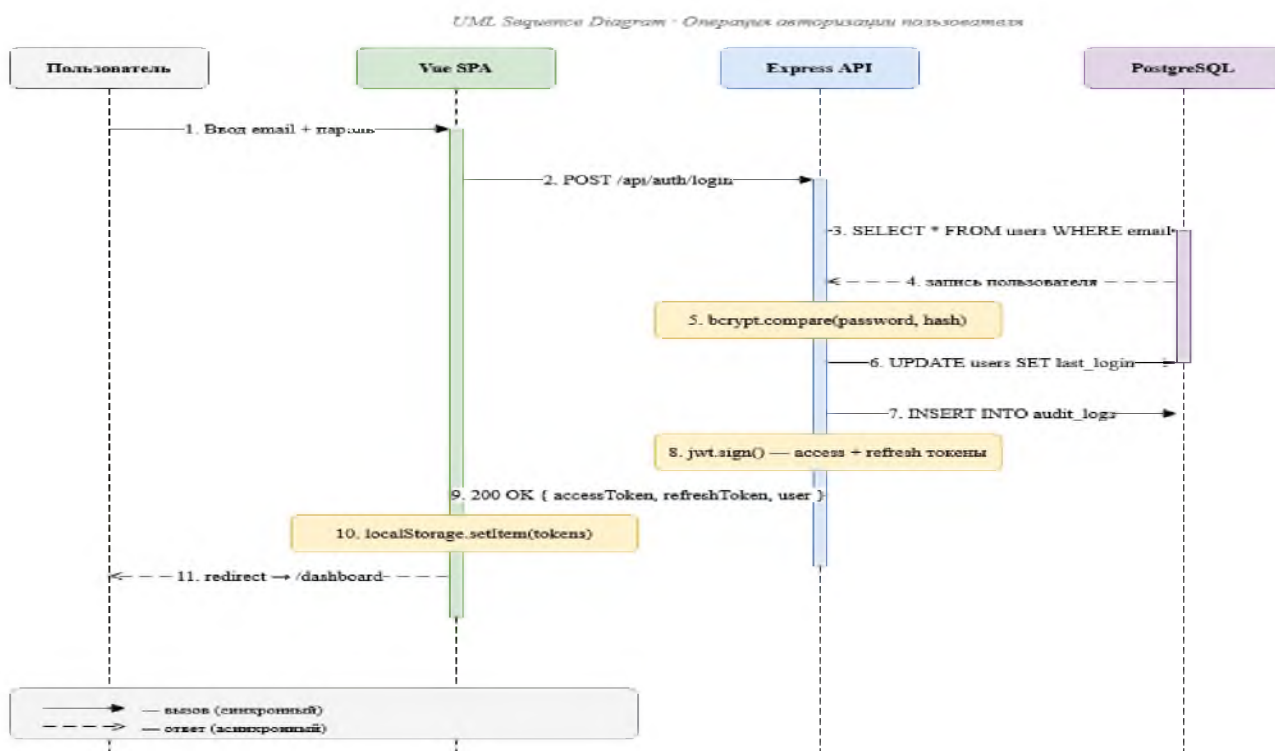


Рисунок 2.2 - Диаграмма последовательности процесса авторизации пользователя

Операция 2: Создание клиента и сделки. Сотрудник заполняет форму в диалоговом окне. Клиент отправляет POST-запрос на `/api/clients` с данными клиента.

Сервер валидирует обязательные поля (`full_name`), создаёт запись в таблице `clients` и возвращает созданный объект с присвоенным `id`.

Затем клиент отправляет POST-запрос на `/api/deals` с `client_id`, `title`, `amount` и `stage_id` первой стадии воронки. Сервер создаёт запись в таблице `deals`, устанавливает `last_activity_at = NOW()` и фиксирует событие `create` в журнале аудита. Канбан-доска обновляется без перезагрузки страницы.

Операция 3: Перемещение сделки по воронке. Сотрудник кликает по кнопке на карточке сделки или открывает её для редактирования. Клиент отправляет PATCH-запрос на `/api/deals/:id/stage` с новым `stage_id`.

Контроллер `dealController.js` обновляет поле `stage_id`, сбрасывает `last_activity_at = NOW()` (это автоматически обнуляет `risk_score`) и, если стадия является финальной (`is_final = true`), устанавливает `closed_at = NOW()`. Событие фиксируется в журнале аудита с сохранением предыдущего и нового значения `stage_id`.

Операция 4: Расчёт риска потери сделки (`risk_score`). При каждом запросе к `/api/deals/kanban` сервер вычисляет `risk_score` для каждой открытой сделки по следующему алгоритму.

Компонент «дни без активности»: если прошло более 7 дней с момента `last_activity_at` — прибавляется 40 баллов; 4–7 дней — 20 баллов; 2–4 дней — 10 баллов. Компонент «сумма сделки»: при сумме свыше 300 000 рублей — дополнительно 15 баллов (крупные сделки требуют большего внимания); 150 000–300 000 рублей — 8 баллов.

Компонент «время в воронке»: если сделка создана более 14 дней назад — 20 баллов; 7–14 дней — 10 баллов.

Итоговое значение ограничивается 100 баллами. При `risk_score` от 60 и выше сделка отображается с красным индикатором, 30–59 — жёлтым, менее 30 — зелёным.

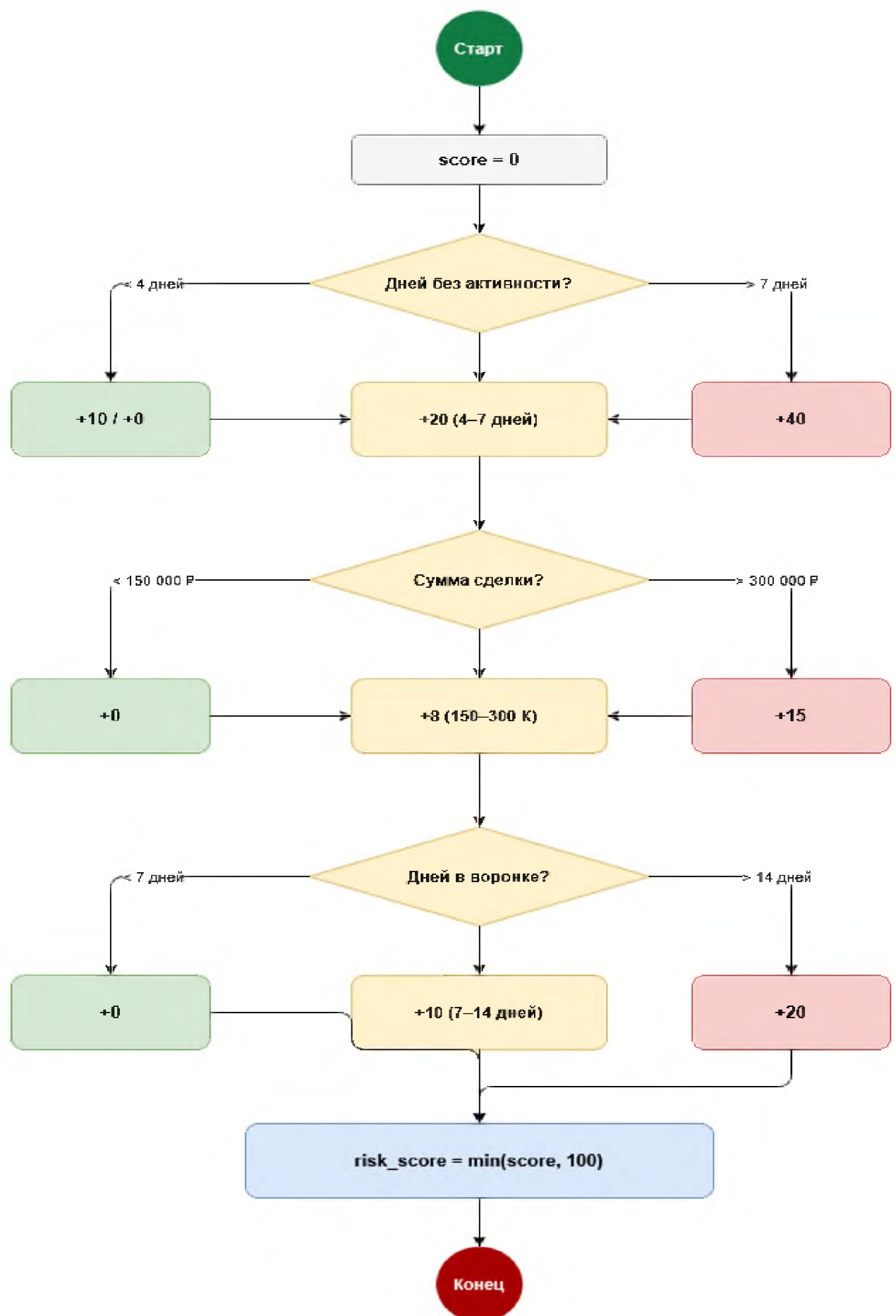


Рисунок 2.3 - Диаграмма деятельности алгоритма расчёта показателя risk_score

Операция 5: Генерация коммерческого предложения. Сотрудник выбирает клиента, вид работ, вводит сумму по смете и выбирает стиль предложения.

Клиент отправляет POST-запрос на `/api/proposal/generate`. Контроллер `proposalController.js` вызывает функцию `buildProposal()`, которая: извлекает имя клиента из строки `full_name`; подбирает текстовое описание вида работ из словаря `workDescriptions`; формирует блок преимуществ компании; формирует строку с суммой (или текст о бесплатном выезде замерщика, если сумма не указана); выбирает приветствие и закрытие в зависимости от выбранного стиля.

Итоговый текст КП собирается из блоков и возвращается клиенту. Генерация происходит мгновенно без обращения к внешним сервисам.

Операция 6: Формирование дашборда. При переходе на страницу «Дашборд» Vue-компонент `DashboardPage.vue` параллельно запрашивает три эндпоинта: `/api/analytics/dashboard` (выручка, конверсия, количество клиентов, сделок), `/api/analytics/funnel` (распределение сделок по стадиям воронки), `/api/proposal/risky-deals` (сделки с `risk_score` ≥ 30).

Запросы выполняются через `Promise.all()` для минимизации времени ожидания. Полученные данные отображаются в карточках KPI, горизонтальной диаграмме воронки и блоке предупреждений.

2.3 Техническое обеспечение

2.3.1 Обоснование проектных решений по техническому обеспечению

Разрабатываемое веб-приложение работает в архитектуре «тонкий клиент — толстый сервер». На стороне клиента запускается только браузер; вся бизнес-логика, хранение данных и вычисления выполняются на сервере. Это обеспечивает централизацию данных и снижает требования к рабочим станциям пользователей.

Выбор операционной системы сервера. Для серверной части рекомендуется ОС Ubuntu Server 22.04 LTS (Long Term Support). Ubuntu

является наиболее распространённой серверной ОС Linux среди отечественных VPS-провайдеров, что гарантирует доступность образов и техническую поддержку.

Срок поддержки версии 22.04 — до апреля 2027 года (стандартная поддержка) и до апреля 2032 года (расширенная).

Обоснование использования VPS-хостинга. Аренда VPS (Virtual Private Server) является оптимальным выбором для production-развёртывания разрабатываемой системы. VPS обеспечивает: выделенные вычислительные ресурсы (в отличие от shared-хостинга); полный root-доступ для установки Docker и настройки ОС; фиксированный публичный IP-адрес; возможность настройки домена и SSL-сертификата; размещение в российских дата-центрах (соответствие Ф3-152). Стоимость аренды подходящего VPS на отечественных платформах (Selectel, TimeWeb, Reg.ru, 1cloud) составляет от 500 до 800 рублей в месяц.

Минимальные технические требования к серверу: процессор — не менее 2 ядер с тактовой частотой 2,0 ГГц и выше; оперативная память — не менее 2 ГБ RAM (рекомендуется 4 ГБ); дисковое пространство — не менее 20 ГБ SSD; операционная система — Ubuntu Server 22.04 LTS или Debian 12; сеть — подключение не менее 100 Мбит/с, статический IP-адрес.

Требования к рабочим станциям пользователей минимальны: любой современный браузер (Google Chrome версии 90 и выше, Mozilla Firefox версии 88 и выше, Microsoft Edge версии 90 и выше, Safari версии 14 и выше); разрешение экрана не менее 1280 × 720 пикселей; стабильный доступ к сети Интернет или локальной сети, в которой развёрнут сервер. Дополнительное программное обеспечение на рабочие станции не устанавливается.

Для демонстрационного запуска — в том числе для представления на защите выпускной квалификационной работы — достаточно любого ПК или ноутбука с установленным Docker Desktop.

В этом случае приложение доступно по адресу <http://localhost:5173> без подключения к Интернету.

2.3.2 Комплекс технических средств

Комплекс технических средств, необходимых для функционирования разработанной системы, приведён в таблице 2.3.

Таблица 2.3 - Комплекс технических средств

Наименование	Характеристики	Количество	Назначение
VPS-сервер (production)	2 ядра / 2 ГБ RAM / 20 ГБ SSD / Ubuntu 22.04	1	Хостинг всех сервисов
Docker Engine	Версия 24+, устанавливается на сервер	1	Контейнеризация
Nginx (контейнер)	Alpine образ, проксирование	1	Раздача SPA, reverse-proxy
Рабочие станции	ПК / ноутбук, CPU 1+ ГГц, 2 ГБ RAM	По числу пользователей	Клиентский доступ
Браузер	Chrome 90+ / Firefox 88+ / Edge 90+	На каждой рабочей станции	Интерфейс системы
Сеть передачи данных	Интернет или корп. LAN, 10 Мбит/с	—	Связь клиент-сервер
ПК для демонстрации (опц.)	Ноутбук с Docker Desktop	1	Демо без сервера

Для обеспечения безопасности передаваемых данных в производственной среде рекомендуется установить SSL/TLS-сертификат. Бесплатный сертификат доступен через сервис Let'sEncrypt. Nginx настраивается как reverse-proxy с перенаправлением HTTP-трафика на HTTPS. Это обеспечивает шифрование данных между браузером пользователя и сервером, что является обязательным требованием при передаче персональных данных.

2.4 Программное обеспечение задачи

2.4.1 Обоснование проектных решений по программному обеспечению

Фронтенд. Для построения пользовательского интерфейса выбран Vue.js

3 с использованием Composition API. Composition API (введён в Vue 3) позволяет организовывать логику компонента по функциональным блокам, а не по типам опций (data, methods, computed), что улучшает читаемость кода и облегчает переиспользование логики через composables [14].

Для управления глобальным состоянием приложения используется Pinia 2.1 — официальный стор для Vue 3, рекомендованный командой разработчиков фреймворка взамен Vuex. Pinia отличается значительно более простым и интуитивным API: стор определяется функцией defineStore() с явными state, getters и actions, что устраняет необходимость в мутациях и намеренно сложных паттернах Vuex [17].

Маршрутизация реализована через VueRouter 4.2. Система использует режим HTML5 History API (createWebHistory), что обеспечивает «красивые» URL без символа # и корректную работу back-button браузера. Защищённые маршруты охраняются глобальным navigationguard: неавторизованный пользователь автоматически перенаправляется на страницу входа.

В качестве UI-библиотеки выбран Element Plus 2.4 — богатый набор готовых Vue 3-компонентов с полной русскоязычной локализацией. Element Plus включает: таблицы с сортировкой и пагинацией; формы с валидацией; диалоговые окна; выпадающие списки; компоненты выбора дат; уведомления и всплывающие сообщения. Использование готовой UI-библиотеки исключает необходимость реализовывать базовые компоненты с нуля и обеспечивает визуальную согласованность интерфейса [29].

Бэкенд. Для серверной части выбрана платформа Node.js 20 LTS с фреймворком Express.js 4. Node.js основан на движке V8 от Google Chrome и выполняет JavaScript-код вне браузера. Асинхронная неблокирующая модель ввода-вывода Node.js хорошо подходит для REST API-серверов, обрабатывающих множество параллельных HTTP-запросов с относительно небольшой вычислительной нагрузкой каждый [15].

Express.js является наиболее популярным Node.js-фреймворком: по данным репозитория npm, он загружается более 30 миллионов раз в неделю.

Фреймворк предоставляет минималистичный, но достаточный набор инструментов: роутинг HTTP-запросов; промежуточные обработчики (middleware) для авторизации, логирования, CORS; обработка ошибок. Его минимализм является достоинством: разработчик получает полный контроль над структурой приложения.

В качестве альтернатив рассматривались: Python/Django — зрелый фреймворк с богатым встроенным функционалом (ORM, административный интерфейс), однако использование другого языка усложняет разработку в рамках одного проекта; Python/FastAPI — современный асинхронный фреймворк с автогенерацией документации API, но требует более высоких компетенций. Выбор Node.js/Express.js обусловлен единством языка (JavaScript) для обеих частей приложения.

СУБД. Выбрана PostgreSQL 15 — свободная объектно-реляционная СУБД с открытым исходным кодом, разрабатываемая с 1986 года. PostgreSQL обеспечивает: строгое следование стандарту SQL:2011; полную поддержку транзакций по модели ACID; богатый набор типов данных, включая JSONB (бинарный JSON) для хранения полуструктурированных данных; встроенную поддержку полнотекстового поиска. Postgres Pro (российский дистрибутив PostgreSQL с дополнительными возможностями) включён в Реестр отечественного ПО Минцифры России под номером 6396, что актуально при внедрении в государственных структурах [16].

Для работы с базой данных используется ORM Sequelize 6.35 — Object-RelationalMapping библиотека для Node.js. Sequelize транслирует JavaScript-объекты (модели) в SQL-таблицы, автоматически генерирует запросы для CRUD-операций и поддерживает определение ассоциаций между моделями (hasMany, belongsTo,hasOne). Это позволяет работать с данными на уровне объектов, не написывая SQL вручную для типовых операций [18].

Аутентификация. Реализована на основе пары JSON Web Token. JWT является компактным самодостаточным форматом токена, содержащим подписанную полезную нагрузку (claims). Схема с двумя токенами (access +

refresh) обеспечивает оптимальный баланс между безопасностью и удобством: короткий срок жизни access-токена (15 минут) снижает риск при его компрометации, а refresh-токен (7 дней) позволяет пользователю оставаться авторизованным без повторного ввода пароля.

2.4.2 Архитектура программного обеспечения

Структура проекта. Приложение организовано в монорепозиторий следующей структуры: папка backend/ содержит серверную часть; папка frontend/ содержит клиентскую часть; файлы docker-compose.yml, README.md и .env.example находятся в корне проекта.

Структура бэкенда (backend/src/): config/ — конфигурация подключения к PostgreSQL и seed-скрипт с демонстрационными данными; models/ — Sequelize-модели шести сущностей и файл index.js с ассоциациями; controllers/ — шесть контроллеров (authController, clientController, dealController, taskController, proposalController, analyticsController); routes/ — файл index.js с определением всех 20 REST-маршрутов; middlewares/ — auth.js с JWT-middleware и requireRole().

Структура фронтенда (frontend/src/): assets/ — глобальные CSS-стили; components/layout/ — AppLayout.vue с боковой панелью и навигацией; components/pages/ — шесть страниц приложения (LoginPage, DashboardPage, KanbanPage, ClientsPage, TasksPage, AIPage, AnalyticsPage, ClientDetailPage); stores/ — Pinia-стор авторизации (auth.js); services/ — Axios-клиент с интерцепторами (api.js); router/ — конфигурация VueRouter с navigationguards.

REST API. Все эндпоинты сгруппированы по ресурсам и размещены под префиксом /api. Полный перечень эндпоинтов приведён в таблице 2.4.

Таблица 2.4 - REST API эндпоинты системы

Метод	URL	Описание	Auth
POST	/api/auth/login	Авторизация пользователя	Нет

Продолжение таблицы 2.4

POST	/api/auth/refresh	Обновление access-токена	Нет
GET	/api/auth/me	Данные текущего пользователя	JWT
GET	/api/clients	Список клиентов (поиск, фильтр, пагинация)	JWT
GET	/api/clients/:id	Карточка клиента со сделками	JWT
POST	/api/clients	Создание клиента	JWT
PUT	/api/clients/:id	Обновление данных клиента	JWT
DELETE	/api/clients/:id	Удаление клиента	Admin
GET	/api/deals	Список сделок с фильтрацией	JWT
GET	/api/deals/kanban	Канбан-доска по стадиям	JWT
GET	/api/deals/:id	Карточка сделки с задачами	JWT
POST	/api/deals	Создание сделки	JWT
PUT	/api/deals/:id	Обновление сделки	JWT
PATCH	/api/deals/:id/stage	Перемещение по воронке	JWT
GET	/api/tasks	Список задач с фильтрацией	JWT
POST	/api/tasks	Создание задачи	JWT
PUT	/api/tasks/:id	Обновление / закрытие задачи	JWT
POST	/api/proposal/generate	Генерация текста КП	JWT
GET	/api/proposal/risky-deals	Сделки с высоким риском	JWT
GET	/api/analytics/dashboard	Данные для дашборда	JWT
GET	/api/analytics/funnel	Данные воронки продаж	JWT
POST	/api/webhooks/tilda	Webhook от сайта Tilda	Нет

2.4.3 Описание программных модулей

Модуль аутентификации (authController.js). Реализует три метода: login(), refresh() и me(). Метод login() выполняет следующую последовательность: поиск пользователя в таблице users по e-mail с проверкой is_active = true; если пользователь не найден или неактивен — возврат ошибки 401; вызов bcrypt.compare(password, user.password_hash) для проверки пароля; при успехе — обновление поля last_login; генерация access-токена через jwt.sign({ id }, secret, { expiresIn: '15m' }); генерация refresh-токена с expiresIn '7d'; запись события login в таблицу audit_logs; возврат обоих токенов и данных пользователя. Метод refresh() верифицирует refresh-токен и возвращает новый access-токен. Метод me() возвращает данные текущего пользователя из req.user, установленного JWT-middleware.

Модуль управления клиентами (clientController.js). Реализует стандартные CRUD-операции. Метод getAll() поддерживает параметры запроса: search (поиск по имени, телефону, e-mail через SequelizeOp.iLike), status (фильтрация по статусу), source (фильтрация по источнику), page и limit (пагинация). Ответ включает поле total для отображения числа страниц в клиентском компоненте. Метод getOne() возвращает клиента вместе с его сделками через Sequelize eager loading (include: [{ model: Deal, as: 'deals', include: [{ model: DealStage, as: 'stage' }] }]).

Модуль управления сделками (dealController.js). Ключевой алгоритм — функция calcRiskScore(deal), описанная в разделе 2.2.2. Метод getKanban() возвращает все стадии воронки с вложенными сделками, отсортированными по дате создания. Метод moveStage() обрабатывает перемещение сделки: обновляет stage_id, устанавливает last_activity_at = newDate(), и при is_final = true устанавливает closed_at. Листинг алгоритма risk_score приведён в приложении Д (листинг Д.1).

Модуль генератора КП (proposalController.js). Функция buildProposal({ client_name, work_type, amount, tone, notes }) формирует текст коммерческого

предложения из нескольких блоков. Словарь `workDescriptions` содержит развёрнутые текстовые описания для 7 видов работ. Массив `advantages` содержит 5 фиксированных преимуществ компании. Строка `amountLine` формируется условно: если сумма указана — «Стоимость работ: X Р», иначе — «Стоимость будет определена после выезда замерщика». Блоки `greeting` и `closing` выбираются в зависимости от параметра `tone`. Итоговый текст собирается через `join('\n')` и возвращается как объект `{ proposal: string }`. Листинг модуля приведён в приложении Д (листинг Д.2).

Модуль аналитики (`analyticsController.js`). Метод `getDashboard()` выполняет четыре SQL-запроса параллельно через `Promise.all()`: `COUNT` сделок за текущий и предыдущий месяц (для расчёта процентного изменения); `SUM(amount)` закрытых сделок за месяц (выручка); `COUNT` новых клиентов за месяц; `COUNT` открытых сделок (активный портфель). Метод `getFunnel()` формирует данные воронки: для каждой стадии (в порядке `order`) подсчитывается число сделок и суммарный объём.

Компонент `AppLayout.vue` (макет). Реализует `sidebar` с навигацией и слот `<router-view />` для вставки текущей страницы. При монтировании компонента через `Axios` запрашиваются задачи с приближающимся дедлайном для отображения счётчика на пункте «Задачи». Компонент отслеживает активный маршрут через `router.currentRoute.value.path` и добавляет CSS-класс `router-link-active` для подсветки текущего раздела.

Компонент `KanbanPage.vue` (канбан-доска). Отображает сделки в колонках стадий воронки. Данные запрашиваются через `/api/deals/kanban` и приходят в виде массива объектов `[{id, name, color, deals: [...]}]`. Компонент отображает каждую колонку как отдельный `div` с вертикально прокручиваемым списком карточек. Кнопка «+ Добавить сделку» в нижней части каждой колонки открывает диалоговое окно `el-dialog` с формой, при сохранении которой данные канбан-доски обновляются.

`Stores/auth.js` (Pinia-стор авторизации). Стор содержит реактивный `ref user` и `computed isAuthenticated`. Метод `login()` отправляет запрос к API, сохраняет

токены в localStorage и устанавливает user.value. Метод logout() очищает localStorage и обнуляет user.value. Стор инициируется из localStorage при загрузке приложения, что обеспечивает сохранение сессии после перезагрузки браузера.

Services/api.js (Axios-инстанс). Создается через axios.create({ baseURL: '/api', timeout: 15000 }). Интерцептор запроса добавляет заголовок Authorization: Bearer<token> из localStorage. Интерцептор ответа перехватывает ошибку 401: при наличии refresh-токена автоматически запрашивает новый access-токен и повторяет исходный запрос. При отсутствии refresh-токена или ошибке его обновления — вызывает auth.logout() и перенаправляет на /login.

2.5 Руководство пользователя

2.5.1 Описание интерфейса

Общая концепция интерфейса. Разработанное веб-приложение «Мини-CRM» имеет тёмную цветовую схему (darkmode), оптимизированную для длительной работы: тёмный фон снижает нагрузку на зрение при многочасовом использовании системы. Основная цветовая палитра: фоновый цвет — #0f1117 (глубокий тёмно-синий), цвет панелей — #161b27, акцентный цвет — #4f8ef7 (синий), цвет успеха — #2ecc7a (зелёный), цвет предупреждения — #f59e0b (оранжевый), цвет ошибки — #ef4444 (красный). Данная цветовая схема соответствует современным стандартам профессиональных CRM-систем и SaaS-платформ.

Интерфейс построен по принципу SPA (Single Page Application): навигация между разделами осуществляется без перезагрузки страницы благодаря VueRouter. Это обеспечивает быстрый отклик и ощущение нативного приложения.

Страница авторизации. Является единственной публичной страницей системы. Содержит центрированную форму с логотипом приложения (буква «М» на синем фоне с наименованием организации), полем ввода e-mail, полем

ввода пароля с возможностью показа/скрытия символов, кнопкой «Войти в систему» с индикатором загрузки при отправке запроса. После успешной авторизации пользователь автоматически перенаправляется на дашборд. При неверных учётных данных отображается всплывающее уведомление с описанием ошибки. Скриншот страницы приведён в приложении Е (рисунок Е.1).

Боковая панель навигации (sidebar). Постоянно присутствует на всех страницах приложения, за исключением страницы авторизации. Состоит из трёх зон. Зона логотипа: иконка системы, наименование «МиниCRM», подзаголовок с наименованием организации. Зона навигации: содержит два раздела — «Главное» (Дашборд, Воронка сделок, Клиенты, Задачи) и «Инструменты» (Генератор КП, Аналитика). Активный пункт выделяется синей подсветкой. На пункте «Задачи» отображается красный счётчик задач с приближающимся дедлайном. Зона пользователя: содержит аватар с инициалами, имя и роль пользователя, кнопку выхода из системы.

Страница «Дашборд». Главная страница системы, доступная сразу после входа. Верхняя строка содержит название страницы и текущую дату. Далее расположены четыре карточки ключевых показателей (KPI): «Выручка (месяц)» с динамикой к предыдущему месяцу в процентах; «Новые клиенты» за текущий месяц; «Конверсия» воронки продаж в процентах; «Активных сделок» — число открытых сделок. Ниже размещены два блока: воронка продаж в виде горизонтальных столбиков с количеством сделок на каждом этапе; лента последних событий с цветовыми индикаторами типа события (зелёный — сделка закрыта, синий — новая заявка, оранжевый — предупреждение, красный — отказ). В нижней части дашборда выводится блок «Сделки в зоне риска» с перечнем сделок, где $risk_score \geq 30$. Скриншот дашборда приведён в приложении Е (рисунок Е.2).

Страница «Воронка сделок» реализует канбан-доску. Сделки распределены по колонкам, соответствующим стадиям воронки. Для каждой колонки отображается: цветная точка, наименование стадии, счётчик карточек.

Каждая карточка сделки отображает: наименование сделки; имя клиента с иконкой пользователя; сумму сделки в рублях (зелёный цвет); время без активности; цветовой бейдж риска (□ Высокий риск / △ Средний риск / ✓ Норма). В нижней части каждой колонки расположена кнопка «+ Добавить сделку», открывающая диалоговое окно с формой. Скриншоты страницы приведены в приложении Е (рисунок Е.3).

Страница «Клиенты». Содержит панель фильтрации (строка поиска с мгновенным поиском, фильтр по статусу, кнопка «+ Новый клиент») и таблицу с колонками: «Клиент» (имя, выделено жирным), «Телефон», «Источник» (с цветным индикатором-точкой), «Статус» (цветной бейдж), «Сделок», «Сумма», «Дата добавления». Таблица поддерживает пагинацию. Клик по строке таблицы открывает карточку клиента. Карточка клиента содержит крупный аватар с инициалами, полные данные клиента, статус-бейдж, список связанных сделок с их стадиями и суммами, блок быстрых действий (ссылки на генератор КП и воронку сделок). Скриншоты в приложении Е (рисунок Е.4).

Страница «Задачи». Верхняя строка содержит переключатель статусов (Открытые / Просроченные / Завершённые) и кнопку «+ Новая задача». Каждый элемент списка задач содержит: кружок статуса (при клике задача помечается выполненной с анимацией зачёркивания); заголовок задачи; привязанную сделку и клиента; бейдж приоритета (Высокий / Средний / Низкий); дедлайн (просроченные задачи выделяются красным цветом); кнопку удаления.

Страница «Генератор КП». Левая часть содержит форму генерации с полями: «Имя клиента» (текстовое поле); «Вид работ» (выпадающий список из 7 вариантов); «Сумма по смете» (числовое поле); «Стиль предложения» (выпадающий список: Профессиональный / Дружелюбный / Срочный с акцией); «Дополнительные пожелания» (текстовое поле, опционально); кнопку «Сформировать коммерческое предложение». После генерации под формой появляется блок результата с заголовком «Готовое КП», кнопками «Копировать текст» и «Очистить», и текстом предложения. Правая часть страницы содержит панель «Сделки без активности» со списком сделок, требующих внимания, и

советами по отправке КП. Скриншот в приложении Е (рисунок Е.5).

Страница «Аналитика». Содержит четыре метрические карточки с числовыми показателями за текущий месяц. Ниже размещены два горизонтальных столбчатых графика: «Заявки по источникам» (с цветовой кодировкой по типу источника) и «Выручка по менеджерам» (в тысячах рублей). В нижней части страницы расположен блок «Конверсия воронки» с двумя строками сравнения: конверсия до внедрения CRM (65,5%) и прогноз после внедрения (77%), отображаемыми горизонтальными полосами. Под ними выводится расчёт экономического эффекта в денежном выражении.

2.5.2 Порядок работы с системой

Установка и первоначальная настройка. Перед запуском системы необходимо убедиться, что на сервере установлен Docker Engine версии 24 и выше, а также DockerComposeplugin. Для этого в терминале сервера выполняется команда `docker --version`. Установка Docker на Ubuntu 22.04 производится по официальной инструкции из репозитория Docker Inc.

Развёртывание системы. В терминале сервера выполняется следующая последовательность команд. Переход в директорию проекта: `cd /home/user/mini-crm`. Запуск всех контейнеров в фоновом режиме: `docker-composeup -d`. После успешного запуска (занимает 30–60 секунд) необходимо выполнить загрузку начальных данных в базу данных командой: `dockerexecmini_crm_backendnprunseed`. По завершении все сервисы доступны: фронтенд — `http://адрес-сервера:5173`; API — `http://адрес-сервера:3000/api`.

Первоначальный вход в систему. После развёртывания в системе существуют демонстрационные учётные записи. Администратор: e-mail `admin@crm.ru`, пароль `admin123`. Менеджеры: e-mail `ivanov@crm.ru` / `kozlov@crm.ru` / `morozova@crm.ru`, пароль `admin123`. Рекомендуется изменить все пароли сразу после первого входа.

Добавление нового клиента. Для добавления клиента в систему

необходимо перейти в раздел «Клиенты» и нажать кнопку «+ Новый клиент» в правом верхнем углу экрана. В открывшемся диалоговом окне следует заполнить поля: «Полное имя» (обязательное поле); «Телефон» в формате +7; «Email»; «Адрес»; выбрать «Источник» обращения из выпадающего списка; при необходимости — добавить «Примечание». После нажатия кнопки «Создать» клиент появится в таблице. При наличии нескольких клиентов для быстрого поиска используется строка поиска в верхней части страницы.

Создание новой сделки. Новую сделку можно создать двумя способами. Первый способ — из раздела «Воронка сделок»: нажать кнопку «+ Новая сделка» в шапке страницы или кнопку «+ Добавить сделку» внизу нужной колонки. Второй способ — из карточки клиента: нажать кнопку «Перейти к воронке». В форме создания сделки указать: имя клиента (или выбрать из существующих); вид работ; сумму в рублях; ответственного менеджера.

Перемещение сделки по стадиям воронки. Для смены стадии сделки необходимо открыть карточку сделки (клик по заголовку на канбан-доске) и выбрать новую стадию из выпадающего списка «Стадия». После сохранения карточка переместится в соответствующую колонку, значение `last_activity_at` обновится и `risk_score` пересчитается.

Постановка задачи. Для создания задачи перейти в раздел «Задачи» и нажать «+ Новая задача». Указать заголовок задачи (обязательное поле), выбрать приоритет (Высокий / Средний / Низкий), установить дату и время выполнения через виджет выбора даты. При необходимости добавить описание. После создания задача появится в списке открытых задач. Для отметки задачи выполненной нажать на круглый значок статуса слева от заголовка.

Генерация коммерческого предложения. Перейти в раздел «Генератор КП». В поле «Имя клиента» ввести имя получателя в формате «Иванов Иван Иванович» (система извлечёт имя из первых двух слов для обращения). Выбрать «Вид работ» из выпадающего списка, ввести «Сумму по смете». Выбрать стиль: «Профессиональный» подходит для деловой переписки; «Дружелюбный» — для постоянных клиентов или неформального общения;

«Срочный (с акцией)» — создаёт ощущение ограниченного времени и стимулирует принятие решения. Нажать «Сформировать». Готовый текст КП можно скопировать в буфер обмена кнопкой «Копировать текст» для последующей отправки клиенту по e-mail или в мессенджере.

Работа с аналитикой. Раздел «Аналитика» предоставляет данные за текущий календарный месяц. При необходимости просмотреть данные за другой период следует обратиться к журналу аудита (доступен администратору) или экспортировать данные из PostgreSQL через инструмент администрирования базы данных.

3 Обоснование экономической эффективности результатов ВКР

3.1 Выбор и обоснование методики расчёта экономической эффективности

Методологическая основа. Экономическая эффективность разработки программного средства оценивается по нормативной методике расчёта затрат на создание программного обеспечения, основанной на нормативных показателях трудоёмкости функций программных средств [20]. Данная методика широко применяется в отечественной практике оценки ИТ-проектов и регламентирует порядок определения: объёма программного средства в условных машино-командах (УМК); нормативной трудоёмкости разработки; затрат на оплату труда разработчиков; затрат на машинное время; накладных и прочих расходов; полной себестоимости разработки.

Метод функциональных точек. Объём программного средства (V_o) определяется как сумма объёмов реализованных функций. Каждая функция идентифицируется по номеру из «Каталога функций программных средств» и имеет нормативный объём в УМК. Условная машино-команда (УМК) является единицей измерения сложности программы, независимой от языка программирования: она учитывает как непосредственно написанный код, так и используемые фреймворки и библиотеки.

Исходные данные для расчёта. Исходные нормативные показатели приведены в таблице 3.1. Значения установлены в соответствии с нормативными рекомендациями для организаций, осуществляющих разработку программных средств.

Таблица 3.1 - Исходные нормативные показатели для расчёта

№	Показатель	Обознач.	Значение	Ед. изм.
1	Коэффициент изменения скорости обработки информации	Кск	0,5	ед.
2	Численность разработчиков	Чр	1	чел.

Продолжение таблицы 3.1

3	Тарифная ставка 1-го разряда (месячная)	Сзм1	77 000	руб.
4	Тарифный коэффициент для должности разработчика	Кт	2,84	ед.
5	Фонд рабочего времени (месяц)	ФРВ	169,3	ч
6	Коэффициент потерь рабочего времени	Кп	1,1	ед.
7	Коэффициент премирования	Кпр	1,1	ед.
8	Норматив доп. заработной платы	Ндз	20	%
9	Ставка отчислений в Фонд социального страхования	Нфсзн	34	%
10	Цена одного машино-часа	Цм	2 200	руб.
11	Норматив прочих затрат	Нпз	10	%
12	Норматив накладных расходов	Ннр	100	%
13	Норматив расходов на сопровождение и адаптацию	Нрса	10	%
14	Уровень рентабельности	Урн	20	%
15	Ставка налога на добавленную стоимость	Нндс	20	%

Перечень функций разрабатываемого программного средства. Объем разрабатываемой системы определяется составом реализованных функций. Классификация и нормативные объемы функций приведены в таблице 3.2. Таблица 3.2 - Перечень и объем функций разрабатываемого программного средства

Код функции	Содержание функции	Объем, УМК
101	Организация ввода информации (формы создания клиентов, сделок, задач; валидация)	350
201	Генерация и управление структурой базы данных (6 сущностей, индексы, миграции)	3 450
301	Формирование последовательного файла (экспорт данных, отчетность)	340
401	Генерация рабочих программ (компиляция Vue SPA через Vite)	3 680
501	Монитор программного средства (управление маршрутизацией и навигацией)	350

Продолжение таблицы 3.2

507	Обеспечение интерфейса между компонентами (REST API, 20+ эндпоинтов)	730
605	Вспомогательные и сервисные программы (middleware, утилиты, logger)	460
704	Процессор отчётов (аналитика, дашборд, расчёт KPI)	1 070
705	Формирование и вывод данных на внешние носители (генератор КП, экспорт)	2 650
	Итого объём программного средства	13 080

Обоснование применяемой методики оценки экономического эффекта. Наряду с расчётом себестоимости разработки в настоящей главе проводится оценка экономического эффекта от внедрения системы.

Эффект рассчитывается по двум составляющим: прямая экономия на трудозатратах (сокращение непродуктивного рабочего времени) и косвенный прирост выручки (ликвидация потерь клиентских обращений).

Данный подход соответствует методологии оценки проектов автоматизации, рекомендованной в отечественной литературе [20, 21].

3.2 Расчёт показателей экономической эффективности

3.2.1 Расчёт себестоимости разработки программного средства.

Шаг 1. Расчёт нормативной трудоёмкости разработки.

Нормативная трудоёмкость определяется по каталогу в зависимости от общего объёма программного средства.

Для объёма $V_o = 13\,080$ УМК нормативная основная трудоёмкость составляет $T_o = 82$ чел.-дн. Дополнительная трудоёмкость на разработку документации вычисляется по формуле (3.1):

$$T_d = T_o \times N_{\text{док}} / 100, \quad (3.1)$$

где Ндок — норматив трудоёмкости документирования (принимается равным 10%).

$$Тд = 82 \times 10 / 100 = 8,2 \text{ чел.-дн.}$$

Нормативная трудоёмкость с учётом документирования по формуле (3.2):

$$Тн = То + Тд, \quad (3.2)$$

$$Тн = 82 + 8,2 = 90,2 \text{ чел.-дн.}$$

Скорректированная трудоёмкость с учётом коэффициента потерь рабочего времени Кп = 1,1 по формуле (3.3):

$$Тобщ = Тн \times Кп, \quad (3.3)$$

$$Тобщ = 90,2 \times 1,1 = 99,22 \approx 100 \text{ чел.-дн.}$$

Шаг 2. Расчёт заработной платы разработчика.

Часовая тарифная ставка разработчика вычисляется по формуле (3.4):

$$Сч = Сзм1 \times Кт / \text{ФРВ}, \quad (3.4)$$

$$Сч = 77\,000 \times 2,84 / 169,3 = 218\,680 / 169,3 = 1\,292,3 \text{ руб./ч.}$$

Основная заработная плата разработчика с учётом премирования по формуле (3.5):

$$ЗПосн = Сч \times Тобщ \times 8 \times Кпр, \quad (3.5)$$

где 8 — продолжительность рабочего дня в часах.

$$ЗПосн = 1\,292,3 \times 100 \times 8 \times 1,1 = 1\,137\,224 \approx 1\,137\,220 \text{ руб.}$$

Дополнительная заработная плата (норматив Ндз = 20%) по формуле (3.6):

$$ЗПдоп = ЗПосн \times Ндз / 100, \quad (3.6)$$

$$ЗП_{\text{доп}} = 1\,137\,220 \times 20 / 100 = 227\,444 \text{ руб.}$$

Общий фонд оплаты труда по формуле (3.7):

$$\text{ФОТ} = ЗП_{\text{осн}} + ЗП_{\text{доп}}, \quad (3.7)$$

$$\text{ФОТ} = 1\,137\,220 + 227\,444 = 1\,364\,664 \text{ руб.}$$

Шаг 3. Расчёт отчислений на социальные нужды.

Отчисления в Фонд социального страхования (ставка $Н_{\text{фсзн}} = 34\%$) по формуле (3.8):

$$\text{Офсс} = \text{ФОТ} \times Н_{\text{фсзн}} / 100, \quad (3.8)$$

$$\text{Офсс} = 1\,364\,664 \times 34 / 100 = 463\,986 \text{ руб.}$$

Шаг 4. Расчёт затрат на машинное время.

Время использования компьютера при разработке принимается равным 60% от общей трудоёмкости:

$$Т_{\text{маш}} = Т_{\text{общ}} \times 8 \times 0,60 = 100 \times 8 \times 0,60 = 480 \text{ ч.}$$

Затраты на машинное время по формуле (3.9):

$$З_{\text{м}} = Т_{\text{маш}} \times Ц_{\text{м}}, \quad (3.9)$$

$$З_{\text{м}} = 480 \times 2\,200 = 1\,056\,000 \text{ руб.}$$

Шаг 5. Расчёт накладных расходов.

Накладные расходы (норматив $Н_{\text{нр}} = 100\%$ от $ЗП_{\text{осн}}$) по формуле (3.10):

$$\text{НР} = ЗП_{\text{осн}} \times Н_{\text{нр}} / 100, \quad (3.10)$$

$$\text{НР} = 1\,137\,220 \times 100 / 100 = 1\,137\,220 \text{ руб.}$$

Шаг 6. Расчёт прочих затрат.

Прочие затраты (норматив Нпз = 10% от суммы ЗПосн, Офсс и Зм) по формуле (3.11):

$$З_{пр} = (З_{Посн} + Офсс + З_{м}) \times Н_{пз} / 100, \quad (3.11)$$

$$З_{пр} = (1\,137\,220 + 463\,986 + 1\,056\,000) \times 10 / 100 = 2\,657\,206 \times 0,10 = 265\,721 \text{ руб.}$$

Шаг 7. Расчёт полной себестоимости разработки.

Полная себестоимость разработки по формуле (3.12):

$$С = \Phi OT + Офсс + З_{м} + НР + З_{пр}, \quad (3.12)$$

$$С = 1\,364\,664 + 463\,986 + 1\,056\,000 + 1\,137\,220 + 265\,721 = 4\,287\,591 \text{ руб.}$$

Шаг 8. Расчёт расходов на сопровождение и адаптацию.

Расходы на сопровождение (норматив Нрса = 10% от себестоимости) по формуле (3.13):

$$З_{рса} = С \times Н_{рса} / 100, \quad (3.13)$$

$$З_{рса} = 4\,287\,591 \times 10 / 100 = 428\,759 \text{ руб.}$$

Общие единовременные затраты на разработку и внедрение:

$$З_{ед} = С + З_{рса} = 4\,287\,591 + 428\,759 = 4\,716\,350 \text{ руб.}$$

Вывод по расчёту себестоимости. Рыночная стоимость разработки аналогичного программного средства по нормативной методике составляет 4 716 350 рублей. Поскольку разработка выполняется в рамках выпускной квалификационной работы, фактические денежные расходы организации на разработку равны нулю. Приведённый расчёт демонстрирует значимость выполненной работы и экономию для предприятия по сравнению с заказом аналогичного решения у стороннего разработчика.

3.2.2 Расчёт экономического эффекта от внедрения.

Для оценки экономического эффекта от внедрения системы применяется метод расчёта годового экономического эффекта, включающий прямую экономию на трудозатратах и косвенный прирост выручки.

Расчёт 1. Прямая экономия на трудозатратах.

Экономия рабочего времени на поиск информации о клиентах. По данным обследования существующего процесса (раздел 1.4), среднее время поиска истории взаимодействий с клиентом составляет 15 минут (0,25 ч) на обращение. После внедрения системы поиск выполняется за 2–3 минуты (0,05 ч) — открытие карточки клиента. Число обращений в месяц — 40. Экономия времени в месяц по формуле (3.14):

$$E_{\text{поиск}} = n \times (t_{\text{до}} - t_{\text{после}}), \quad (3.14)$$

где n — число обращений в месяц; $t_{\text{до}}$, $t_{\text{после}}$ — время операции до и после внедрения.

$$E_{\text{поиск}} = 40 \times (0,25 - 0,05) = 40 \times 0,20 = 8,0 \text{ ч/мес.}$$

Экономия рабочего времени на подготовку отчётности. До внедрения: 3,5 ч/мес. После внедрения: 0,5 ч/мес (ручное скачивание готовых данных из системы).

$$E_{\text{отч}} = 3,5 - 0,5 = 3,0 \text{ ч/мес.}$$

Экономия на передаче задач между сотрудниками. До внедрения: 20 мин (0,33 ч) в среднем на передачу контекста при делегировании задачи. Среднее число передач в месяц — 6. После внедрения: 2 мин (0,03 ч) — переназначение ответственного в один клик.

$$E_{\text{задачи}} = 6 \times (0,33 - 0,03) = 6 \times 0,30 = 1,8 \text{ ч/мес.}$$

Суммарная экономия рабочего времени:

$$E_{\text{время}} = 8,0 + 3,0 + 1,8 = 12,8 \text{ ч/мес.}$$

Экономия на трудозатратах в денежном выражении по формуле (3.15):

$$\text{Э}_{\text{зат}} = \text{Е}_{\text{время}} \times \text{С}_{\text{ч}} \times 12, \quad (3.15)$$

где $\text{С}_{\text{ч}}$ — стоимость часа работы сотрудника (принимается 350 руб./ч — оценочно, на основе средней заработной платы в регионе).

$$\text{Э}_{\text{зат}} = 12,8 \times 350 \times 12 = 53\,760 \text{ руб./год.}$$

Расчёт 2. Прирост выручки от ликвидации потерянных обращений.

По данным обследования, в существующей системе не менее 2 обращений в месяц не получают ответа в течение 3 рабочих дней, что фактически означает потерю потенциальной сделки. При среднем чеке по ИТ-сервисным услугам 50 000 рублей и конверсии обращений в сделки 30% каждое потерянное обращение влечёт упущенную выручку:

$$\text{Упущ.1} = 50\,000 \times 0,30 = 15\,000 \text{ руб.}$$

Прирост выручки за год от ликвидации потерь по формуле (3.16):

$$\text{Э}_{\text{выр}} = \text{Н}_{\text{пот}} \times \text{Упущ.1} \times 12, \quad (3.16)$$

где $\text{Н}_{\text{пот}}$ — число потерянных обращений в месяц.

$$\text{Э}_{\text{выр}} = 2 \times 15\,000 \times 12 = 360\,000 \text{ руб./год.}$$

Расчёт 3. Годовой экономический эффект.

Годовой экономический эффект по формуле (3.17):

$$\text{Э}_{\text{год}} = \text{Э}_{\text{зат}} + \text{Э}_{\text{выр}}, \quad (3.17)$$

$$\text{Э}_{\text{год}} = 53\,760 + 360\,000 = 413\,760 \text{ руб./год.}$$

Расчёт 4. Годовые эксплуатационные затраты.

Годовые эксплуатационные затраты на содержание системы включают: аренду VPS-сервера ($800 \text{ руб.} \times 12 \text{ мес.} = 9\,600 \text{ руб.}$); регистрацию доменного имени (500 руб./год); итого: $\text{З}_{\text{экс}} = 10\,100 \text{ руб./год}$.

Расчёт 5. Чистый годовой экономический эффект.

Чистый годовой экономический эффект по формуле (3.18):

$$\mathcal{E}_\text{ч} = \mathcal{E}_\text{год} - \mathcal{Z}_\text{экс}, \quad (3.18)$$

$$\mathcal{E}_\text{ч} = 413\,760 - 10\,100 = 403\,660 \text{ руб./год.}$$

Расчёт 6. Срок окупаемости.

Срок окупаемости по формуле (3.19):

$$T_\text{ок} = \mathcal{Z}_\text{ед} / \mathcal{E}_\text{год}, \quad (3.19)$$

Поскольку фактические единовременные затраты ($\mathcal{Z}_\text{ед_факт}$) равны нулю (разработка выполнена в рамках ВКР), за $\mathcal{Z}_\text{ед}$ принимаются годовые эксплуатационные затраты $\mathcal{Z}_\text{экс}$:

$$T_\text{ок} = 10\,100 / 413\,760 \approx 0,024 \text{ года} \approx 8\text{--}9 \text{ дней.}$$

Расчёт 7. Сравнение с использованием коммерческих аналогов.

При использовании amoCRM для команды из 5 сотрудников: $599 \text{ руб.} \times 5 \times 12 = 35\,940 \text{ руб./год}$. При использовании Битрикс24 (тариф CRM): $1\,990 \text{ руб.} \times 12 = 23\,880 \text{ руб./год}$.

Экономия относительно amoCRM: $35\,940 - 10\,100 = 25\,840 \text{ руб./год}$.

Экономия относительно Битрикс24: $23\,880 - 10\,100 = 13\,780 \text{ руб./год}$.

Сводные технико-экономические показатели внедрения разработанной системы приведены в таблице 3.3.

Таблица 3.3 - Сводные технико-экономические показатели

Показатель	Значение	Ед. изм.
Объём программного средства	13 080	УМК
Нормативная трудоёмкость разработки	100	чел.-дн.
Рыночная стоимость разработки (расч.)	4 716 350	руб.
Фактические затраты на разработку	0	руб.
Годовые эксплуатационные затраты	10 100	руб./год
Экономия на трудозатратах (поиск + отчётность + задачи)	53 760	руб./год
Прирост выручки (ликвидация потерь обращений)	360 000	руб./год
Годовой экономический эффект	413 760	руб./год

Продолжение таблицы 3.3

Чистый годовой экономический эффект	403 660	руб./год
Срок окупаемости	≤9 дней	—
Экономия к amoCRM (5 сотрудников)	25 840	руб./год
Экономия к Битрикс24	13 780	руб./год
ROI (возврат инвестиций)	3 997%	—

Показатель рентабельности инвестиций (ROI) рассчитан по формуле:

$$ROI = (\mathcal{E}_ч / \mathcal{Z}_{экс}) \times 100\% = (403\,660 / 10\,100) \times 100\% = 3\,997\%$$

Столь высокое значение ROI объясняется нулевыми затратами на разработку: все инвестиции системы — это только эксплуатационные расходы (аренда сервера и домена), которые на порядок меньше экономического эффекта.

Выводы по разделу. Проведённые расчёты подтверждают высокую экономическую эффективность разработанной системы «Мини-CRM». Расчётный годовой экономический эффект составляет 413 760 рублей при годовых эксплуатационных затратах 10 100 рублей. Срок окупаемости эксплуатационных затрат — менее 9 дней. Рыночная стоимость аналогичной разработки, рассчитанная по нормативной методике, составляет 4 716 350 рублей, что характеризует значимость выполненной выпускной квалификационной работы.

Дополнительный нефинансовый эффект от внедрения: снижение риска потери клиентских обращений; повышение прозрачности загрузки сотрудников для руководителя; ускорение адаптации новых сотрудников (вся история взаимодействий доступна в системе); соответствие требованиям ФЗ-152 при обработке персональных данных клиентов; независимость от зарубежных вендоров и облачных сервисов.

Заключение

В настоящей выпускной квалификационной работе разработано веб-приложение «Мини-CRM для малого бизнеса» — специализированная информационная система, автоматизирующая процессы ведения клиентской базы, управления сделками и задачами, формирования оперативной отчётности для небольших команд.

В ходе выполнения работы решены все поставленные во введении задачи.

1. Исследованы теоретические основы CRM-систем: изучены понятие, история развития, функциональная классификация (операционные, аналитические, коллаборативные) и особенности применения CRM в малом бизнесе. Установлено, что малый бизнес нуждается в простых, бесплатных, русскоязычных решениях с возможностью self-hosted развёртывания.

2. Изучена нормативно-правовая база: Федеральный закон № 152-ФЗ «О персональных данных», Федеральный закон № 149-ФЗ, Постановление Правительства № 1119. Определены технические меры обеспечения соответствия, реализованные в системе: хэширование паролей bcrypt, JWT-аутентификация, ролевая модель доступа, журнал аудита, хранение данных на российском сервере.

3. Проведён анализ объекта исследования — туапсинского филиала ООО ИК «СИБИНТЕК». Описана специфика деятельности компании (ИТ-аутсорсинг в нефтяной отрасли), линейно-функциональная организационная структура из четырёх подразделений, степень автоматизации деятельности. Выявлено противоречие: высокая общая степень ИТ-автоматизации при полном отсутствии специализированного инструмента оперативного управления клиентскими взаимодействиями.

4. Выявлены и количественно оценены недостатки существующей технологии: потеря 2 обращений в месяц; 10–20 минут на поиск истории; 3,5 часа на ежемесячную отчётность; суммарно 8–10 непродуктивных часов на сотрудника в месяц.

5. Проведён сравнительный анализ пяти CRM-систем (Битрикс24, amoCRM, HubSpot CRM, EspoCRM, инструменты-заменители). Анализ подтвердил отсутствие оптимального решения для целевой аудитории и обосновал необходимость собственной разработки.

6. Разработаны диаграммы бизнес-процессов: контекстная диаграмма IDEF0 (уровень A0) и диаграмма декомпозиции (уровень A1) для AS-IS технологии; диаграмма BPMN 2.0 для TO-BE технологии с тремя дорожками (Клиент, Сотрудник, Система). Диаграммы наглядно демонстрируют качественные изменения в процессе после внедрения системы.

7. Спроектирована реляционная база данных из шести сущностей (users, clients, deal_stages, deals, tasks, audit_logs), приведённая к третьей нормальной форме. Разработана ER-диаграмма и Use-Case диаграмма с описанием 11 сценариев использования двумя актёрами (Менеджер, Администратор).

8. Реализовано полнофункциональное веб-приложение. Приложение включает: бэкенд на Node.js/Express.js с 20 REST API эндпоинтами; фронтенд на Vue.js 3 с 6 функциональными страницами; PostgreSQL 15 как СУБД; DockerCompose для контейнеризации. Общий объём исходного кода — 42 файла.

9. Разработано подробное руководство пользователя с описанием каждого раздела интерфейса и пошаговыми инструкциями для всех основных сценариев работы.

10. Выполнен расчёт экономической эффективности по нормативной методике. Рыночная стоимость разработки по расчёту составляет 4 716 350 рублей. Годовой экономический эффект от внедрения — 413 760 рублей. Срок окупаемости эксплуатационных затрат — менее 9 дней. ROI — 3 997%.

Инновационные функции. В рамках работы реализованы два оригинальных решения, отсутствующих в конкурирующих системах в бесплатном варианте. Шаблонный генератор коммерческих предложений поддерживает три стиля (профессиональный, дружелюбный, срочный) и 7 видов работ, формирует персонализированный текст мгновенно без обращения

к внешним сервисам. Алгоритм расчёта риска потери сделки (risk_score) анализирует три параметра (дни без активности, сумму, время в воронке) и обеспечивает проактивное управление портфелем сделок.

Практическая значимость. Разработанная система представляет собой готовое к развёртыванию решение, которое может быть немедленно применено в туапсинском филиале ООО ИК «СИБИНТЕК» или в любой другой малой организации сервисного направления. Отсутствие лицензионных затрат и поддержка self-hosted развёртывания делают систему доступной широкому кругу организаций.

Перспективы развития. В качестве направлений дальнейшего совершенствования системы рассматриваются: интеграция с мессенджерами (Telegram-бот для уведомлений и получения заявок); мобильная адаптация интерфейса (PWA или нативное мобильное приложение); экспорт отчётности в форматы Excel и PDF; интеграция с системами IP-телефонии для автоматической регистрации звонков; расширенная аналитика с прогнозированием выручки на основе исторических данных.

Все поставленные задачи выполнены в полном объёме, цель работы достигнута.

Список использованных источников

1. Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных» // Собрание законодательства РФ. — 2006. — № 31. — Ст. 3451.
2. Гринберг, П. CRM со скоростью света / П. Гринберг. — СПб.: Символ-Плюс, 2007. — 528 с.
3. Нарчук, А.В. Информационные технологии управления взаимоотношениями с клиентами // Экономика и управление. — 2023. — № 3. — С. 45–52.
4. Кузнецов, И.А., Смирнова, Е.В. Классификация CRM-систем в современных условиях // Информационные технологии в бизнесе. — 2022. — № 4. — С. 18–24.
5. Петров, С.А. Особенности внедрения CRM в малом бизнесе // Управление малым предприятием. — 2023. — № 2. — С. 30–38.
6. Федеральный закон от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации» // Собрание законодательства РФ. — 2006. — № 31. — Ст. 3448.
7. Официальный сайт ООО ИК «СИБИНТЕК» [Электронный ресурс]. — URL: <https://www.sibintek.ru> (дата обращения: 05.05.2026).
8. Битрикс24. Тарифы и возможности [Электронный ресурс]. — URL: <https://www.bitrix24.ru/prices/> (дата обращения: 06.05.2026).
9. amoCRM. Тарифы [Электронный ресурс]. — URL: <https://www.amocrm.ru/prices/> (дата обращения: 06.05.2026).
10. HubSpot CRM. Pricing [Электронный ресурс]. — URL: <https://www.hubspot.com/pricing> (дата обращения: 07.05.2026).
11. Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер. — М.: Вильямс, 2020. — 544 с.
12. Моуэт, Д. Использование Docker / Д. Моуэт. — СПб.: Питер, 2022. — 392 с.
13. Блэк, М. Аутентификация на основе JWT в Node.js [Электронный ресурс] // habr.com. — URL: <https://habr.com/ru/articles/572244/> (дата обращения: 15.04.2026).

14. Документация Vue.js 3 [Электронный ресурс]. — URL: <https://vuejs.org/guide/introduction.html> (дата обращения: 20.04.2026).
15. Документация Express.js [Электронный ресурс]. — URL: <https://expressjs.com/en/guide/routing.html> (дата обращения: 20.04.2026).
16. Моргунов, Е.П. PostgreSQL. Основы языка SQL / Е.П. Моргунов. — СПб.: BHV-Петербург, 2023. — 336 с.
17. Документация Pinia [Электронный ресурс]. — URL: <https://pinia.vuejs.org/introduction.html> (дата обращения: 22.04.2026).
18. Документация Sequelize ORM [Электронный ресурс]. — URL: <https://sequelize.org/docs/v6/> (дата обращения: 22.04.2026).
19. Документация PostgreSQL 15 [Электронный ресурс]. — URL: <https://www.postgresql.org/docs/15/> (дата обращения: 25.04.2026).
20. Методические рекомендации по оценке экономической эффективности разработки программных средств. — М.: ФГУП НТИЦ «Информрегистр», 2021. — 48 с.
21. Орлов, С.А. Технологии разработки программного обеспечения / С.А. Орлов. — СПб.: Питер, 2022. — 608 с.
22. Дейт, К.Дж. Введение в системы баз данных / К.Дж. Дейт. — М.: Вильямс, 2021. — 1328 с.
23. Якименко, А.В. Проектирование информационных систем / А.В. Якименко. — М.: ИНФРА-М, 2022. — 288 с.
24. Постановление Правительства РФ от 01.11.2012 № 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных».
25. Иванченко, О.В. Веб-разработка на современном стеке технологий // Информационные системы. — 2023. — № 1. — С. 12–19.
26. Котляров, В.П., Кириллова, Т.В. Основы тестирования программного обеспечения / В.П. Котляров. — М.: БИНОМ, 2021. — 285 с.
27. Сиразетдинов, Р.Р., Белоус, Д.В. Архитектура информационных систем // Техника средств связи. — 2020. — № 3. — С. 45–50.

28. Хэррон, Д. Node.js в действии / Д. Хэррон. — СПб.: Питер, 2022. — 424 с.
29. Документация Element Plus [Электронный ресурс]. — URL: <https://element-plus.org/en-US/> (дата обращения: 22.04.2026).
30. Реестр отечественного программного обеспечения Минцифры России [Электронный ресурс]. — URL: <https://reestr.digital.gov.ru> (дата обращения: 10.05.2026).

Приложение А

Диаграммы IDEF0 (AS-IS)



Рисунок А.1 - Контекстная диаграмма (IDEF0, уровень A0, AS-IS)

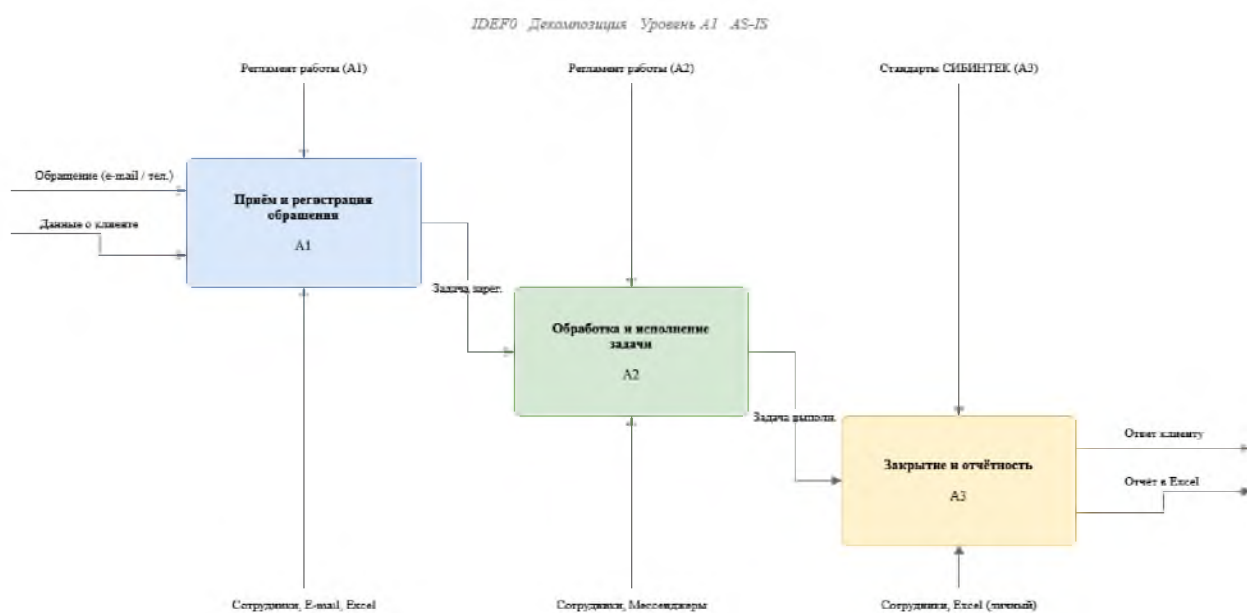


Рисунок А.2 - Диаграмма декомпозиции (IDEF0, уровень A1, AS-IS)

Приложение Б

Диаграмма BPMN (TO-BE)

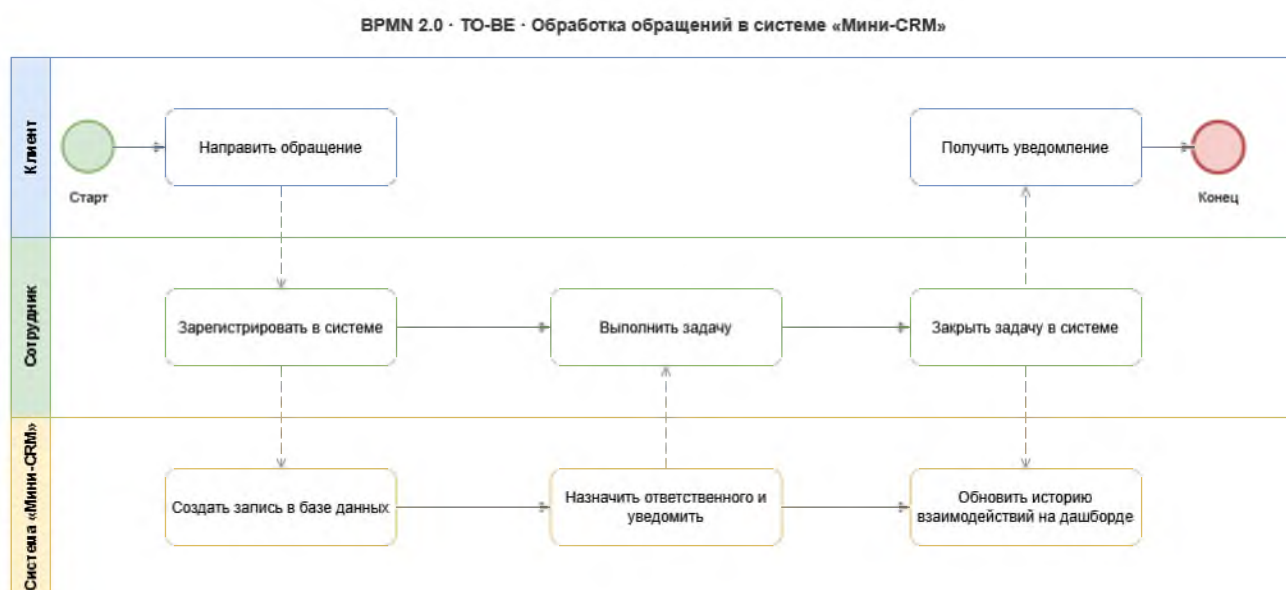


Рисунок Б.1 - Предлагаемая технология обработки обращений (BPMN 2.0)

Приложение В

ER-диаграмма базы данных

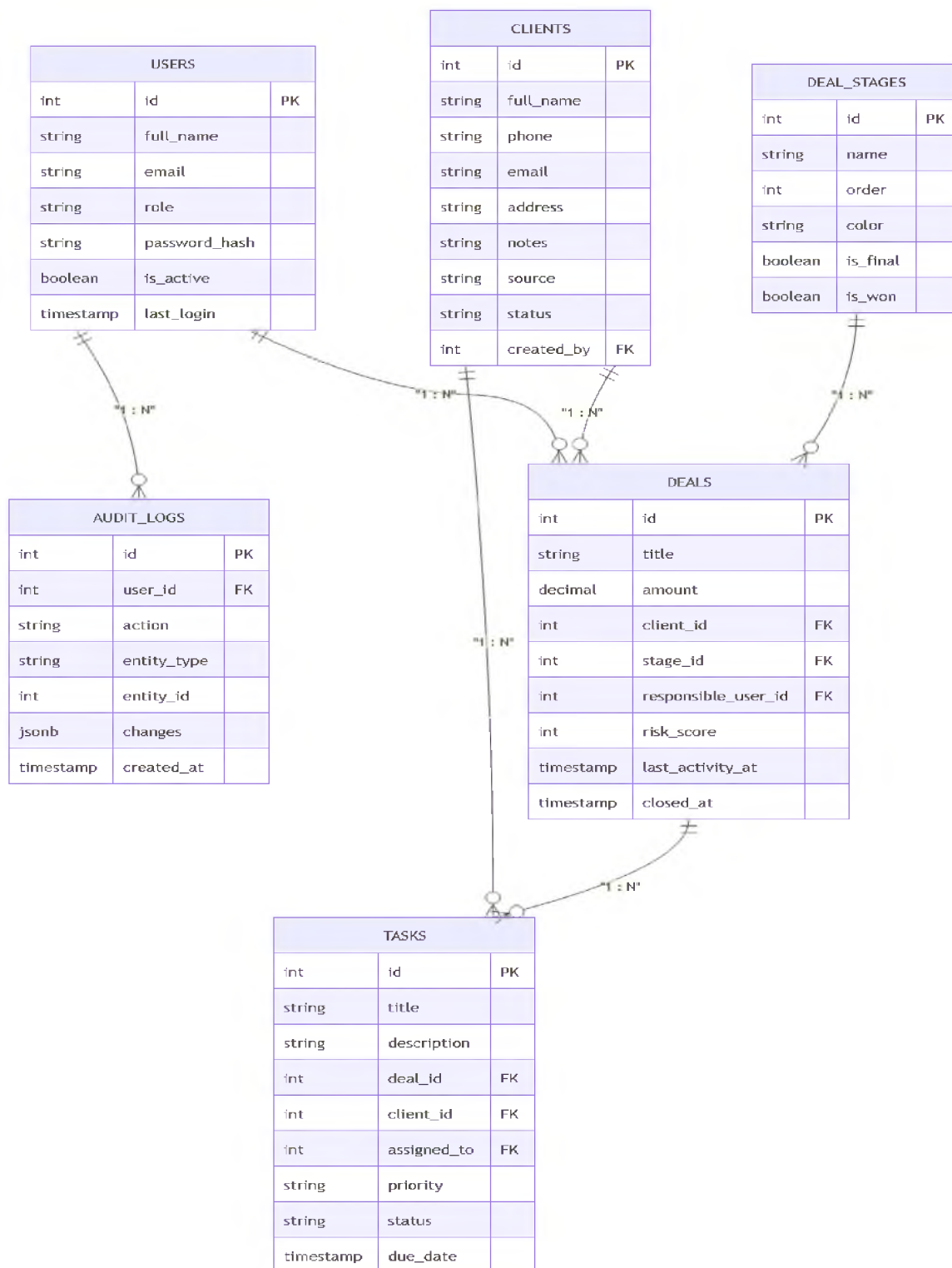


Рисунок В.1 - ER-диаграмма базы данных системы «Мини-CRM» (нотация Чена)

Приложение Г

Use-Case диаграмма

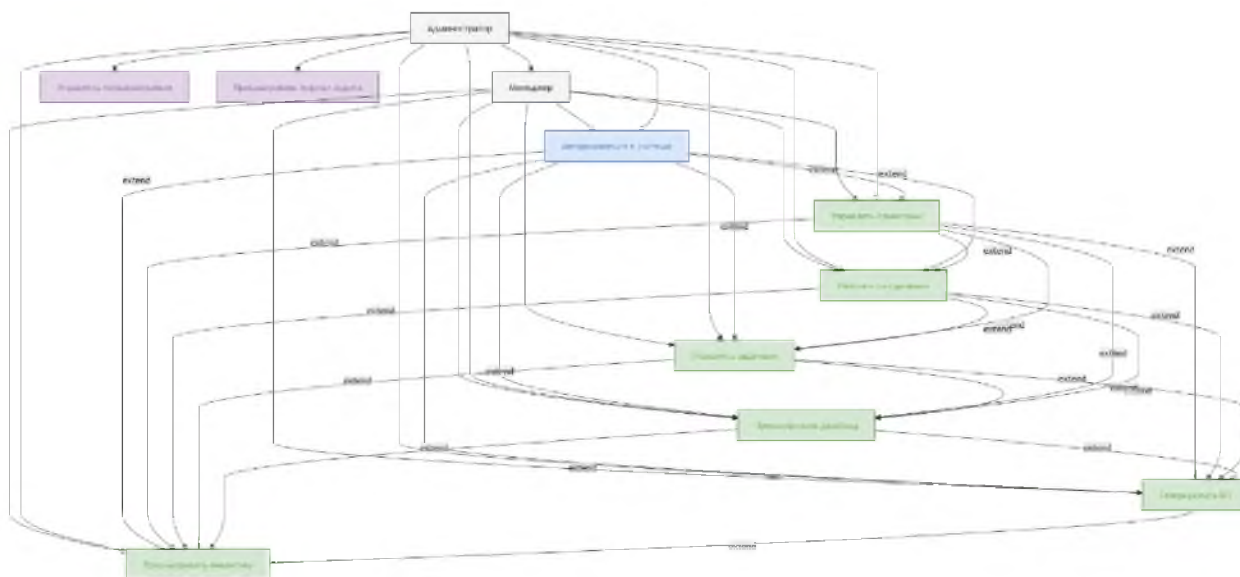


Рисунок Г.1 - Диаграмма вариантов использования системы «Мини-CRM»
(UML Use-Case)

Приложение Д

Фрагменты исходного кода

Листинг Д.1 — Алгоритм расчёта риска потери сделки (dealController.js)

```
function calcRiskScore(deal) { let score = 0; const days =
deal.last_activity_at ? Math.floor((Date.now() - new
Date(deal.last_activity_at)) / 86400000) : 0; if (days > 7)
score += 40; else if (days > 4) score += 20; else if (days >
2) score += 10; const amount = parseFloat(deal.amount) || 0;
if (amount > 300000) score += 15; else if (amount > 150000)
score += 8; const ageDays = Math.floor((Date.now() - new
Date(deal.created_at)) / 86400000); if (ageDays > 14) score +=
20; else if (ageDays > 7) score += 10; return Math.min(score,
100); }
```

Листинг Д.2 — Шаблонный генератор КП (proposalController.js)

```
exports.generateProposal = (req, res) => { const {
client_name, work_type, amount, tone, notes } = req.body;
if (!client_name || !work_type) return
res.status(400).json({ error: 'Укажите клиента и вид работ'
}); const proposal = buildProposal({ client_name,
work_type, amount, tone, notes }); res.json({ proposal }); }
```

Приложение Е

Скриншоты интерфейса приложения

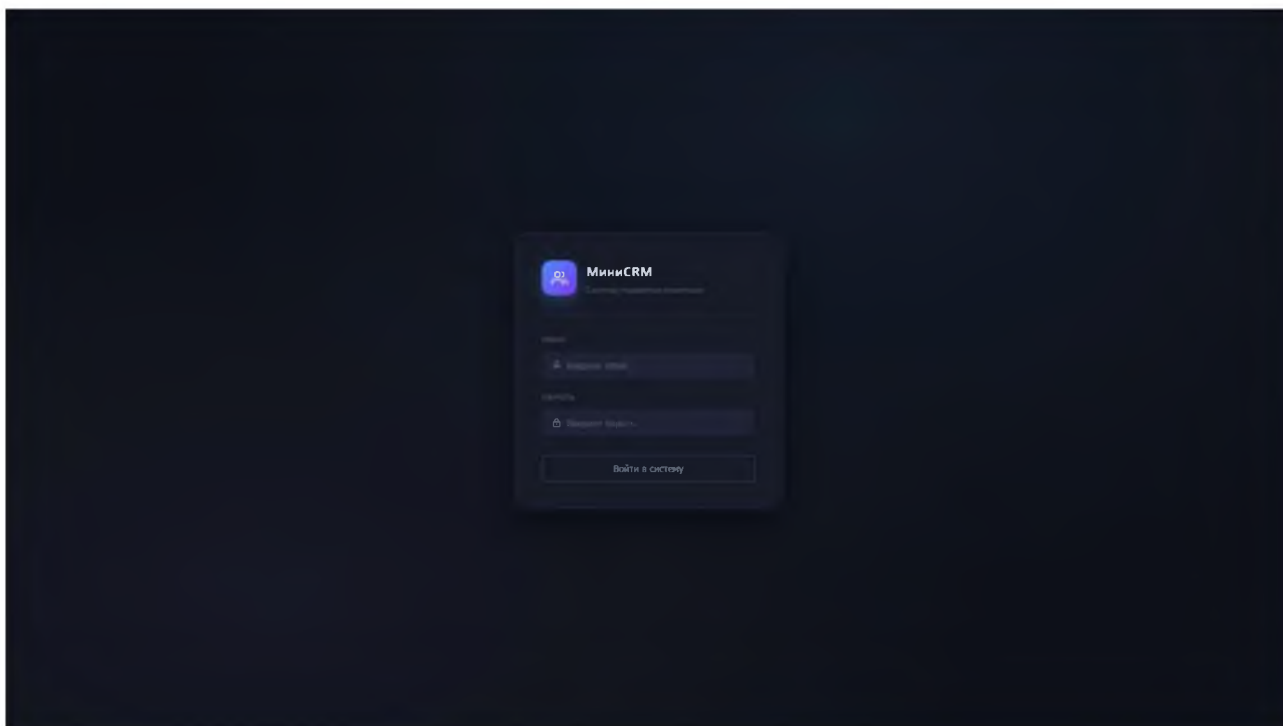


Рисунок Е.1 - Страница авторизации системы «Мини-CRM»

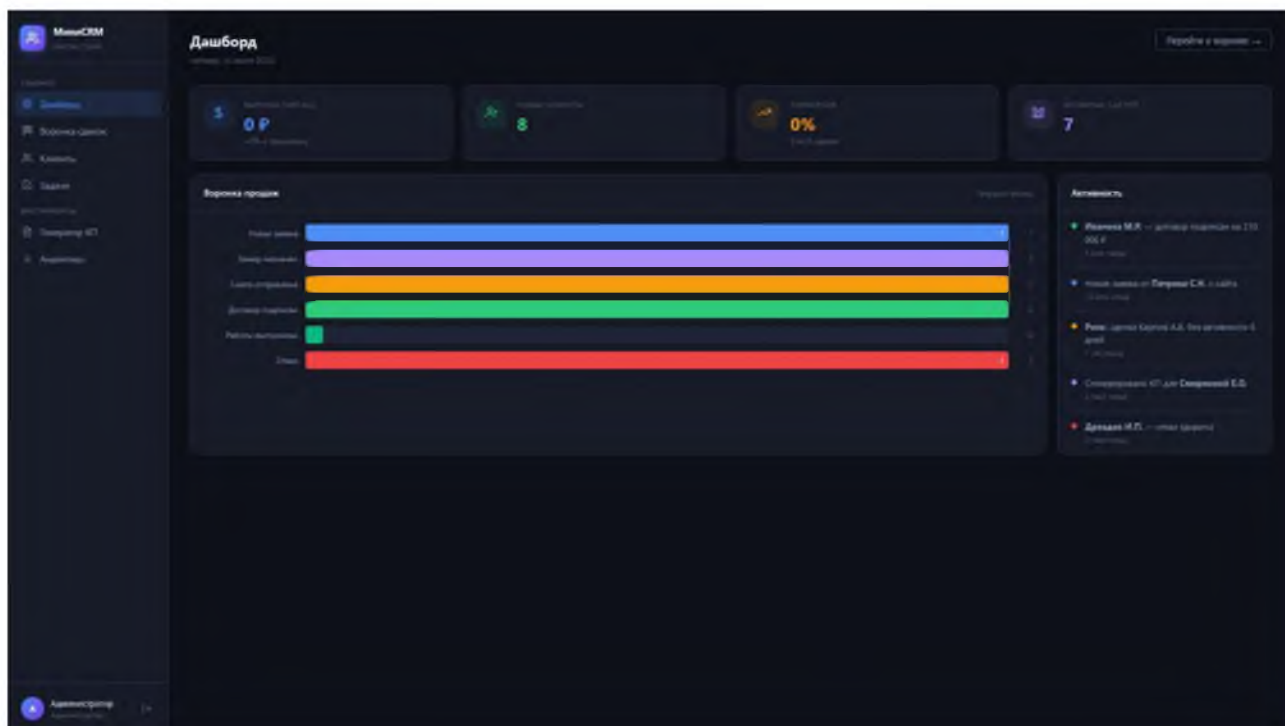


Рисунок Е.2 - Дашборд системы «Мини-CRM»

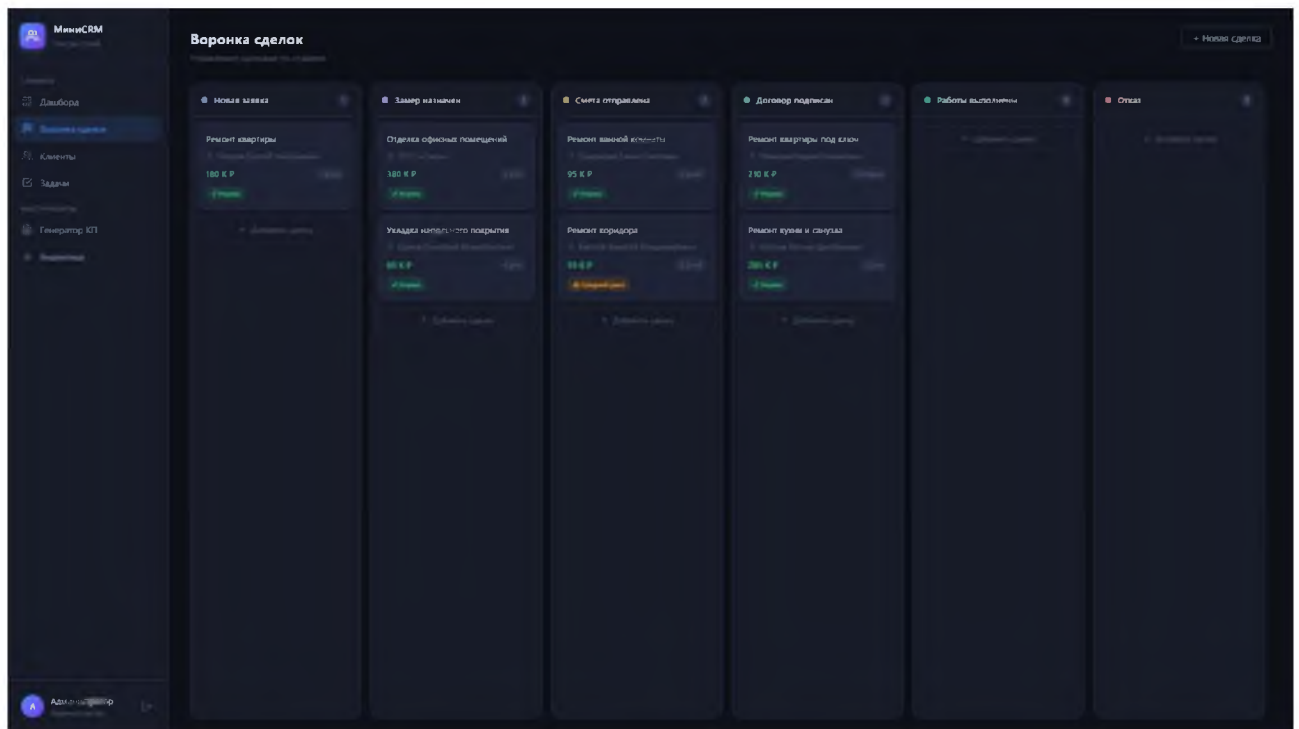


Рисунок Е.3 - Канбан-доска воронки сделок

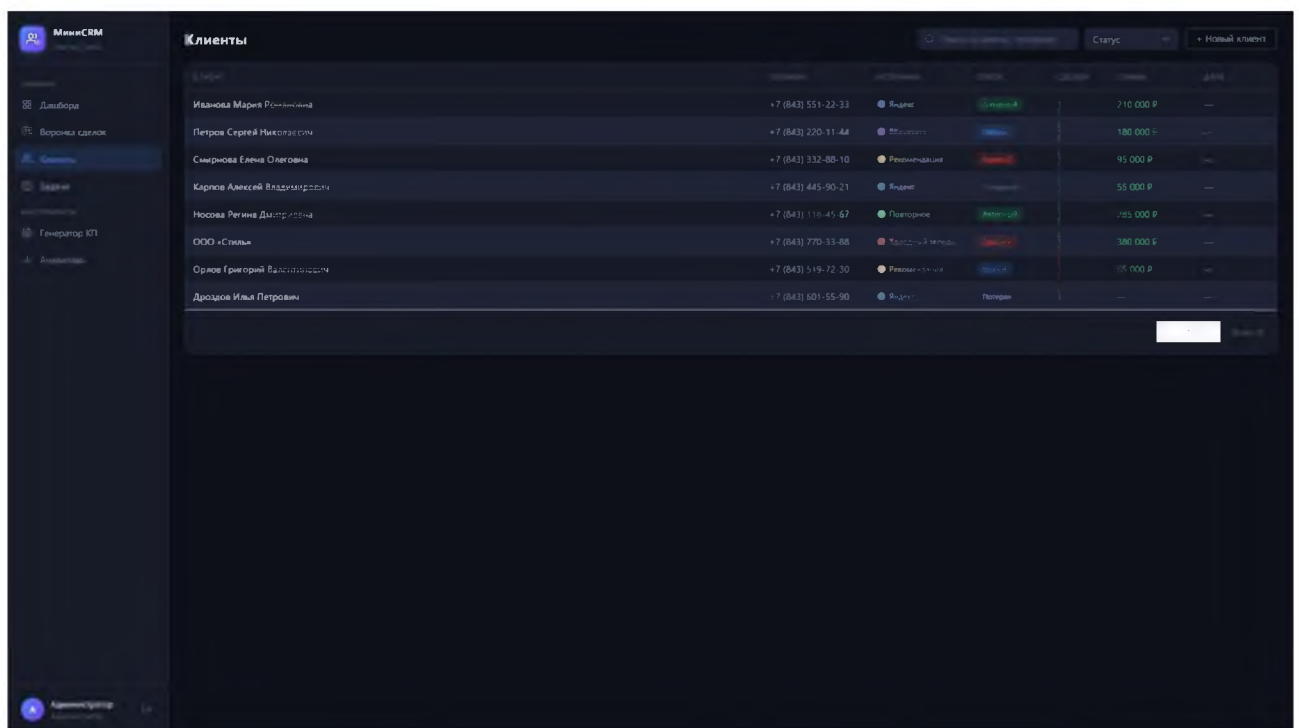


Рисунок Е.4 - Страница клиентской базы

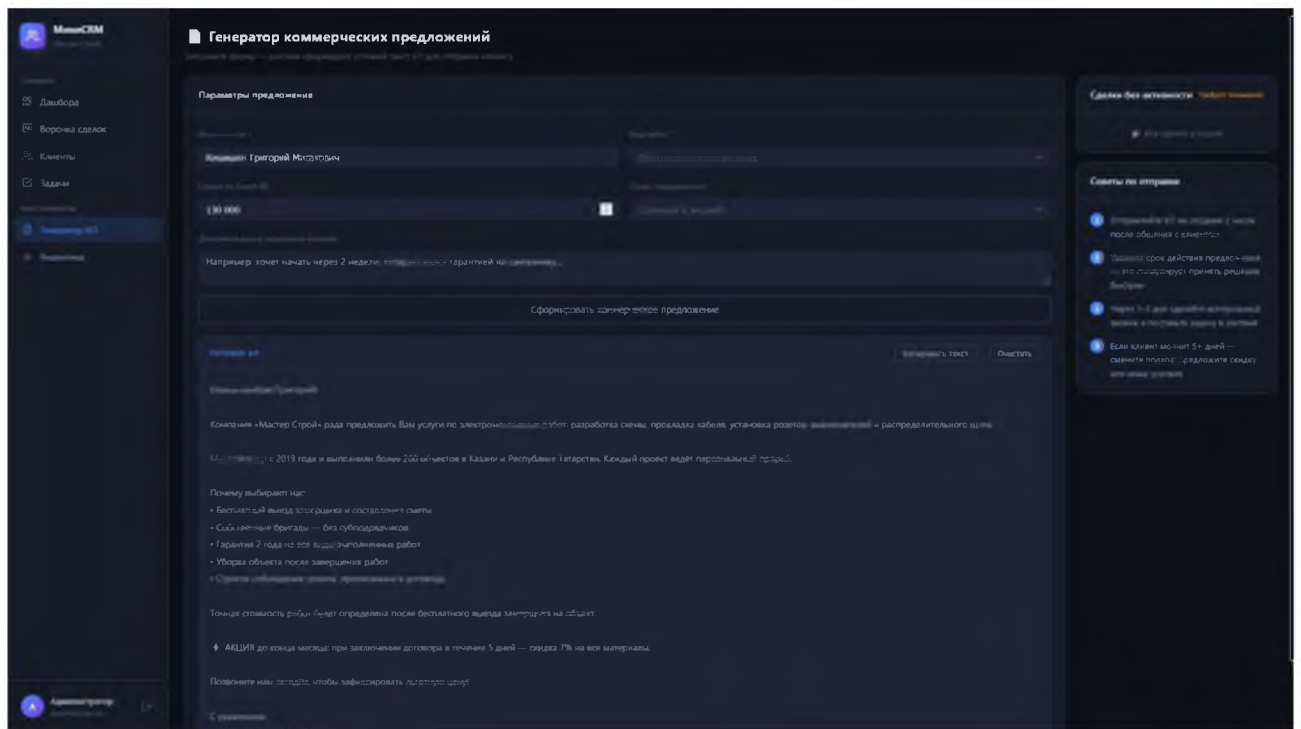


Рисунок Е.5 - Страница генератора коммерческих предложений