

*ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОМПЬЮТЕРНЫХ НАУК
КАФЕДРА АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ И ДИЗАЙНА
НАПРАВЛЕНИЕ 09.04.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ*

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА МАГИСТРА

на тему: Исследование, проектирование и реализация CRM-платформы
для автоматизации бизнес-процессов малого и среднего бизнеса на
основе современных JavaScript-фреймворков

Студент _____	Б.Д. Славинский
Руководитель работы _____	Е.Г. Коржов
Нормоконтроль проведен _____	А.С. Оганесян
Проверка на заимствования проведена _____	А.А. Петрыкина

Работа рассмотрена кафедрой и допущена к защите в ГЭК

Заведующий кафедрой _____	Е.Г. Коржов
Директор института _____	С.В. Солодов

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ «МИСИС»

УТВЕРЖДАЮ

Институт Информационных технологий
и компьютерных наук

Кафедра Автоматизированного

проектирования и дизайна

Направление 09.04.02 Информационные

системы и технологии _____

Зав. кафедрой _____ Е.Г. Коржов

« _____ » _____ 2025г.

**ЗАДАНИЕ
НА ВЫПОЛНЕНИЕ ВЫПУСКНОЙ
КВАЛИФИКАЦИОННОЙ РАБОТЫ МАГИСТРА**

Студенту группы МИСТ-23-3-3 Славинскому Борису Дмитриевичу

1. Тема работы: Исследование, проектирование и реализация CRM-платформы для автоматизации бизнес-процессов малого и среднего бизнеса на основе современных JavaScript-фреймворков
2. Цели Работы: Разработка CRM-системы на базе JavaScript-стека, предназначенной для управления клиентами, сделками, задачами и отчетностью
3. Основная литература, в том числе: НИР “ Исследование актуальности и потенциала применения CRM-платформы для малого и среднего бизнеса”, НИР “ Исследование актуальности разработки CRM-платформы для малого и среднего бизнеса”
4. Использование информационных технологий при проведении исследований: современный стек JavaScript-технологий: Next.js, TypeScript, Zustand, Tailwind CSS, Turso
5. Руководитель диссертации: Заведующий кафедрой автоматизированного проектирования и дизайна, к.т.н. доцент, Коржов Евгений Геннадьевич
(подпись)

Дата выдачи задания 03.03.2025

Задание принял к исполнению студент _____

(подпись)

Аннотация

Выпускная квалификационная работа посвящена созданию CRM-системы, ориентированной на автоматизацию бизнес-процессов в малом и среднем бизнесе. Основной целью является создание универсального решения, позволяющего эффективно управлять клиентами, сделками и задачами, учитывая реальные потребности предприятий и недостатки существующих CRM-систем.

В ходе выполнения работы была проведена оценка текущих решений на рынке, сформулированы функциональные и технические требования, спроектирована архитектура информационной системы и реализована полнофункциональная CRM-платформа. В реализации использован современный стек JavaScript-технологий: Next.js, TypeScript, Zustand, Tailwind CSS, Turso.

Выпускная квалификационная работа изложена на 69 страницах, содержит 25 рисунков, 6 таблиц и список литературы из 24 наименований.

Annotation

The graduate qualification work is dedicated to the development of a CRM platform aimed at automating business processes in small and medium-sized enterprises. The main objective is to create a universal solution that enables efficient management of clients, deals, and tasks, taking into account the actual needs of businesses and the limitations of existing CRM systems.

As part of the work, an assessment of current market solutions was carried out, functional and technical requirements were formulated, the architecture of the information system was designed, and a fully functional CRM platform was implemented. The development uses a modern JavaScript technology stack: Next.js, TypeScript, Zustand, Tailwind CSS, and Turso.

The graduate work is presented on 69 pages, contains 25 figures, 6 tables, and a list of sources used from 24 titles.

Оглавление

Введение	6
Раздел 1. Анализ рынка и подходов	8
1.1 Анализ потребностей малого и среднего бизнеса в CRM-автоматизации	8
1.2 Классификация CRM-систем	12
1.3 Анализ существующих CRM-систем	17
1.4 Выбор и обоснование метода разработки	23
1.5 Выводы к разделу 1	25
Раздел 2. Проектирование CRM-системы	27
2.1 Формулировка требований к CRM-системе	27
2.2 Моделирование бизнес-процессов (BPMN)	30
2.3 Выбор технологий	34
2.3.1 Frontend: Next.js и TypeScript	34
2.3.2 Backend: API Routes и Turso	34
2.3.3 Стилизация: Tailwind CSS и Headless UI	36
2.3.4 Управление состоянием: Zustand	37
2.3.5 Хостинг и деплой: Vercel	37
2.4 Архитектура CRM-системы	37
2.5 UI/UX прототипирование	41
2.6 Выводы к разделу 2	46
Раздел 3. Практическая реализация и результаты	48
3.1 Реализация ключевых модулей	48
3.1.1 Модуль управления клиентами	48
3.1.2 Модуль работы со сделками	50
3.1.3 Модуль задач	51
3.1.4 Модуль авторизации/аутентификации	52
3.2 Структура данных	54
3.3 Демонстрация интерфейса	57
3.4 Количественные и качественные результаты	61
3.5 Выводы к разделу 3	63
Заключение	65
Список используемой литературы	67

Введение

Актуальность работы. В настоящее время в ведении бизнеса возрастает потребность в эффективном управлении клиентскими отношениями и внутренними процессами. CRM-системы (Customer Relationship Management) стали надёжным и верным инструментом, который помогает малым и средним компаниям работать продуктивнее, лучше взаимодействовать с клиентами и уверенно развиваться в конкурентной среде [4].

Согласно исследованию Сбер Бизнес Софт и TenChat “Использование CRM-систем в России 2024”, уровень внедрения таких решений в бизнес-среде продолжает расти, однако многие компании по-прежнему сталкиваются с трудностями [12]. Основные барьеры – высокая стоимость, сложность настройки под конкретные процессы и перегруженность лишними функциями, которые не нужны малому бизнесу. Это создаёт спрос на более простые, доступные и адаптируемые решения, не требующие сложной интеграции и длительного обучения персонала.

Обзор существующих CRM-систем показывает, что большинство популярных решений ориентированы на крупные организации и включают универсальные, но часто избыточные модули. Такие системы не всегда подходят небольшим компаниям, которым важно быстро внедрить самые необходимые инструменты для повседневной работы. Именно поэтому актуальна разработка лёгкой и гибкой CRM-системы, адаптированной под нужды малого и среднего бизнеса.

Объектом исследования в данной работе являются бизнес-процессы в организациях малого и среднего масштаба, которые можно автоматизировать с помощью CRM-систем.

Предметом исследования выступают подходы к проектированию и реализации веб-приложений на основе современных JavaScript-фреймворков, используемые при создании CRM-систем.

Цель исследования. Разработка CRM-системы на базе JavaScript-стека, предназначенной для управления клиентами, сделками, задачами и отчетностью. Исследование направлено на создание удобного, гибкого и адаптируемого инструмента, отвечающего потребностям малого и среднего бизнеса и учитывающего особенности их внутренних процессов.

Для достижения этой цели были поставлены следующие **задачи**:

1. Изучить рынок и выделить ключевые требования к CRM для малого и среднего бизнеса;
2. Разработать архитектуру системы с использованием современных фреймворков;
3. Реализовать полноценный прототип с авторизацией, CRUD-функционалом и аналитикой.

Практическая значимость работы заключается в создании CRM-системы, предназначенной для автоматизации ключевых бизнес-процессов в малых и средних предприятиях. Разработка и внедрение такой системы позволят упростить управление клиентскими данными, сделками, задачами и внутренними коммуникациями, что в конечном итоге способствует повышению эффективности работы компаний.

Раздел 1. Анализ рынка и подходов

1.1 Анализ потребностей малого и среднего бизнеса в CRM-автоматизации

Малый и средний бизнес (МСБ) играет важную роль в экономике России. Доля этого сегмента ВВП страны составляет более 20%, а в структуре занятости – более 25% [7]. Однако компании МСБ функционируют в условиях ограниченных ресурсов, высокой конкуренции и постоянной необходимости адаптации к изменениям рыночной среды.

Цифровизация бизнес-процессов становится необходимым условием устойчивого развития [11]. Использование современных цифровых решений позволяет повысить управляемость, прозрачность операций и эффективность взаимодействия с клиентами. Несмотря на это, уровень цифровизации в секторе МСБ существенно отстаёт от показателей крупных компаний [7].

Одним из наиболее перспективных направлений цифровизации является внедрение CRM-систем. Это программные решения, предназначенные для автоматизации процессов взаимодействия с клиентами: от хранения контактных данных до контроля сделок, задач и отчетности [4]. Такие системы позволяют структурировать работу отдела продаж, оптимизировать внутренние процессы и повысить качество обслуживания.

Для МСБ особенно важно не упускать клиентов и держать с ними связь в процессе сотрудничества. Основные цели CRM-систем изложены ниже (Рисунок 1).



Рисунок 1 – Цели CRM-системы

Согласно исследованию Сбер Бизнес Софт и TenChat, CRM-системы используются в 49 % российских компаний, при этом полный охват достигается лишь в 22 % случаев, а остальные применяют их частично или готовятся к внедрению (Рисунок 2) [12]. В качестве основных препятствий указываются высокая стоимость внедрения, сложность настройки под внутренние процессы и избыточность функциональности, не отвечающей реальным потребностям. Кроме того, пользователи отмечают отсутствие адаптации систем к российским реалиям, включая локализацию и соблюдение нормативных требований.

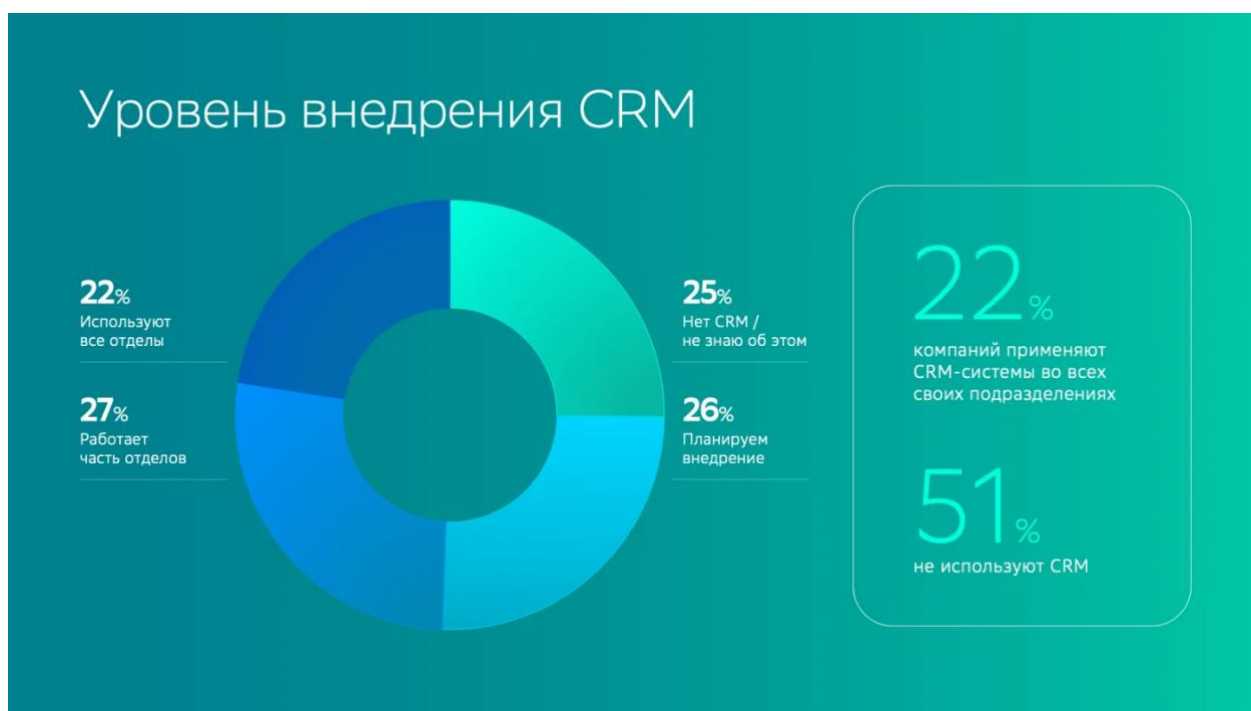


Рисунок 2 – Уровень внедрения CRM, согласно исследованию Сбер Бизнес Софт и TenChat

Вместо современных CRM-систем компании используют таблицы, мессенджеры, электронную почту или социальные сети для ведения базы клиентов и взаимодействия с ними (Рисунок 3) [12].

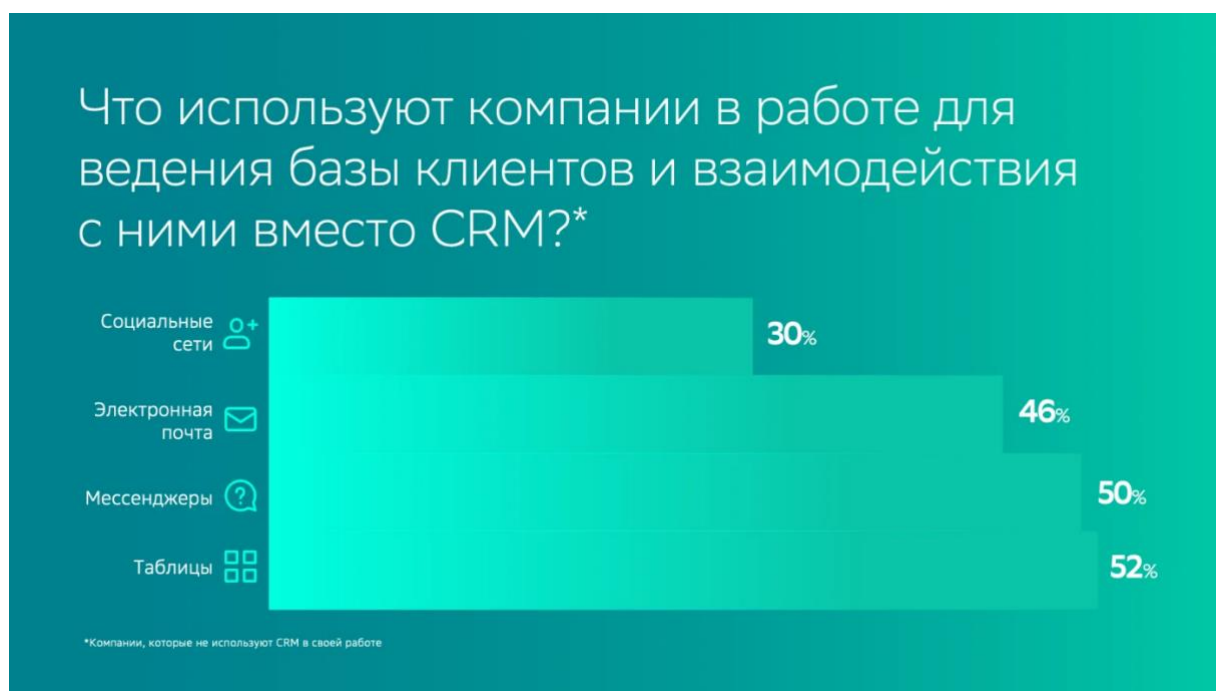


Рисунок 3 – Инструменты, заменяющие CRM в компаниях, согласно исследованию Сбер Бизнес Софт и TenChat

Большинство коммерчески доступных решений ориентированы на корпоративный сегмент и имеют сложную структуру, требующую участия специалистов на этапе внедрения и сопровождения. Это снижает доступность таких систем для малых предприятий, которые не располагают значительными IT-ресурсами.

Для МСБ особенно актуальны решения с модульной архитектурой и гибкой конфигурацией. Подобный подход позволяет организациям поэтапно внедрять функциональные блоки, начиная с базового набора, и постепенно расширять систему в соответствии с ростом и развитием бизнеса. Такие решения чаще всего реализуются с использованием современных JavaScript-фреймворков.

Среди ключевых требований, которые МСБ предъявляет к CRM-системам, можно выделить:

- простоту и интуитивность пользовательского интерфейса;
- минимальный и понятный набор базовых функций;

- низкий порог вхождения для сотрудников;
- быстрое развёртывание без длительной настройки;
- возможность кастомизации без привлечения разработчиков;
- совместимость с отечественными сервисами (мессенджерами, онлайн-кассами, телефонией и пр.);
- соответствие требованиям по защите персональных данных;
- прозрачную ценовую политику и наличие бесплатных тарифов или open source-альтернатив.

Таким образом, отечественный рынок МСБ демонстрирует устойчивый спрос на CRM-системы, которые сочетают технологичность, простоту внедрения и доступность [24]. Выявленные особенности подтверждают необходимость создания CRM-системы, адаптированной под нужды МСБ и основанной на современных подходах к веб-разработке.

1.2 Классификация CRM-систем

CRM-системы, в зависимости от их функциональности и отраслевой специализации, можно классифицировать на разные группы (Рисунок 4). Это помогает лучше ориентироваться во всем многообразии доступных на рынке систем и выбрать наиболее подходящее из них, учитывая специфику и особенности конкретной компании.



Рисунок 4 – Классификация CRM-систем

1. По способу развертывания. В зависимости от способа развертывания CRM-системы делятся на облачные и коробочные (локальные).

- **Облачные CRM** работают по модели SaaS (Software as a Service) и размещаются на удаленных серверах. Пользователи получают доступ к такой системе через интернет, без необходимости установки ПО на локальные устройства. Зачастую облачные CRM оплачиваются по подписке. Исходя из перечисленной выше информации можно выделить главные плюсы такого подхода – невысокий порог входа, минимальные начальные расходы, быстрое внедрение и возможность подключения к системе с любого устройства, имеющего стабильное интернет-соединение.
- **Коробочные CRM**, в отличие от облачных, устанавливаются на собственные сервера компании, что обеспечивает высокий уровень контроля данных и возможность гибкой настройки ПО

под внутренние процессы. При таком подходе все обслуживание ведется силами организации, либо с помощью сторонних специалистов, что требует больших затрат. Однако коробочные CRM идеально подойдут для тех компаний, где важна повышенная безопасность данных и возможность интеграции с внутренними системами компании.

2. По назначению. CRM-системы возможно также классифицировать по их основному назначению. Здесь можно выделить 4 основных класса.

- **CRM для продаж** ориентированы на управление лидами, клиентской базой, воронкой продаж и сделками. Они используются преимущественно в таких сферах, как торговля, B2B продажи и сервисные компании.
- **CRM для маркетинга** включает в себя такие функции, как сегментация базы клиентов, проведение e-mail и мессенджер-рассылок, управление компаниями, аналитика откликов и эффективность рекламы. Такие решения актуальные для e-commerce и digital-агентств.
- **CRM для клиентского сервиса** сосредоточены на поддержке клиентов, а именно на ведении обращений, учете тикетов, управлении SLA и формировании базы знаний. Подобные системы больше всего востребованы в службах технической и информационной поддержки, страховых компаниях и call-центрах.
- **Универсальные CRM** сочетают в себе инструменты всех вышеперечисленных категорий. Они имеют модульную структуру, которая позволяет адаптировать систему под нужды конкретного бизнеса.

3. По функциональной направленности. Исходя из того, какой функционал и модули предоставляет система, CRM можно поделить на четыре группы.

- **Операционные CRM-системы** автоматизируют повседневные процессы, которые связаны с маркетингом, продажами и обслуживанием клиентов. Сюда можно отнести:
 - ведение базы клиентов и сделок;
 - управление контактами;
 - постановку задач и напоминаний;
 - шаблоны писем и скриптов общения;
 - отслеживание этапов воронки продаж и т.д.
- **Аналитические CRM-системы** сфокусированы на обработке данных и ведении отчетности. Сюда относятся такие функции, как:
 - построение индивидуальных и сводных отчетов;
 - визуализация ключевых показателей (KPI);
 - сегментация клиентов по заданным критериям;
 - прогнозирование продаж и клиентской активности;
 - анализ эффективности маркетинговых кампаний и т.д.
- **Коллаборативные CRM-системы** направлены на обеспечение слаженной работы между разными отделами внутри компании, взаимодействующих с клиентами. Ключевые функции этой подгруппы:
 - централизованное хранение истории взаимодействий;

- единый интерфейс для работы отделов маркетинга, продаж и поддержки;
 - маршрутизация заявок и обращений;
 - поддержка общения с клиентами через разные каналы (почта, чат, звонки);
 - внутренняя коммуникация между сотрудниками.
- **Комбинированные CRM-системы** сочетают в себе функции всех трех вышеуказанных групп. Это некий универсальный инструмент, который по статистике чаще всего используется в крупных компаниях. Сюда можно отнести такие CRM, как “Битрикс24”, “Мегаплан” и AmoCRM.

4. По отраслевой специализации. CRM-системы можно также поделить по уровню универсальности на 2 типа.

- **Универсальные** – подходят под большинство сфер деятельности и объединяют в себе базовый функционал и гибкие настройки.
- **Отраслевые** – ориентированы на определенный рынок (финансы, строительство, образование, медицина и пр.). CRM такого типа содержат модули, разработанные с учетом характерных процессов и терминов конкретной отрасли.

Вне зависимости от отраслевой специализации, функциональности или назначения, CRM должна включать в себя базовый функционал, достаточный для закрытия минимальных потребностей (Рисунок 5).



Рисунок 5 – Базовый функционал CRM

1.3 Анализ существующих CRM-систем

С каждым годом CRM-системы становятся более популярными среди компаний различного масштаба. На российском рынке представлено множество решений, отличающихся между собой по функциональности, модели распространения, цене и степени адаптации под конкретные бизнес-сценарии.

В число наиболее распространённых в России можно отнести следующие инструменты: Bitrix24 [17], AmoCRM [18], Мегаплан; а к международным решениям: HubSpot CRM [19], Zoho CRM, Salesforce.

Ниже будет представлен краткий обзор нескольких популярных CRM-систем российского и зарубежного рынка с акцентом на ключевые функции, ценовую политику, преимущества и ограничения. Эти данные основаны на открытых источниках и аналитике [12][17][18][19].

1. AmoCRM (Рисунок 6) – российская CRM-система, запущенная в 2010 году и ориентированная преимущественно на малый и средний

бизнес. Основной акцент сделан на автоматизации продаж, визуализации воронки, быстрой интеграции с мессенджерами и простоте настройки процессов без участия программистов. Сервис активно используется в сегменте B2B и B2C, особенно среди компаний в сфере услуг и розничной торговли.

- **Ключевые функции:** учёт лидов и сделок, визуальная воронка продаж, интеграции с мессенджерами (Telegram, WhatsApp), email-рассылки, встроенные боты.
- **Ценовая политика:** подписка, от базового тарифа до PRO; цена зависит от количества пользователей и используемых функций.
- **Преимущества:** удобный интерфейс “из коробки”, быстрая настройка, высокая интеграция с российскими сервисами.
- **Ограничения:** отсутствие расширенной аналитики и отчётности без доработок; кастомизация требует работы с API.

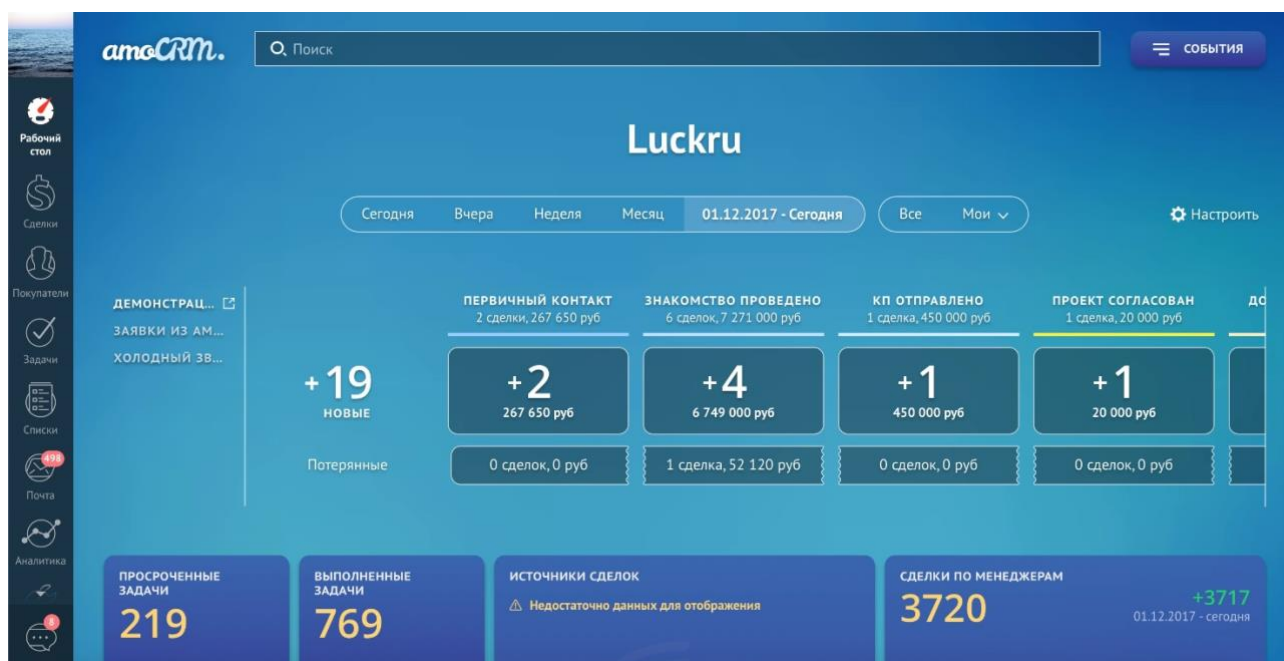


Рисунок 6 – Интерфейс AmoCRM

2. Bitrix24 (Рисунок 7) – многофункциональная платформа, разработанная в 2012 году и включающая CRM-модуль,

инструменты для совместной работы, телефонию, задачи, календари и чат. Продукт широко распространён как в России, так и за её пределами, и используется более чем 15 миллионами организаций. Среди клиентов – компании из сферы образования, строительства, интернет-торговли и даже участники списка Forbes.

- **Ключевые функции:** задачи, чаты, телефония, конструктор сайтов, HR-модули.
- **Ценовая политика:** бесплатный тариф с базовой функциональностью; расширенные модули доступны в платных версиях.
- **Преимущества:** единая платформа, сильная экосистема, адаптация под российское законодательство.
- **Ограничения:** перегруженность интерфейса, сложности при начальном освоении, необходимость в адаптации для глубокой настройки.

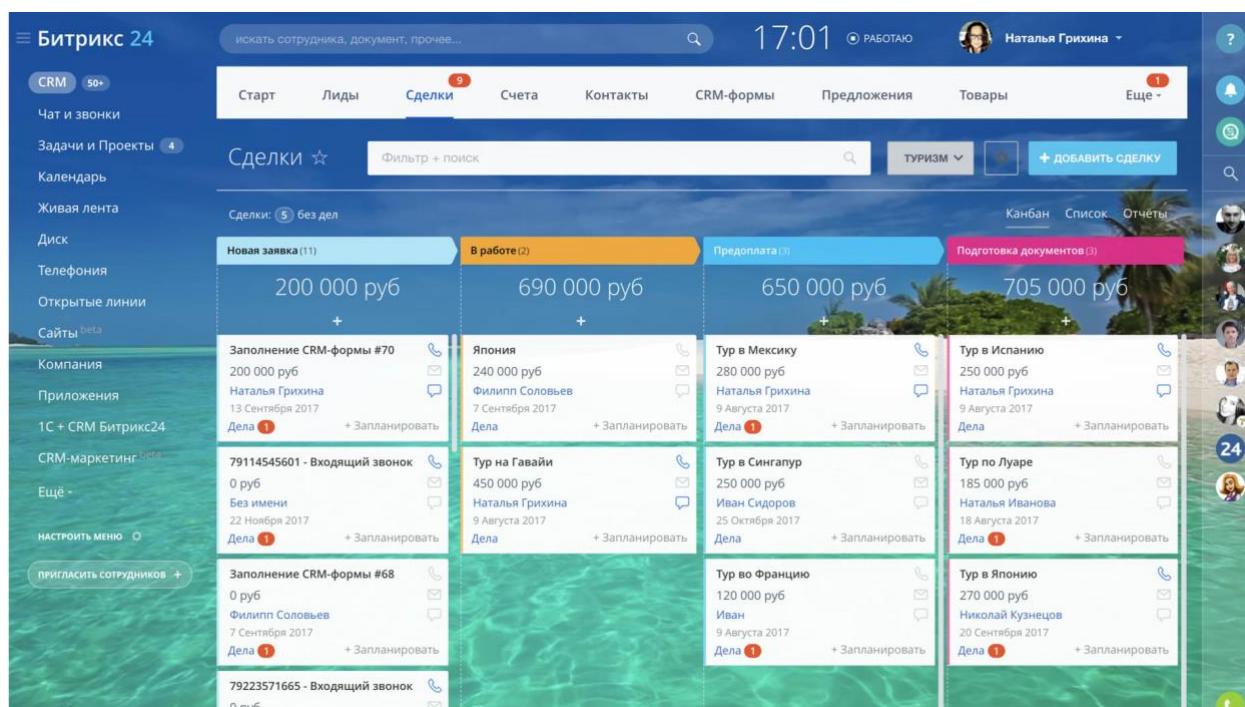


Рисунок 7 – Интерфейс Bitrix24

3. Zoho CRM (Рисунок 8) – международная облачная CRM-система, входящая в состав экосистемы Zoho Corporation. Решение появилось в 2005 году и с тех пор активно развивается, предлагая широкие возможности по автоматизации продаж, интеграции с офисными приложениями и гибкой настройке. В числе пользователей — такие компании, как Nike, Cisco, Amazon и Google, а общее число клиентов превышает 250 000 по всему миру.

- **Ключевые функции:** управление лидами, автоматизация воронки, интеграция с Zoho-продуктами (Books, Desk), отчётность.
- **Ценовая политика:** помесечная подписка, несколько уровней тарифов.
- **Преимущества:** масштабируемость, широкий функционал даже в базовых планах, мобильное приложение.
- **Ограничения:** некоторые функции слабо адаптированы под российские реалии; англоязычный интерфейс по умолчанию.

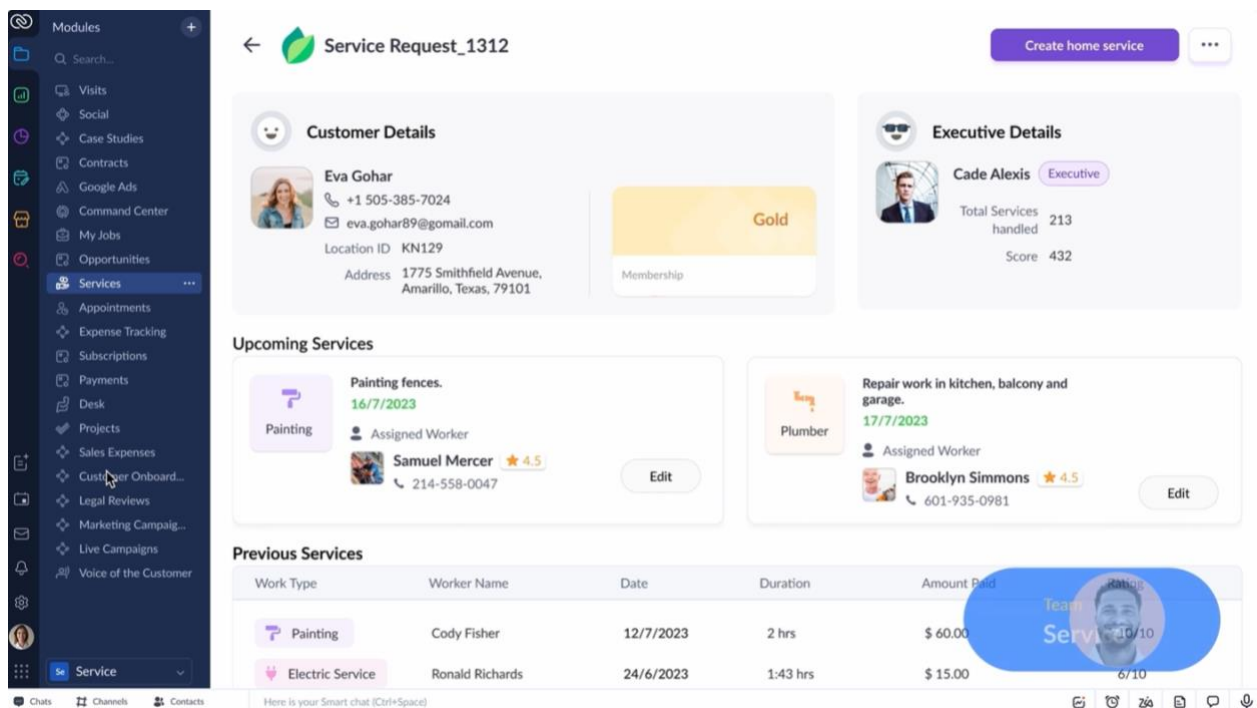


Рисунок 8 – Интерфейс Zoho CRM

4. HubSpot CRM (Рисунок 9) – одно из наиболее популярных решений на западном рынке, ориентированное на inbound-маркетинг и автоматизацию взаимодействия с клиентами. Система была представлена в 2014 году и отличается интуитивно понятным интерфейсом, наличием бесплатной версии и развитым модульным подходом. На сегодняшний день платформу используют более 100 000 компаний в различных отраслях, включая сферу образования, IT и e-commerce.

- **Ключевые функции:** управление контактами и сделками, email-маркетинг, отчётность, интеграции с CMS.
- **Ценовая политика:** бесплатный базовый пакет; дополнительные модули (Sales, Marketing, Service) — по подписке.
- **Преимущества:** интуитивный интерфейс, сильные аналитические инструменты, удобная работа с воронкой.

- **Ограничения:** важные функции (например, A/B-тесты, кастомные отчёты) только в платных модулях; ограниченная локализация для РФ.

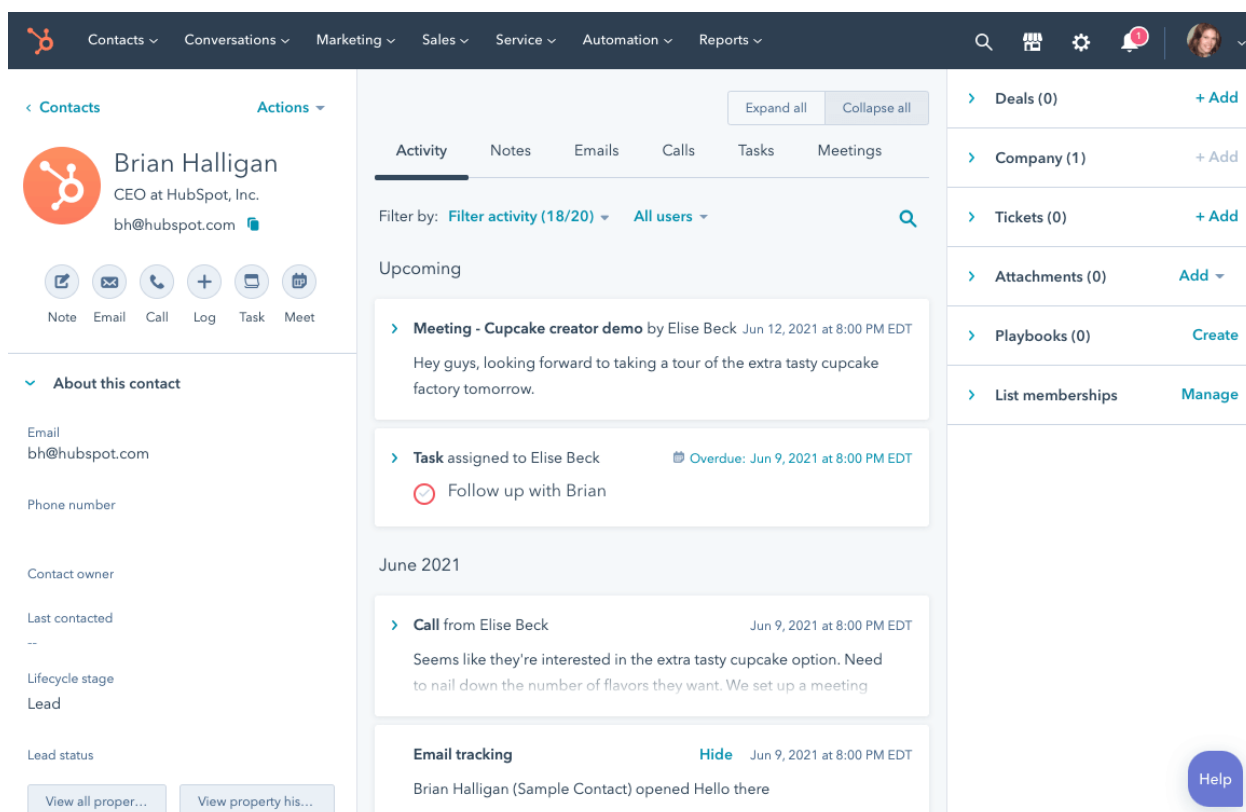


Рисунок 9 – Интерфейс HubSpot CRM

С целью обобщения представленных сведений было проведено сравнительное сопоставление существующих систем по ряду характеристик, важных для сегмента МСБ: специализация платформы, удобство внедрения, поддержка локализации, наличие бесплатных тарифов и степень интеграционной совместимости. Сводные данные представлены в таблице 1.

Таблица 1 – Сравнение популярных CRM-систем для МСБ

Критерий	АмоCRM	Bitrix24	Zoho CRM	HubSpot CRM
Основной фокус	Продажи, чат-боты	Универсальная платформа	Гибкая автоматизация	Inbound-маркетинг

Продолжение таблицы 1

Простота внедрения	Высокая	Средняя	Средняя	Высокая
Русская локализация	Полная	Полная	Частичная	Отсутствует
Бесплатный тариф	Нет	Да	Нет	Да
Поддержка интеграций	Широкая (1С, мессенджеры)	Широкая (внутри экосистемы)	Модульная	Ограниченная

1.4 Выбор и обоснование метода разработки

На основании анализа потребностей МСБ, проведенного в разделе 1.1, и обзора существующих CRM-решений, представленного в разделе 1.3, был выбран метод разработки, сочетающий современные JavaScript-фреймворки и модульную архитектуру. Выбранный подход обеспечивает гибкость, масштабируемость и соответствие ключевым требованиям МСБ: простота внедрения, низкий порог вхождения для пользователей и адаптивность к изменяющимся условиям [6].

Для реализации проекта будет применен инкрементальный подход, который позволяет последовательно реализовывать ключевые модули системы, фиксируя функциональность на каждом этапе и обеспечивая стабильность разрабатываемого решения [2]. Это особенно важно для проектов с ограниченными ресурсами, так как он дает возможность быстро проверять гипотезы и адаптировать систему под новые требования без значительных затрат [5].

Для проектирования архитектуры был применён метод Feature-Sliced Design (FSD). Это современный подход к организации фронтенд-кода, который фокусируется на бизнес-сценариях (features), а не на типах сущностей

(components, pages) [10]. FSD обеспечивает слабую связность модулей, независимость фич друг от друга, повышает читаемость, а также упрощает масштабирование проекта. Это особенно важно для CRM-систем, подверженных частым доработкам.

Ниже перечислим основные преимущества выбора FSD для разрабатываемой CRM-системы.

- 1. Масштабируемость** – новые функции добавляются как отдельные модули (features/call-integration/) без переписывания существующего кода.
- 2. Удобство поддержки**
 - Исправление бага в форме клиента (features/client-form) не затрагивает другие части системы.
 - Четкие границы модулей упрощают онбординг новых разработчиков.
- 3. Тестируемость** – изолированные фичи легко покрывать unit- и интеграционными тестами.
- 4. Гибкость** – можно заменять отдельные слои (например, перейти с Redux на Zustand) без рефакторинга всей кодовой базы.
- 5. Соответствие бизнес-логике** – структура проекта отражает реальные процессы компании (управление клиентами, сделки, отчеты), а не технические детали.

Выбор технологического стека основывался на следующих критериях:

- широкое распространение и проверенная практика использования в продакшене;
- поддержка активного сообщества и наличие официальной документации;

- возможность быстрого деплоя, простота настройки и сопровождения [21];
- лицензионная доступность (open source или бесплатные тарифы).

На основании вышеизложенных критериев были отобраны современные инструменты веб-разработки, позволяющие реализовать кроссплатформенное CRM-решение с высокой производительностью и возможностью масштабирования. Конкретный состав технологий и структура архитектуры системы будут рассмотрены в следующем разделе.

1.5 Выводы к разделу 1

В первом разделе выпускной квалификационной работы был проведен комплексный анализ рынка CRM-систем, в том числе их классификации и подходы к разработке. Все это позволило выявить ключевые проблемы и потребности МСБ. Особое внимание было уделено актуальности создания специализированного решения, которое сочетало бы в себе простоту внедрения, гибкость настройки и адаптивность к изменяющимся условиям работы компаний.

Исследование рынка показало, что существующие CRM-системы зачастую не учитывают специфику МСБ. Такие выводы были сделаны, исходя из избыточности функционала этих платформ и требования к трате значительных ресурсов на настройку и обслуживание. Это создает необходимость в разработке CRM-системы, включающей в себя высокую производительность, масштабируемую архитектуру, низкие требования к инфраструктуре и удобный пользовательский интерфейс. Все это позволит МСБ быстро адаптировать инструмент под свои бизнес-процессы без привлечения сторонних специалистов.

Научная новизна работы состоит в том, что в ходе разработки CRM-системы будут применяться современные технологии, которые будут подробно описаны в следующем разделе. Также, важным аспектом является учет

российских бизнес-реалий, включающий локализацию и соответствие требованиям по защите данных (№ 152-ФЗ "О персональных данных"). Выбранный подход не только улучшит процесс управления клиентскими отношениями, но и сократит затраты на внедрение и поддержку системы, что особенно актуально для компаний с ограниченными ресурсами.

Таким образом, проведенный в первом разделе анализ обозначил четкие направления для дальнейшей работы, включая проектирование архитектуры, выбор технологий и разработку функциональных модулей, которые в совокупности приведут к созданию востребованного и конкурентоспособного продукта на рынке CRM-систем.

Полученные результаты анализа легли в основу следующего этапа исследования – проектирования CRM-системы, где будут детально рассмотрены:

1. формулировка функциональных требований на основе выявленных потребностей МСБ;
2. обоснованный выбор технологического стека для реализации системы;
3. разработка архитектурных решений;
4. моделирование ключевых бизнес-процессов;
5. создание интуитивно понятного пользовательского интерфейса.

Переход к проектированию системы логически вытекает из проведенного анализа и позволяет перейти от теоретических исследований к практической реализации CRM-системы, учитывающей все выявленные требования и особенности МСБ.

Раздел 2. Проектирование CRM-системы

2.1 Формулировка требований к CRM-системе

На основании анализа, проведенного в первом разделе выпускной квалификационной работы, к разрабатываемой CRM-системе были сформулированы функциональные, нефункциональные, технические и UI/UX требования (Рисунок 10).

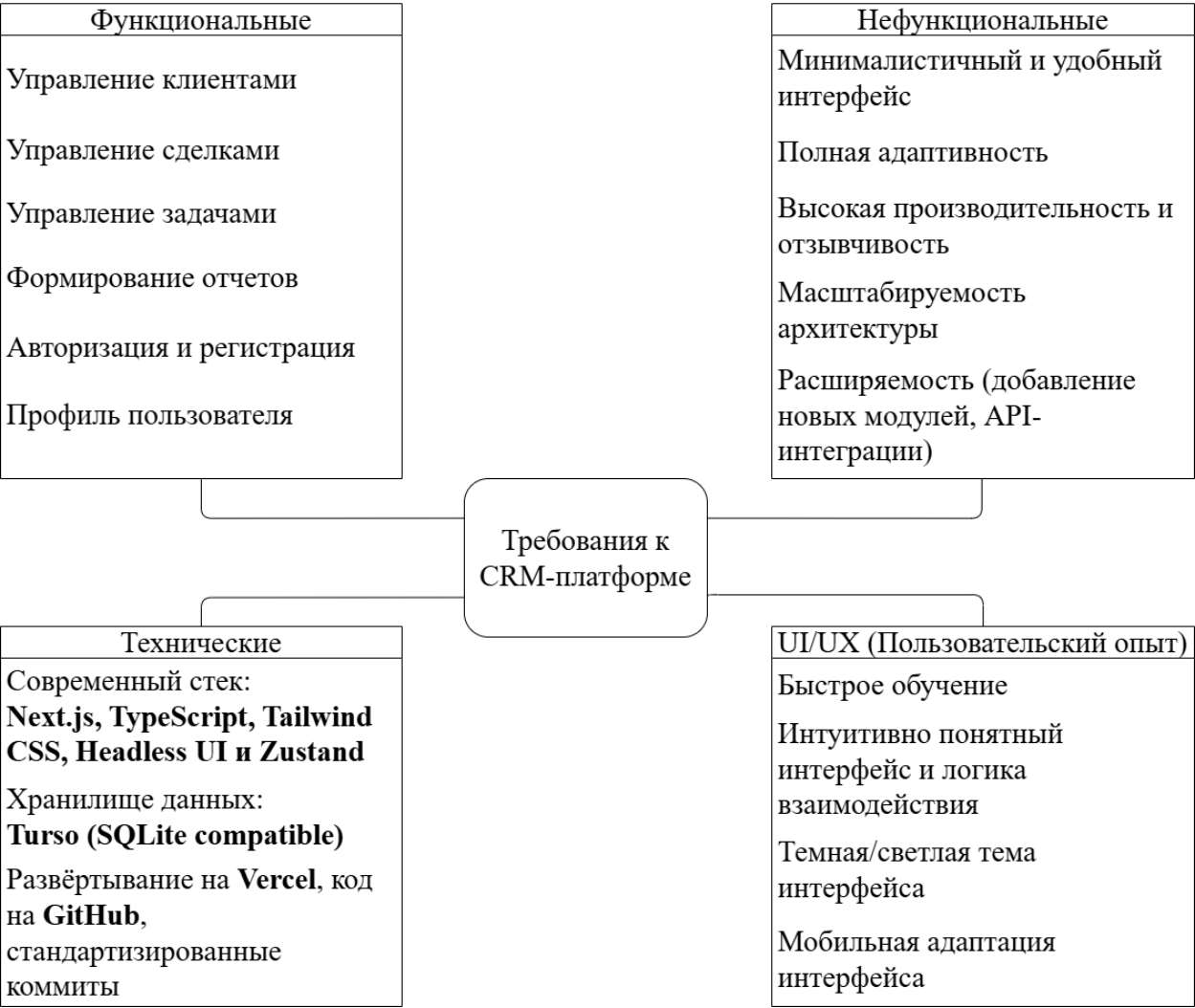


Рисунок 10 – Требования к разрабатываемой CRM-системе

Теперь подробно расскажем о каждой группе требований:

1. **Функциональные требования.** Подробное описание представлено в таблице 2.

Таблица 2 - Функциональные требования

Требование	Описание
Управление клиентами	Ведение базы клиентов, хранение контактных данных и истории взаимодействий
Управление сделками	Контроль этапов сделок, визуализация воронки продаж, назначение ответственных
Управление задачами	Создание, назначение и контроль выполнения задач, интеграция с календарем
Формирование отчетов	Генерация аналитических отчетов, визуализация ключевых показателей
Авторизация и регистрация	Система управления доступом пользователей, ролевая модель
Профиль пользователя	Персонализированный раздел с настройками аккаунта, аватаром, контактными данными и предпочтениями интерфейса

2. **Нефункциональные требования.** Их список представлен в таблице 3:

Таблица 3 - Нефункциональные требования

Требование	Описание
Минималистичный интерфейс	Простой и понятный UI без избыточных элементов
Полная адаптивность	Корректное отображение на всех типах устройств

Высокая производительность	Быстрое время отклика интерфейса
----------------------------	----------------------------------

Продолжение таблицы 3

Масштабируемость	Возможность масштабирования системы при росте нагрузки
Расширяемость	Поддержка добавления новых модулей и интеграций

3. **Технические требования.** Полный перечень данных требований представлен в таблице 4:

Таблица 4 - Технические требования

Требование	Описание
Next.js	Фреймворк для фронтенд-разработки
TypeScript	Основной язык программирования
Tailwind CSS	CSS-фреймворк для стилизации
Headless UI	Библиотека UI-компонентов
Zustand	State-менеджер для управления состоянием
Turso	SQLite-совместимая база данных
Vercel	Платформа для развертывания
GitHub	Система контроля версий

4. Требования к UI/UX. Они детализированы в таблице 5:

Таблица 5 - Требования к UI/UX

Требование	Описание
Быстрое обучение	Интуитивно понятный интерфейс, не требующий длительного обучения

Логика взаимодействия	Предсказуемое поведение системы
Темная/светлая тема	Поддержка двух цветовых схем
Мобильная адаптация	Полноценная работа на мобильных устройствах

Сформулированные требования охватывают все основные аспекты разработки CRM-системы и служат основой для последующих этапов проектирования и реализации. Особое внимание уделено балансу между функциональностью и простотой использования, что особенно важно для МСБ.

2.2 Моделирование бизнес-процессов (BPMN)

Бизнес-процесс – это последовательность взаимосвязанных действий или операций, направленных на создание продукта или услуги для конечного потребителя [4]. Для моделирования таких процессов используются различные нотации. Одной из таких является методология моделирования, анализа и реорганизации бизнес-процессов BPMN 2.0 (Business Process Model and Notation), широко используемая для описания и оптимизации бизнес-процессов в информационных системах [2].

Моделирование процессов с помощью методологии BPMN направлено на достижение таких целей, как:

- визуализация сценариев использования системы[1];
- выявление точек автоматизации;
- обеспечение согласованности требований к функциональности и интерфейсу;
- подготовка к разработке архитектуры и пользовательских сценариев.

BPMN 2.0 выбрана в качестве основной нотации благодаря своей универсальности, поддержке стандартизированных графических элементов и

способности описывать сложные процессы в доступной форме. Ниже представлены основные бизнес-процессы, которые были смоделированы для разрабатываемой CRM-системы.

1. **Управление клиентами** (Рисунок 11). Бизнес-процесс управления клиентами начинается с открытия соответствующего раздела в CRM-системе. Менеджер либо создаёт новую карточку клиента и вносит все необходимые данные, либо работает с существующим клиентом, проверяя актуальность информации и обновляя сведения.

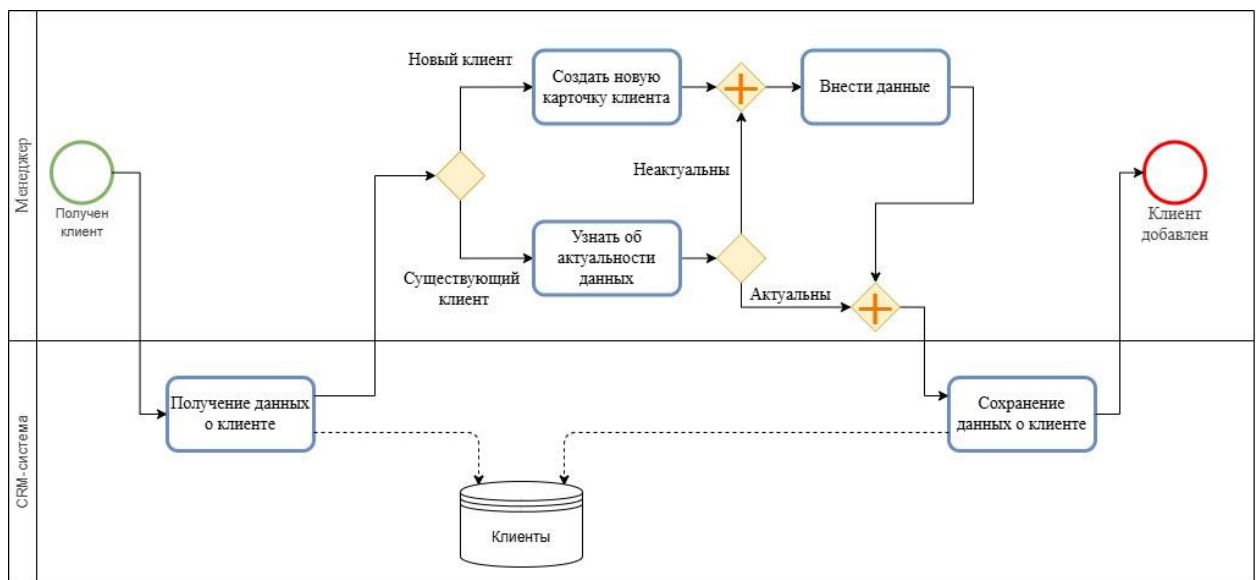


Рисунок 11 - BPMN-диаграмма бизнес-процесса “Управление клиентами”

2. **Управление сделками** (Рисунок 12). Бизнес-процесс управления сделками начинается с открытия соответствующего раздела в CRM-системе. Менеджер либо выбирает существующую сделку по клиенту, либо создаёт новую. Далее осуществляется актуализация данных: изменение статуса, взаимодействие с клиентом, добавление заметок в описание. Сделка может находиться в статусах “новая”, “в работе”, “завершена” или “отменена”. Цикл продолжается до наступления одного из финальных состояний.

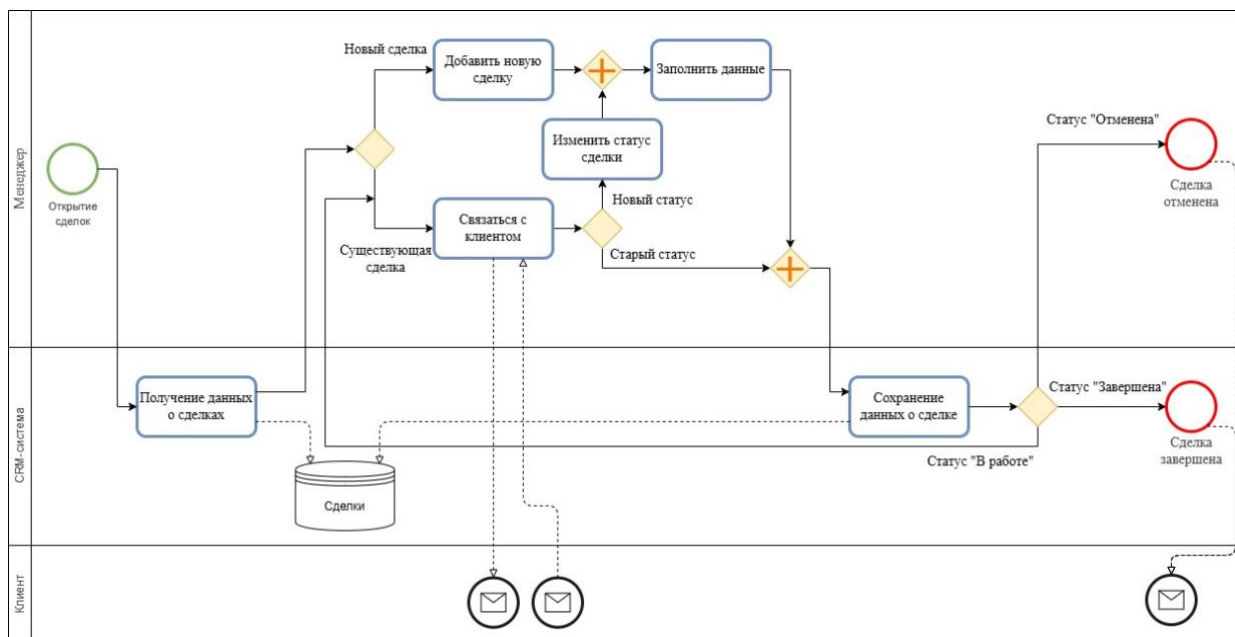


Рисунок 12 - BPMN-диаграмма бизнес-процесса “Управление сделками”

3. Постановка и выполнение задачи (Рисунок 13). Бизнес-процесс начинается с постановки задачи менеджером через интерфейс CRM-системы. Задача назначается конкретному сотруднику с указанием сроков выполнения и описания. После получения уведомления сотрудник выполняет задание и отмечает его как завершённое. Менеджер проверяет результат и, в случае соответствия, закрывает задачу. При необходимости доработки задача возвращается исполнителю.

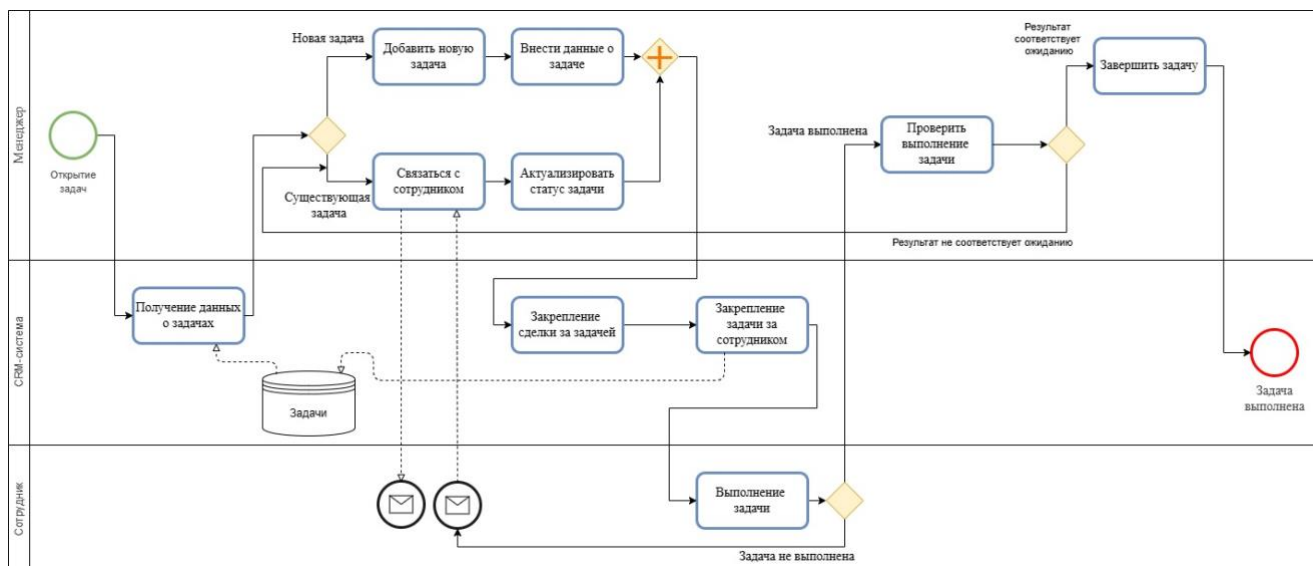


Рисунок 13 - BPMN-диаграмма бизнес-процесса “Постановка и выполнение задачи”

Помимо представленных BPMN-диаграмм, была подготовлена блок-схема основных бизнес-процессов от авторизации пользователя в CRM-системе до завершения сделки и формирования отчетов (Рисунок 14).

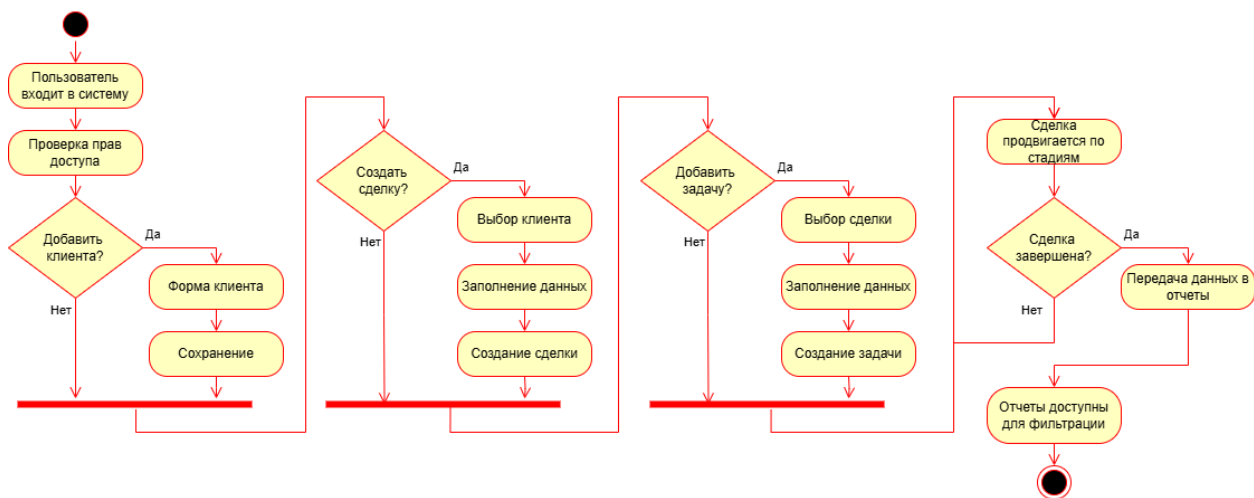


Рисунок 14 - Блок-схема основных бизнес-процессов разрабатываемой CRM-системы

2.3 Выбор технологий

Исходя из ранее сформулированных требований и смоделированных бизнес-процессов был осуществлен подбор технологического стека для разработки CRM-системы.

2.3.1 Frontend: Next.js и TypeScript

В качестве основы клиентской части CRM-системы выбран Next.js. Это фреймворк, построенный на базе React. Он предоставляет такие возможности, как серверный рендеринг (SSR), статическая генерация (SSG), автоматическая маршрутизация и встроенная поддержка API-эндпоинтов [13]. Перечисленное позволяет строить полноценные fullstack-приложения, минимизируя количество внешних зависимостей и инфраструктурных решений.

В дополнении к Next.js используется язык программирования TypeScript – строго типизированное надмножество JavaScript. Его применение для разработки CRM-системы позволит снизить количество ошибок и повысит читаемость кода, а также упростит навигацию в проекте за счет автокомплита и системы типов. TypeScript поддерживает современные практики веб-разработки и широко используется в сочетании с React-based фреймворками, такими как Next.js [10].

2.3.2 Backend: API Routes и Turso

Вместо использования отдельного backend-фреймворка реализована модель “единого проекта” – серверная логика обрабатывается через Next.js API Routes. Это упрощает структуру проекта, снижает накладные расходы на поддержку и облегчает деплой [9].

Для СУБД была выбрана Turso – облачное решение на базе SQLite/libSQL, оптимизированное под edge-развёртывания [16]. Среди её преимуществ – низкая задержка, простота настройки, глобальная репликация и минимальные требования к серверной инфраструктуре. Это делает её особенно подходящей для CRM-систем, разворачиваемых в МСБ.

Для выбора подходящей СУБД были проанализированы несколько популярных решений, применяемые в веб-разработке, таких как PostgreSQL, MongoDB и Firebase Realtime Database [22]. Результат анализа продемонстрирован в таблице 6.

Таблица 6 — Сравнение Turso с другими популярными СУБД

Критерий	Turso	PostgreSQL	MongoDB	Firebase Realtime DB
Тип СУБД	Реляционная (SQL)	Реляционная (SQL)	Документо-ориентированная	Документо-ориентированная
Поддержка ACID	Да	Да	Частично (на уровне документа)	Нет
Архитектура	Edge-first, распределённая	Централизованная (master/replica)	Шардированная, replica sets	Централизованная (облако)
Репликация	Глобальная, read-optimized	Синхронная/асинхронная	Асинхронная	Автоматическая
Масштабируемость	Чтение – горизонтально	Вертикальная/горизонтальная	Горизонтальная (через шардирование)	Горизонтальная
Поддержка Edge-развёртываний	Нативная (через Turso infra)	Ограниченно через сторонние решения	Возможна, но требует настройки	Нет

Продолжение таблицы 6

Возможность автономной работы	Да (локальная SQLite-совместимость)	Да (при self-hosting)	Частично	Нет
Необходимость DevOps-поддержки	Низкая (Zero config, SaaS)	Средняя (нужен DevOps-ресурс)	Средняя	Минимальная (SaaS)
Совместимость с ORM	Да (через SQLite-адаптеры)	Да	Частично (через ODM)	Нет
Лицензия/стоимость	Open source / Freemium	Open source (но нужен хостинг)	Open source (хостинг оплачивается отдельно)	Бесплатно (до лимитов)

Исходя из проведенного анализа, условий ограниченного DevOps-ресурса и необходимости быстрого масштабирования Turbo особенно актуальна для CRM-системы, ориентированной на МСБ.

2.3.3 Стилизация: Tailwind CSS и Headless UI

Для стилизации интерфейса CRM-системы выбран Tailwind CSS – это утилитарный CSS-фреймворк, обеспечивающий компактный и читаемый код, а также высокую производительность за счет минимизации избыточных стилей [14]. Выбранный фреймворк позволяет быстро создавать адаптивные и визуально согласованные интерфейсы, что особенно важно для МСБ, где требуется интуитивно понятный дизайн с минимальными затратами на разработку [7]. В дополнении к Tailwind CSS используется библиотека Headless UI, которая предоставляет функциональные шаблоны для создания

модальных окон, выпадающих списков, форм и других интерактивных элементов.

2.3.4 Управление состоянием: Zustand

Глобальное состояние приложения обрабатывается с помощью Zustand. Данная библиотека – это легковесная альтернатива Redux, предоставляющая удобный и лаконичный API для работы с состоянием [15]. В отличие от Redux, Zustand не требует большого объема шаблонного кода, а также хорошо интегрируется с React и Next.js, что обеспечивает высокую производительность CRM-системы.

2.3.5 Хостинг и деплой: Vercel

Проект развёртывается на платформе Vercel, тесно интегрированной с экосистемой Next.js [21]. Выбрана она за счет следующих преимуществ:

- автоматическая сборка;
- деплой по git-коммитам;
- встроенная CI/CD-цепочка;
- масштабируемость;
- поддержка edge-функций.

Данный подход снизит технические риски и позволит быстро поставлять обновления без участия DevOps-специалистов.

Выбранный технологический стек полностью соответствует поставленным требованиям и позволит реализовать все запланированные функции CRM-системы с учетом будущего масштабирования.

2.4 Архитектура CRM-системы

При проектировании CRM-системы, в рамках данной выпускной квалификационной работы, особое внимание уделяется архитектуре. Она должна быть основана на сформулированных ранее требованиях к

функциональности системы, а также на принципах, заложенных при формировании технологического стека.

Построение системы происходит по принципам монолитного SPA-приложения, реализованного с помощью фреймворка Next.js, что позволяет объединить клиентскую и серверную части в едином проекте. Такое решение упростит организацию кода и ускорит процесс разработки.

В качестве архитектурной модели выбран подход Feature-Sliced Design (FSD). FSD – это подход к структурированию frontend-приложений, при котором организация кода строится на основе бизнес-сценариев (features), а не типов сущностей (например, components или pages). Это позволяет достичь высокой модульности и слабой связанности между разными частями приложения, что особенно важно для масштабируемых информационных систем, таких как CRM [5].

Архитектура системы включает следующие ключевые разделы:

src/

— app/	# Роутинг Next.js
— components/	# UI-компоненты
— feature/	# Компоненты фич
— navigation/	# Элементы навигации
— shared/	# Общие компоненты
└ ui/	# Базовые UI-элементы
— lib/	# Вспомогательные утилиты
— services/	# Запросы к API и работа с данными, БД
— store/	# Глобальное состояние (Zustand)
└ styles/	# Глобальные стили (Tailwind CSS)

└─ utils/ # Вспомогательные функции общего назначения

FSD делит проект на несколько уровней: “app”, “processes”, “pages”, “features”, “entities” и “shared”. Такая структура обеспечивает логическое разделение обязанностей, упрощает повторное использование компонентов, поддержку и тестирование кода [23]. Разработка новых модулей и функций в рамках FSD не требует вмешательства в уже существующую функциональность, что снижает риски при доработках и обновлениях.

В рамках архитектуры, реализуемой в данной CRM-системе, каждый слой выполняет строго определённую роль:

1. App – инициализация приложения, настройка маршрутов и глобальных провайдеров;
2. Processes – реализация сквозных процессов (например, авторизация);
3. Pages – представление страниц пользовательского интерфейса;
4. Widgets – модульные механизмы собранные из Features;
5. Features – прикладная логика взаимодействия с пользователем (создание сущностей, фильтрация и т.д.);
6. Entities – бизнес-сущности: клиенты, сделки, задачи и их поведение;
7. Shared – переиспользуемые утилиты, стили и компоненты.

Для наглядности была создана комбинированная схема, отражающая архитектурные слои и взаимодействие между компонентами CRM-системы (Рисунок 15).

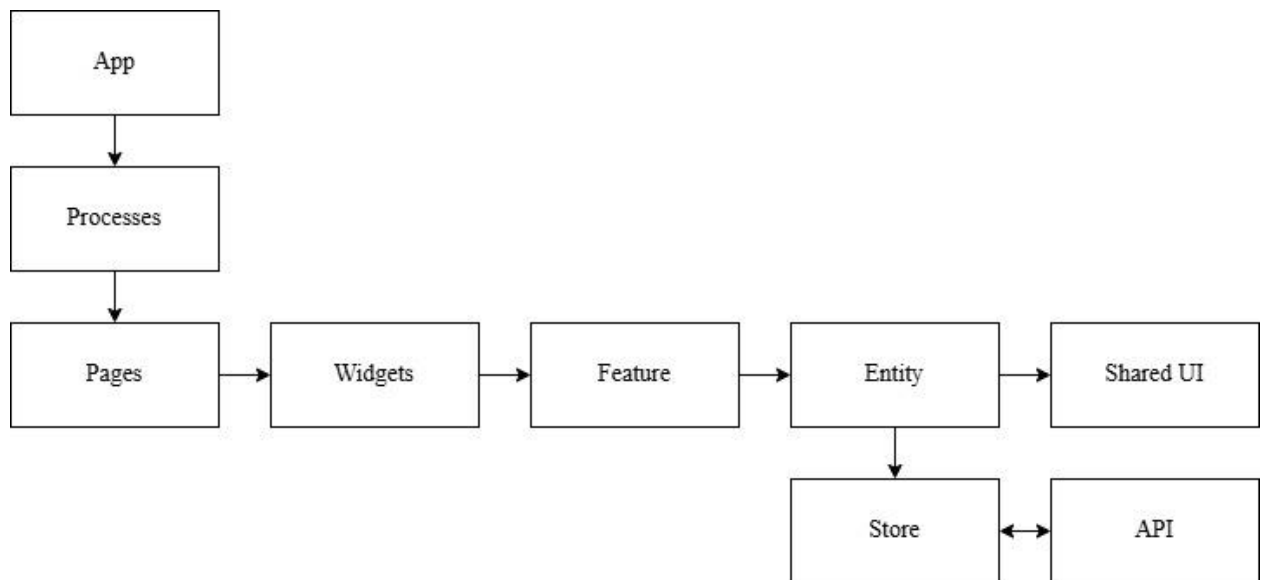


Рисунок 15 – Архитектура и поток взаимодействия компонентов CRM-системы в модели FSD

Выбранный архитектурный подход обеспечивает не только логичную и масштабируемую структуру проекта, но и позволяет эффективно организовать совместную разработку, внедрение новых функций и последующую техническую поддержку платформы.

Преимущества архитектуры FSD для CRM-системы.

- **Модульность.** Каждый слой отвечает за строго определённую функцию.
- **Масштабируемость.** Добавление новых разделов, функций и сущностей не нарушает структуру существующего кода и не требует его переработки.
- **Гибкость внедрения изменений.** Возможность локально вносить изменения в отдельные feature-модули без затрагивания остальной системы. Это особенно актуально при доработках по запросу заказчика.

- **Упрощение командной разработки.** Благодаря независимости слоёв, над разными частями системы могут одновременно работать несколько разработчиков без конфликта интересов.
- **Повторное использование компонентов.** Наличие слоя “shared” позволяет создавать универсальные UI-элементы и утилиты, которые можно повторно применить в других модулях.

2.5 UI/UX прототипирование

Чтобы обеспечить высокий уровень удобства и интуитивно понятный пользовательский интерфейс, перед разработкой CRM-системы был предварительно создан детальный UI/UX прототип. Цель этого процесса – визуализировать структуру интерфейса, определить ключевые пользовательские сценарии и утвердить логику размещения всех элементов до начала реализации.

Проектирование интерфейса осуществлялось с учётом специфики МСБ, поэтому предполагается, что будущие пользователи системы не обладают достаточной технической подготовкой и не имеют времени на длительное обучение. Отсюда появляются приоритетные принципы – простота, предсказуемость, лаконичность интерфейса, а также минимизация количества действий для достижения требуемой цели.

В качестве среды проектирования использовалась платформа Figma, которая позволяет создавать интерактивные макеты, переиспользуемые компоненты (через UI-kit), а также вести одновременную работу над интерфейсами для различных устройств [8]. Все экраны системы были спроектированы как для десктопной, так и для мобильной версии, в том числе реализованы в светлой (Рисунок 16) и тёмной (Рисунок 17) темах.



Рисунок 16 — Общий вид структуры макета в Figma для светлой темы
(Desktop и Mobile)

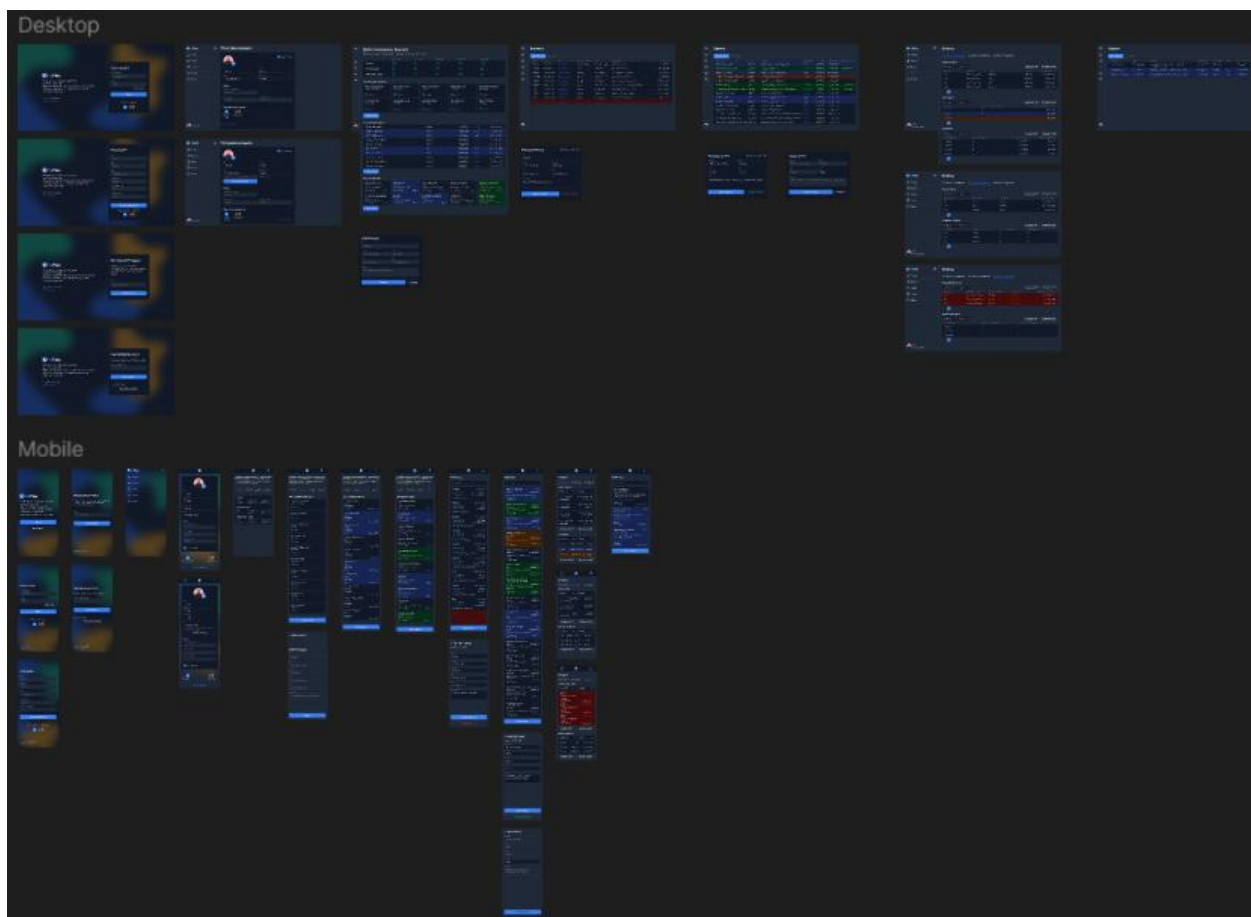


Рисунок 17 — Общий вид структуры макета в Figma для темной темы (Desktop и Mobile)

При разработке прототипа были созданы следующие разделы:

1. страница входа и регистрации;
2. главная страница (дашборд);
3. страница клиентов;
4. страница сделок;
5. страница задач;
6. страница профиля.

Для наглядности, приведен пример интерфейса страницы входа и регистрации для десктопной версии (Рисунок 17).

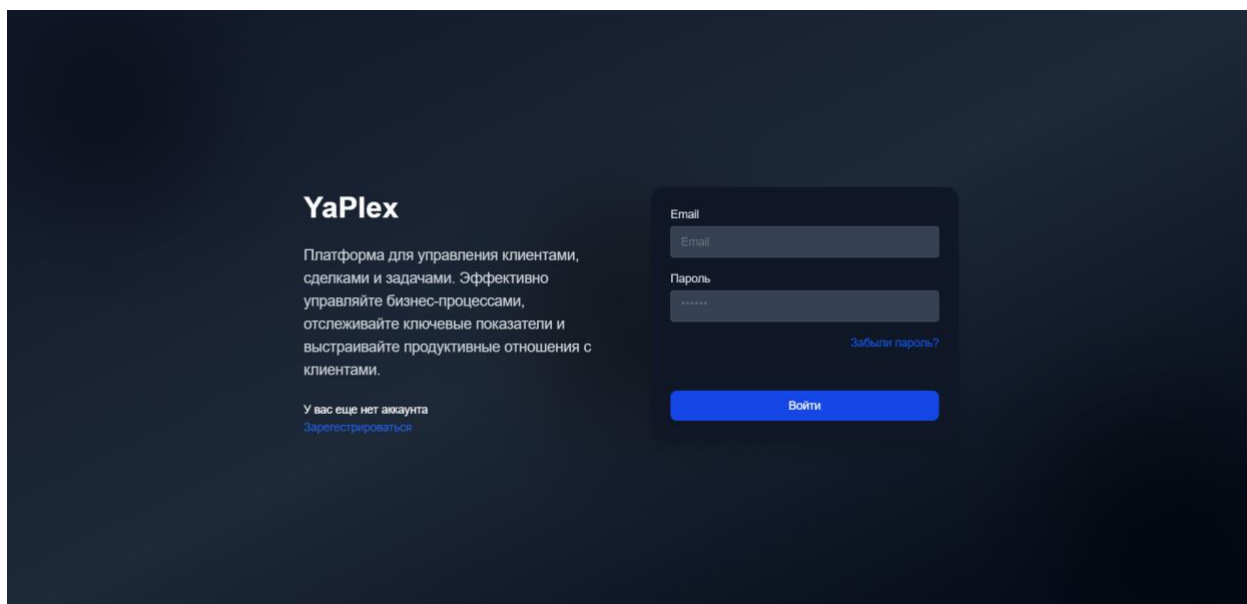


Рисунок 18 – Интерфейс страницы входа и регистрации для десктопной версии

Для мобильной версии интерфейс адаптирован с учётом ограниченного размера экрана и специфики взаимодействия с сенсорными устройствами. Вместо боковой панели навигации используется компактное выпадающее меню, открывающееся слева и предоставляющее доступ ко всем основным разделам. Ниже приведен пример интерфейса страницы входа и регистрации для мобильной версии (Рисунок 19).

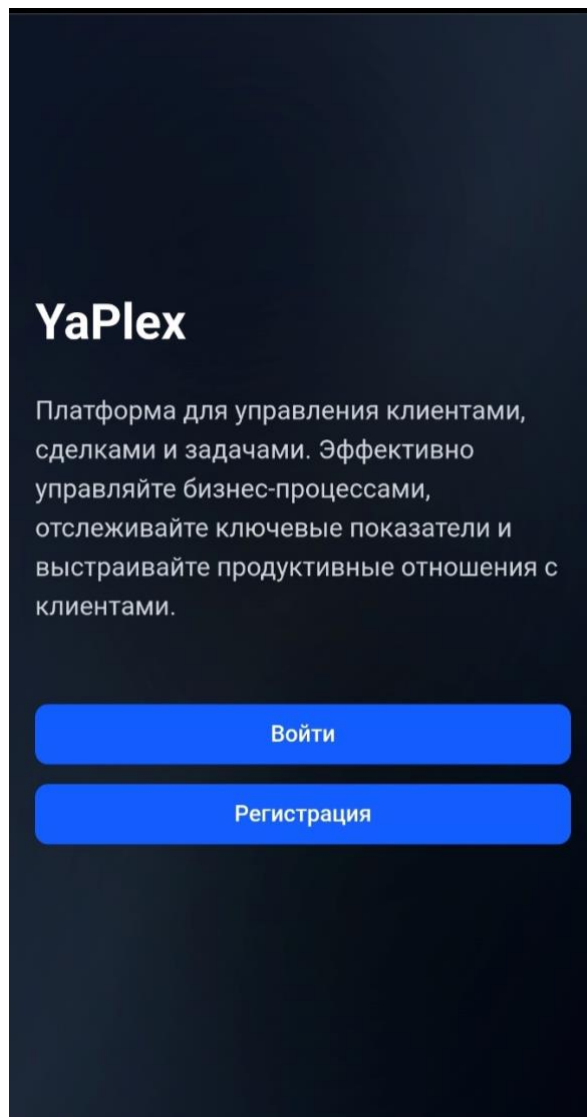


Рисунок 19 – Интерфейс страницы входа и регистрации для мобильной версии

При проектировании для повышения удобства были реализованы следующие UX-принципы:

- минимум действий до результата, все основные действия выполняются в 1-2 клика;
- единая структура экранов (заголовок, область действия, фильтры);
- интуитивная навигация между разделами;
- фокус на прикладных сценариях, а не технической реализации.

С точки зрения визуального оформления были реализованы базовые UI-принципы:

- согласованная цветовая палитра с акцентами и нейтральными оттенками;
- единая типографика и иерархия текста;
- унифицированные элементы управления (кнопки, поля, переключатели);
- поддержка как светлой, так и тёмной темы;
- соблюдение выравнивания и модульной сетки.

Библиотека компонентов интерфейса базировалась на Tailwind UI и Headless UI, что обеспечило гибкость и согласованность всех визуальных решений.

2.6 Выводы к разделу 2

Во втором разделе выпускной квалификационной работы были сформулированы функциональные, нефункциональные и технические требования к разрабатываемой CRM-системе, учитывающие особенности и потребности МСБ. Проведён обоснованный выбор технологического стека и архитектурных решений: в качестве клиент-серверного фреймворка выбран Next.js, состояние управляется через Zustand, а для хранения данных используется облачная база Turso с совместимостью SQLite.

Проектирование архитектуры системы выполнено с применением принципов Feature-Sliced Design, обеспечивающих модульность, расширяемость и удобство сопровождения кода. Помимо этого, с использованием нотации BPMN были формализованы ключевые бизнес-процессы, лежащие в основе работы CRM: управление клиентами, управление сделками и постановка задач. Это позволило определить типовые сценарии

использования системы и связать их с архитектурными и интерфейсными решениями.

В ходе проектирования был создан UI/UX-прототип, охватывающий все основные разделы системы. Интерфейс спроектирован с акцентом на простоту, предсказуемость и адаптивность. Использование Figma, Tailwind UI и принципов модульной сетки позволило обеспечить единообразие и удобство взаимодействия как в десктопной, так и в мобильной версии.

Полученные результаты позволяют перейти к этапу практической реализации, который будет рассмотрен в следующем разделе.

Раздел 3. Практическая реализация и результаты

3.1 Реализация ключевых модулей

На этапе практической реализации были созданы основные модули, обеспечивающие ключевой функционал CRM-системы: управление клиентами, обработка сделок, работа с задачами, а также система авторизации и аутентификации пользователей. Каждый из модулей представлен в виде изолированных компонентов и API-обработчиков, что соответствует архитектурной модели, описанной ранее.

Функциональность реализована в рамках единого Next.js-приложения [3]. Серверная логика организована через API Routes, клиентская часть построена на компонентной модели с использованием React и TypeScript. Для хранения состояния применяется библиотека Zustand, а взаимодействие с базой данных Turso осуществляется напрямую через SQL-запросы [20]. Такой подход позволил обеспечить модульность, прозрачность бизнес-логики и простоту масштабирования системы при необходимости.

Ниже приведено описание реализации каждого модуля, с акцентом на их функциональные особенности, структуру взаимодействия с данными и интеграцию в интерфейс пользователя.

3.1.1 Модуль управления клиентами

Модуль управления клиентами реализует базовую сущность CRM-системы и предназначен для хранения, отображения и редактирования информации о клиентах компании из МСБ. Клиентская информация включает имя, контактные данные (телефон, e-mail), компания, сайт, а также комментарий.

На уровне пользовательского интерфейса реализованы следующие возможности:

- просмотр списка клиентов в табличном представлении;

- фильтрация и поиск по имени;
- открытие детальной карточки клиента;
- добавление и редактирование клиента через модальное окно;
- удаление клиента.

Интерфейс страницы реализован в директории `src/app/clients`, где отображается список клиентов и вызываются основные действия. Форма добавления и редактирования размещена в компоненте `src/components/feature/ClientForm`, в котором также выполняется валидация пользовательского ввода. Для управления отображением и состоянием компонентов используется глобальный store на базе Zustand.

Серверная логика построена с использованием API Routes Next.js и размещена в директориях `src/app/api/clients` и `src/app/api/clients/[id]`. Реализованы следующие HTTP-методы:

- GET `/api/clients` – получение списка клиентов текущего пользователя;
- POST `/api/clients` – создание нового клиента;
- PATCH `/api/clients/[id]` – обновление существующего клиента;
- DELETE `/api/clients/[id]` – удаление клиента по идентификатору.

Каждый эндпоинт выполняет базовую валидацию данных. Он связан с авторизованным пользователем и обрабатывает возможные ошибки. Работа с базой данных Turso осуществляется через SQL-запросы, совместимые с SQLite. Пример запроса:

```
SELECT * FROM clients WHERE user_id = ? ORDER BY created_at
DESC;
```

Таким образом, модуль управления клиентами реализует полный цикл операций с основной сущностью системы и служит связующим звеном для модулей сделок и задач, с которыми клиент может быть связан.

3.1.2 Модуль работы со сделками

Модуль сделок реализует функционал управления коммерческими взаимодействиями с клиентами. Каждая сделка содержит название, информацию о клиенте, текущий статус, сумму и описание. Статус сделки может принимать одно из четырёх значений: “новая”, “в работе”, “завершена” или “отменена”, что соответствует типовой воронке продаж.

Пользовательский интерфейс модуля реализован в директории `src/app/deals`. В рамках страницы отображается список всех сделок пользователя, а также реализована возможность:

- фильтрации по статусу и названию сделки;
- открытия карточки сделки;
- добавления новой сделки;
- редактирования и удаления существующей записи.

Форма создания и редактирования сделки вызывается из модального окна. Все данные валидируются на стороне клиента перед их отправкой.

Серверная логика реализована в `src/app/api/deals` и `src/app/api/deals/[id]`. API-маршруты обрабатывают запросы:

- GET `/api/deals` – получение списка сделок;
- POST `/api/deals` – добавление новой сделки;
- PATCH `/api/deals/[id]` – обновление сделки;
- DELETE `/api/deals/[id]` – удаление сделки.

Каждая сделка связана с клиентом по `client_id`. При создании записи выполняется проверка принадлежности клиента текущему пользователю, что обеспечивает логическую целостность данных. Взаимодействие с базой Turso реализовано через прямые SQL-запросы, адаптированные под структуру таблицы `deals`.

На клиентской стороне состояние фильтров и отображаемых сделок управляется с использованием Zustand, а UI построен на базе компонентов `src/components/feature/DealForm` и `src/components/ui/Table`.

Таким образом, модуль работы со сделками обеспечивает реализацию одного из ключевых бизнес-процессов в CRM – отслеживание и управление клиентскими взаимодействиями, включая их результаты и статус завершения.

3.1.3 Модуль задач

Модуль задач реализует функционал постановки, отслеживания и управления действиями, связанными с клиентами и сделками. Каждая задача содержит название, описание, дату создания, срок выполнения, статус и привязку к конкретной сделке.

Основной пользовательский интерфейс модуля размещён в директории `src/app/tasks`. На странице отображается список задач с возможностью:

- фильтрации по статусу, исполнителю и названию;
- добавления новой задачи;
- редактирование существующей задачи;
- отметки задачи как завершённой;
- удаления задачи.

Визуально задачи представлены в виде таблицы. Форма создания и редактирования реализована через модальное окно, подключённое к глобальному хранилищу состояния.

Серверная логика модуля реализована в API-роутах `src/app/api/tasks` и `src/app/api/tasks/[id]`, где обрабатываются следующие методы:

- GET `/api/tasks` – получение всех задач пользователя;
- POST `/api/tasks` – добавление новой задачи;
- PATCH `/api/tasks/[id]` – редактирование содержимого и статуса задачи;
- DELETE `/api/tasks/[id]` – удаление задачи по идентификатору.

Все операции выполняются с привязкой к конкретному пользователю. При наличии связи с клиентом задача содержит `client_id`, обеспечивая логическое объединение сущностей. Валидация данных проводится как на клиентской стороне, так и на уровне API.

Взаимодействие с Turso осуществляется через SQL-запросы. Для повышения производительности используются выборки по `user_id` с сортировкой по дате. Пример запроса:

```
SELECT * FROM tasks WHERE user_id = ? ORDER BY due_date ASC;
```

На стороне клиента используется Zustand для хранения состояния задач, отображения активных фильтров и контроля модальных окон. Переиспользуемые компоненты формы и таблицы задач находятся в `src/components/feature/TaskForm` и `src/components/ui`.

Таким образом, модуль задач предоставляет пользователю средство управления операционной деятельностью в рамках клиентских взаимодействий, дополняя функциональность модулей клиентов и сделок.

3.1.4 Модуль авторизации/аутентификации

Модуль авторизации и аутентификации отвечает за управление доступом к системе, обеспечение безопасности пользовательских данных и поддержку сессионной модели работы. В рамках CRM-системы реализована

базовая система авторизации с использованием токенов и встроенных механизмов Next.js.

Пользовательский интерфейс модуля размещён в директории `src/app/login`, содержащей форму ввода логина и пароля. После успешного входа осуществляется автоматическая переадресация на главную страницу системы. Интерфейс выполнен в едином стиле с остальной частью платформы и поддерживает как светлую, так и тёмную темы оформления.

Серверная логика реализована в API-роутах:

- `src/app/api/user/login` — авторизация пользователя;
- `src/app/api/user/logout` — выход из системы;
- `src/app/api/user/profile` — получение информации о текущем пользователе.

Аутентификация осуществляется путём отправки учётных данных в `POST /api/user/login`, где проводится валидация, сверка с базой данных и, при успешной проверке, возвращается токен. Для хранения сессии используется `HttpOnly cookie`, что повышает безопасность и предотвращает доступ к токenu из клиентского JavaScript. Доступ к защищённым маршрутам осуществляется только при наличии валидной сессии, что контролируется через `middleware` и проверку состояния авторизации на клиенте.

На клиентской стороне используется глобальный Zustand-стор для хранения информации о текущем пользователе. После успешной аутентификации профиль пользователя запрашивается с помощью `GET /api/user/profile` и используется для отображения имени, e-mail и определения прав доступа.

Таким образом, модуль авторизации обеспечивает базовую защиту системы и корректную реализацию сессионной модели работы с CRM, соответствующую требованиям типового веб-приложения для МСБ.

3.2 Структура данных

Структура данных CRM-системы организована на основе трёх ведущих сущностей: клиенты (clients), сделки (deals) и задачи (tasks). Каждая запись в базе данных связана с конкретным пользователем, что обеспечивает безопасное разграничение доступа и изоляцию данных между аккаунтами. Хранение реализовано с использованием реляционной СУБД Turso, совместимой с SQLite, что позволяет использовать SQL-запросы и обеспечивает высокую производительность при минимальных инфраструктурных затратах [20].

Сущность клиента представляет собой основную информационную единицу. Для каждого клиента в базе данных хранятся следующие поля:

- id – уникальный идентификатор записи;
- userId – идентификатор владельца (пользователя);
- name – имя клиента;
- email – адрес электронной почты;
- phone – номер телефона;
- company – наименование компании;
- website – сайт клиента;
- comment – комментарий к карточке клиента;
- created_at – дата и время добавления карточки клиента;
- update_at – дата и время обновления карточки клиента.

Такие данные позволяют идентифицировать клиента, вести контактную базу и формировать связи с другими сущностями. Связь с пользователем обеспечивает фильтрацию клиентов в рамках конкретного аккаунта.

Сделки представляют собой бизнес-взаимодействия, связанные с клиентами. Каждая сделка описывается следующими полями:

- id – идентификатор сделки;
- userId – идентификатор владельца;
- clientId – внешний ключ на клиента;
- name – название сделки;
- description – описание сделки;
- status – статус сделки (new, in_progress, completed, cancelled);
- amount – сумма сделки (опционально);
- created_at – дата и время создания сделки;
- update_at – дата и время обновления сделки.

Данная структура позволяет отслеживать прогресс сделок, контролировать статус взаимодействия и формировать аналитику на основе этапов воронки продаж.

Задачи используются для планирования и отслеживания пользовательской активности, связанной с клиентами. Это могут быть, например, запланированные звонки, встречи, отправка коммерческих предложений или внутренние напоминания. Структура задачи включает следующие поля:

- id – уникальный идентификатор задачи;
- userId – идентификатор пользователя;
- dealId – ссылка на связанную сделку;
- name – название задачи;
- description – описание задачи;
- deadline – срок выполнения;

- `executor` – назначенный исполнитель;
- `status` – текущий статус задачи (выполняется, завершена и т. д.);
- `created_at` – дата создания записи;
- `update_at` – дата последнего изменения.

Привязка задачи к сделке через `dealId` обеспечивает контекстную связность действий, а наличие полей исполнителя и срока позволяет эффективно распределять обязанности и контролировать выполнение в рамках бизнес-процесса.

Дополнительно в системе используется сущность пользователя, включающая следующие поля:

- `userId` — уникальный идентификатор пользователя;
- `email` — адрес электронной почты, используемый как логин;
- `password` — хеш пароля;
- `firstName` и `lastName` — имя и фамилия;
- `userName` — отображаемое имя;
- `phone` — контактный номер;
- `avatar` — путь к изображению профиля;
- `userCompanyKey` — ключ компании (для группировки или разграничения доступа);
- `createdAt`, `updatedAt` — даты создания и обновления записи.

Эта таблица используется для аутентификации, персонализации интерфейса и логической привязки между пользователем и другими сущностями системы.

Таким образом, структура данных отражает логику предметной области, обеспечивает масштабируемость и логическую связность, а также формирует основу для выполнения бизнес-процессов внутри системы (Рисунок 20). При необходимости структура может быть расширена с сохранением реляционной модели хранения.

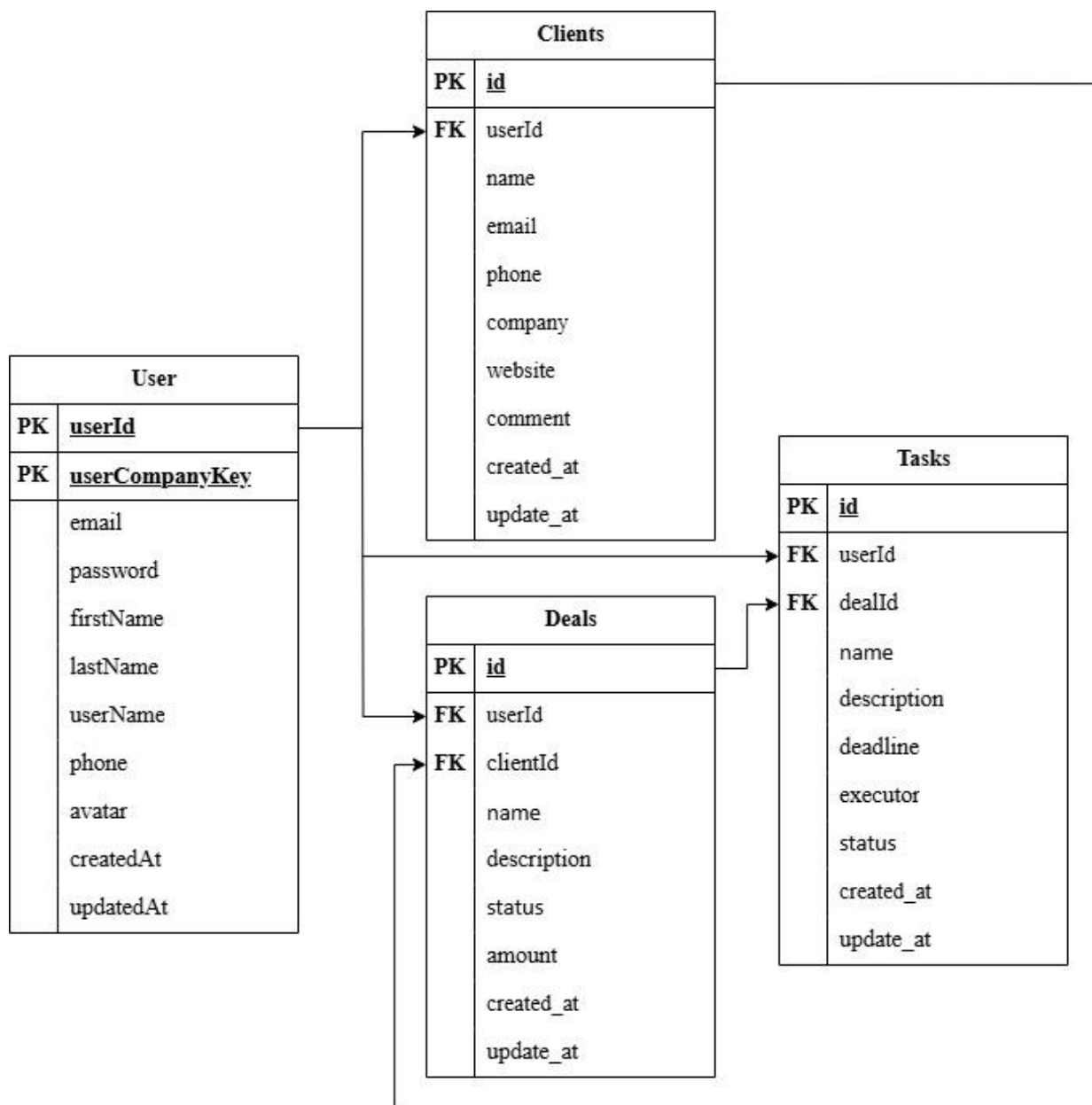


Рисунок 20 – ER-диаграмма структуры данных CRM-системы

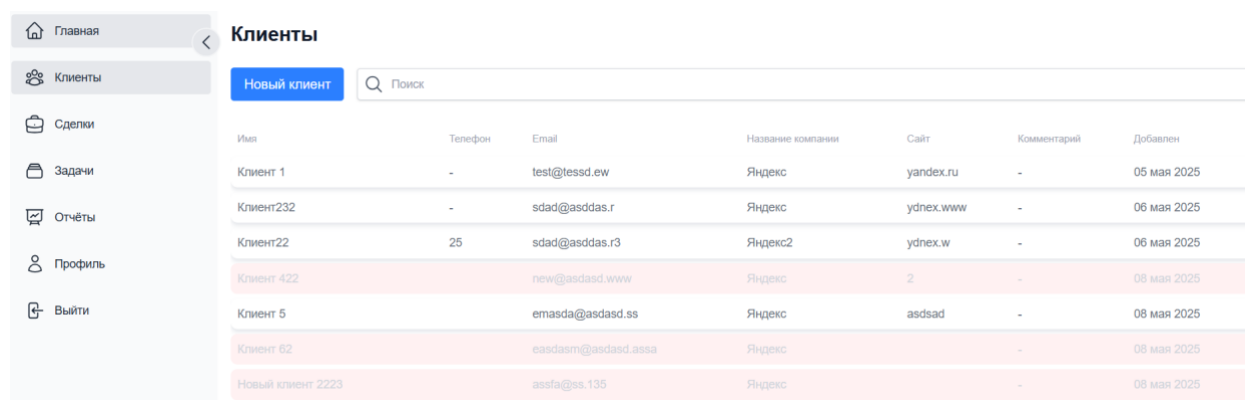
3.3 Демонстрация интерфейса

Пользовательский интерфейс CRM-системы разработан с использованием компонентной модели React и фреймворка Next.js. Все

страницы разработаны в рамках маршрутов app router, каждая из которых соответствует определённому разделу системы. В основе визуального оформления лежит Tailwind CSS с кастомными компонентами и модульной сеткой. Управление состоянием и взаимодействием между компонентами реализовано через глобальный Zustand-стор (useStore), что позволило сократить количество промежуточных пропсов и организовать динамическое обновление компонентов без необходимости перерендера всей страницы.

Главная рабочая зона – это дашборд, объединяющий ключевые сущности системы: клиентов, сделки и задачи. Навигация между разделами осуществляется через боковое меню (реализовано в компоненте Sidebar) или выпадающее меню в мобильной версии. Компоненты интерфейса распределены по директориям components/feature, components/ui и components/shared. Состояния модальных окон, фильтров и выбранных элементов контролируются через Zustand-хранилище.

На странице clients (Клиенты) представлен список всех клиентов пользователя. Отображение реализовано в виде адаптивной таблицы ClientTable, которая поддерживает фильтрацию и поиск. Добавление нового клиента происходит через модальное окно ClientForm с валидацией. Карточка клиента включает контактные данные, историю активности и связанные действия (Рисунок 21).



Клиенты						
<div>Новый клиент</div> <div>Поиск</div>						
Имя	Телефон	Email	Название компании	Сайт	Комментарий	Добавлен
Клиент 1	-	test@tesd.ew	Яндекс	yandex.ru	-	05 мая 2025
Клиент232	-	sdad@asddas.r	Яндекс	ydrex.www	-	06 мая 2025
Клиент22	25	sdad@asddas.r3	Яндекс2	ydrex.w	-	06 мая 2025
Клиент 422		new@asdasd.www	Яндекс	2	-	08 мая 2025
Клиент 5		emasda@asdasd.ss	Яндекс	asdsad	-	08 мая 2025
Клиент 62		asdasdm@asdasd.assa	Яндекс		-	08 мая 2025
Новый клиент 2223		assfa@ss.135	Яндекс		-	08 мая 2025

Рисунок 21 – Интерфейс управления клиентами

Страница deals (Сделки) отвечает за отображение и управление сделками. Каждая сделка содержит название, сумму, описание, статус и привязку к клиенту. Интерфейс поддерживает смену статусов, фильтрацию, а также добавление и редактирование записей через форму DealForm. Статусы визуально различаются по цвету и сгруппированы по этапам (Рисунок 22).

Название	Клиент	Описание	Этап (статус)	Сумма	Дата создания	Дата завершения
Сделка Попов В.А.	Клиент 1	-	Завершена	222	12 мая 2025	-
Сделка с главной страницы	Клиент22		Завершена	2123	12 мая 2025	-

Рисунок 22 – Интерфейс сделок с фильтрацией по статусам

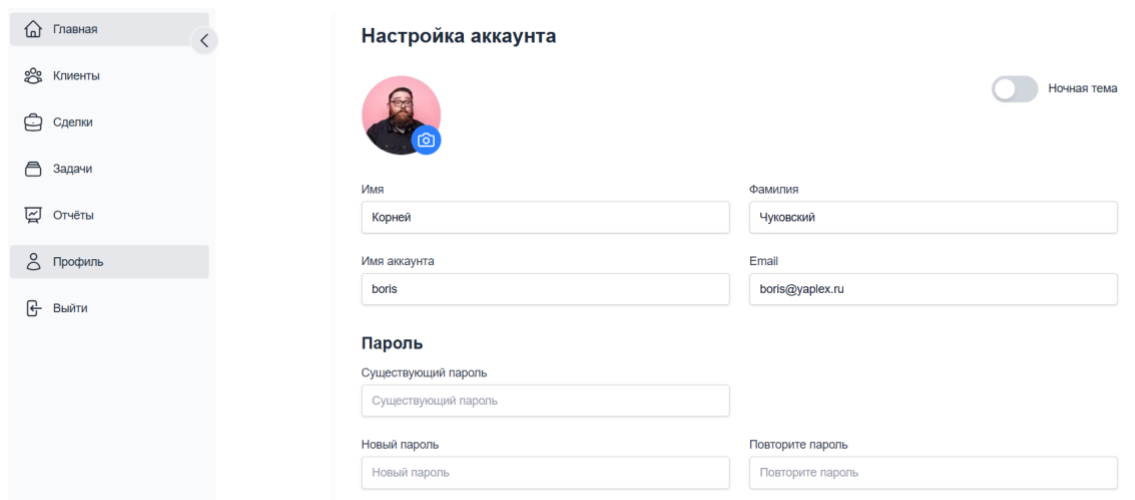
Раздел tasks (Задачи) реализует работу с задачами, связанными с конкретными сделками. Здесь представлены активные и завершённые задачи, отображаемые в табличной или карточной форме. Пользователь может редактировать задачу, менять её статус и устанавливать сроки выполнения. Добавление задачи происходит через компонент TaskForm. Список обновляется в режиме реального времени за счёт централизованного состояния (Рисунок 23).

Название	Сделка	Описание	Выполнить до	Исполнитель	Статус	Дата создания
Задача1	Сделка 2	2	02 февр. 2022	Исполнитель1	Просрочено	08 мая 2025
Задача 3-версия2	Сделка 3	Какое-то описание	10 июн. 2023	ЖЖЖ	Просрочено	08 мая 2025
ЖакиИв223	Сделка 3		17 июн. 2025	я	Завершена	10 мая 2025
Новая задача2	Сделка 3		01 янв. 2026	ы	В работе	12 мая 2025

Рисунок 23 – Интерфейс управления задачами

Также реализованы страницы login (Авторизация) и profile (Профиль). Страница входа включает форму авторизации с проверкой на стороне клиента

и передачи токена в cookie. В profile пользователь может просматривать и редактировать свои данные (Рисунок 24).



Настройка аккаунта

Ночная тема

Имя: Корней

Фамилия: Чуковский

Имя аккаунта: boris

Email: boris@yaplex.ru

Пароль

Существующий пароль: Существующий пароль

Новый пароль: Новый пароль

Повторите пароль: Повторите пароль

Рисунок 24 – Интерфейс профиля пользователя

Интерфейс полностью адаптивен. В мобильной версии реализована скрытая панель навигации, а макеты автоматически перестраиваются под ширину экрана (Рисунок 25). Также поддерживается переключение между светлой и тёмной темами. Все стили реализованы с использованием Tailwind CSS и кастомизированного UI-кита.

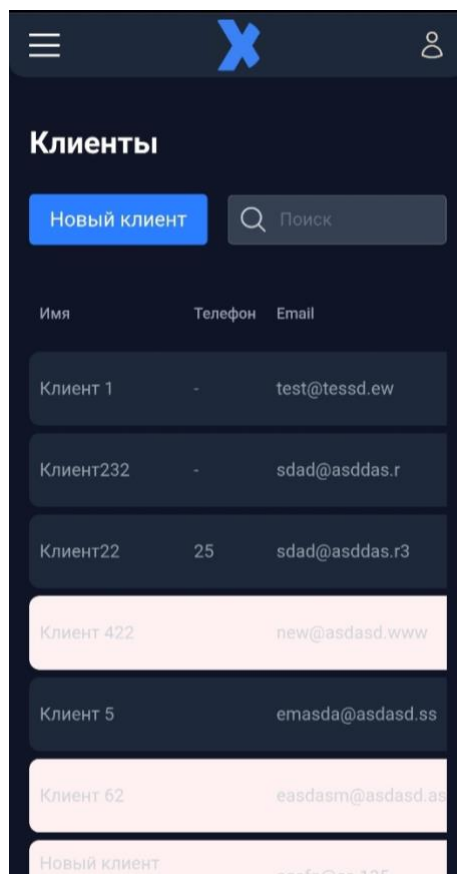


Рисунок 25 – Интерфейс CRM на мобильном устройстве

Таким образом, реализованный интерфейс покрывает весь ранее заявленный функционал, при этом оставаясь интуитивно понятным и технологически гибким. Применение модульной архитектуры компонентов, централизованного состояния и адаптивного дизайна позволяет масштабировать систему и адаптировать её под новые сценарии, требуемые в МСБ.

3.4 Количественные и качественные результаты

В результате разработки CRM-системы были достигнуты как функциональные, так и эксплуатационные цели, поставленные на этапе проектирования. Система охватывает ключевые пользовательские сценарии МСБ и демонстрирует стабильную работу как в десктопной, так и в мобильной среде.

С количественной точки зрения:

- реализовано 6 полноценных страниц интерфейса (clients, deals, tasks, login, profile, reports);
- создано более 15 переиспользуемых компонентов (формы, таблицы, карточки, модальные окна);
- интегрировано 8 API-маршрутов, обеспечивающих полный CRUD-цикл;
- используется централизованное управление состоянием с отдельными хранилищами для клиентов, сделок и задач;
- используется реляционная база данных Turso с 4 основными таблицами (users, clients, deals, tasks), оптимизированная под edge-развёртывание.

Качественные показатели также демонстрируют соответствие поставленным требованиям. Интерфейс адаптивен, полностью работает на мобильных устройствах, поддерживает тёмную и светлую темы. Все формы и действия снабжены встроенной валидацией, а структура компонентов позволяет масштабировать систему без дублирования кода.

В рамках демонстрации производительности система показала следующие характеристики:

- время загрузки страниц составляет менее 1 секунды (при хостинге на Vercel);
- действия по созданию, редактированию и удалению сущностей выполняются без перезагрузки интерфейса, за счёт использования SPA-архитектуры;
- среднее время ответа API при взаимодействии с базой данных ≤ 200 мс.

Благодаря модульной архитектуре и отказу от серверного монолита в пользу API Routes, проект может быть легко развёрнут, обновлён и масштабирован. Использование современных технологий, таких как Next.js, Turso, Zustand и Tailwind CSS, обеспечило надёжную архитектуру проекта, сократило время на реализацию ключевых функций и упростило сопровождение и развитие системы.

Таким образом, полученные результаты подтверждают реализуемость поставленных целей и демонстрируют соответствие платформы требованиям, предъявляемым к современным CRM-системам для МСБ.

3.5 Выводы к разделу 3

В третьем разделе выпускной квалификационной работы была выполнена практическая реализация разработанной CRM-системы с опорой на сформулированные ранее требования и архитектурные решения. Основной акцент был сделан на создании ключевых функциональных модулей, охватывающих управление клиентами, сделками и задачами. Каждый из модулей реализует полный цикл операций и интегрирован в единую интерфейсную структуру.

Реализованная модель хранения данных построена на основе реляционной структуры и охватывает основные сущности предметной области. Использование облачной СУБД Turso обеспечило производительность и надёжность при минимальных затратах на сопровождение и инфраструктуру. Все бизнес-сущности логически связаны и адаптированы под потребности сегмента МСБ.

Пользовательский интерфейс разработан с использованием современных технологий и принципов UI/UX-дизайна. Адаптивная верстка, тёмная и светлая темы, а также централизованное управление состоянием обеспечивают удобную и предсказуемую работу с системой. Навигация

реализована через компактные панели, а действия пользователя выполняются без перезагрузки страницы благодаря архитектуре SPA.

В ходе реализации были достигнуты положительные количественные и качественные показатели, подтверждающие эффективность выбранного технологического стека. Система показывает стабильную работу, хорошую масштабируемость и готовность к дальнейшему развитию. Полученные результаты соответствуют поставленным целям и требованиям, предъявляемым к современным CRM-решениям в контексте малого и среднего бизнеса.

Заключение

Выпускная квалификационная работа была посвящена разработке CRM-системы, ориентированной на автоматизацию бизнес-процессов в МСБ. Основной целью работы являлось создание универсального решения, позволяющего эффективно управлять клиентами, сделками и задачами, учитывая реальные потребности предприятий и недостатки существующих CRM-систем. В ходе выполнения работы были успешно решены поставленные задачи, включая анализ рынка, проектирование архитектуры системы и её практическую реализацию.

В первом разделе работы проведён комплексный анализ потребностей МСБ в CRM-автоматизации, изучены существующие решения и их классификация. Были выявлены ключевые проблемы, такие как высокая стоимость, сложность адаптации под внутренние процессы и избыточность функционала, что подтвердило актуальность разработки адаптированного под МСБ решения. На основании анализа выбран метод разработки с использованием современных JavaScript-технологий, обеспечивающих гибкость и масштабируемость системы.

Во втором разделе были сформулированы функциональные, нефункциональные, технические и UI/UX требования к системе. Разработана архитектура на основе подхода Feature-Sliced Design, обеспечивающего модульность, читаемость и удобство сопровождения. В качестве технологического стека использованы Next.js, TypeScript, Zustand, Tailwind CSS и Turso, что позволило реализовать высокопроизводительное одностраничное приложение. Также было проведено моделирование ключевых бизнес-процессов с использованием нотации BPMN и создан UI/UX-прототип, ориентированный на простоту и интуитивность взаимодействия.

Третий раздел посвящён практической реализации системы. Были разработаны основные модули: управление клиентами, работа со сделками,

задачи и авторизация. Каждый модуль реализован с полным CRUD-функционалом, валидацией данных и интеграцией в единый интерфейс. Система демонстрирует стабильную работу, высокую производительность и адаптивность как для десктопных, так и для мобильных устройств. А количественные и качественные результаты подтвердили соответствие системы изначально сформулированным требованиям.

Практическая значимость работы заключается в создании CRM-системы, которая устраняет ключевые барьеры для МСБ: простота внедрения, низкий порог вхождения и гибкость настройки. Разработанное решение может быть использовано малыми и средними предприятиями для оптимизации управления клиентскими отношениями и внутренними процессами, что способствует повышению их конкурентоспособности.

В рамках дальнейшего развития системы возможна реализация дополнительных функций: интеграция с внешними сервисами (например, почтовыми API, мессенджерами и платёжными платформами), создание модуля аналитики с визуализацией ключевых показателей, добавление поддержки ролей и прав доступа, а также расширение возможностей отчётности. Кроме того, перспективным направлением может стать внедрение механизмов машинного обучения для анализа клиентской активности и прогнозирования сделок.

Исходя из предоставленной выше информации, цели и задачи выпускной квалификационной работы достигнуты. Полученные результаты могут быть применены в реальной бизнес-среде, а также послужить основой для дальнейшего развития системы, например, за счёт добавления новых модулей или интеграции с внешними сервисами.

Список используемой литературы

1. Барабанов П.А., Захаров А.И. Разработка информационных систем. – М.: Форум, 2020 – 272 с.
2. Лаптев В.В. Основы проектирования информационных систем: Учебник. – М.: КНОРУС, 2021 – 320 с.
3. Соловьев В.Д. Web-программирование: учебник. – СПб.: Питер, 2023 – 416 с.
4. Макаров В.А. CRM-системы и их применение в бизнесе. – М.: Юрайт, 2020 – 256 с.
5. Бредихин А.А. Архитектура программных систем. – М.: Директ-Медиа, 2022 – 198 с.
6. Джонсон Д., Синглтон М. JavaScript и современный веб: от основ к фреймворкам. – СПб.: БХВ-Петербург, 2021 – 334 с.
7. Богданов В.И. Перспективы цифровизации МСП в России // Бизнес-информ. – 2021 – № 6 – С. 87-91.
8. Ильин А.Н. Тренды в разработке веб-приложений // Программные продукты и системы. – 2023 – № 4 – С. 22-28.
9. Абдуллин Р.А. Применение современных фреймворков при создании CRM-систем // Информационные технологии и вычислительные системы. – 2022 – № 3 – С. 15–20.
10. Шумаков А.В. Эволюция JavaScript-фреймворков // Современные технологии. Системный анализ. Моделирование. – 2023 – № 2 – С. 45-49.
11. Сеницын Д.П. Цифровые платформы в управлении бизнесом // Управление в экономических системах. – 2020 – № 3 – С. 33-39.

- 12.Сбер Бизнес Софт; TenChat. Использование CRM-систем в России 2024. [Электронный ресурс]. – Режим доступа: <https://sbercrm.com/blog/issledovanie-sber-i-tenchat> (дата обращения: 10.04.2025).
- 13.Next.js Documentation. [Электронный ресурс]. – Режим доступа: <https://nextjs.org/docs> (дата обращения: 02.04.2025).
- 14.Tailwind CSS Documentation. [Электронный ресурс]. – Режим доступа: <https://tailwindcss.com/docs> (дата обращения: 02.04.2025).
- 15.Zustand – Bear necessities for state management. [Электронный ресурс]. – Режим доступа: <https://zustand-demo.pmnd.rs/> (дата обращения: 04.04.2025).
- 16.Turso – Edge Database by ChiselStrike. [Электронный ресурс]. – Режим доступа: <https://turso.tech/> (дата обращения: 02.04.2025).
- 17.Bitrix24. Обзор CRM. [Электронный ресурс]. – Режим доступа: <https://www.bitrix24.ru> (дата обращения: 04.04.2025).
- 18.AmoCRM. Возможности CRM-системы. [Электронный ресурс]. – Режим доступа: <https://www.amocrm.ru/> (дата обращения: 05.04.2025).
- 19.HubSpot CRM Features. [Электронный ресурс]. – Режим доступа: <https://www.hubspot.com/products/crm> (дата обращения: 06.04.2025).
- 20.SQLite Documentation. [Электронный ресурс]. – Режим доступа: <https://www.sqlite.org/docs.html> (дата обращения: 07.04.2025).
- 21.Vercel Docs. [Электронный ресурс]. – Режим доступа: <https://vercel.com/docs> (дата обращения: 07.04.2025).
- 22.IONOS Digitalguide. What is a Data Warehouse? [Электронный ресурс]. – Режим доступа: <https://www.ionos.co.uk/digitalguide/online->

- marketing/web-analytics/dwh-what-is-a-data-warehouse/ (дата обращения: 08.04.2025).
23. Паттерны проектирования в современных CRM // Journal of Software Engineering [Электронный ресурс]. – URL: <https://www.jser.org/archive/2021/crm-design-patterns> (дата обращения: 10.04.2025).
24. Джеентаева К. М., Зимина В. С. Анализ применения CRM-систем на предприятиях // Молодой ученый. – 2020. – № 11 (301). – С. 237–239. – URL: <https://moluch.ru/archive/301/68089/> (дата обращения: 03.04.2025).