

## **1. Анализ предметной области**

### **1.1 Сведения об организации**

**«ФармФирма «Сотекс»** — современный производитель лекарственных средств. В продуктовом портфеле компании более 150 позиций востребованных, завоевавших доверие специалистов и потребителей препаратов, применяющихся в социально значимых терапевтических направлениях: неврология, онкология, ревматология, нефрология, психоневрология и др. Современные и эффективные препараты компании «Сотекс» не уступают западным аналогам по качеству и доступнее экономически.

В компании находится множество отделов, благодаря которым она способна функционировать и ежедневно производить продукцию. По всему производственному корпусу установлено большое количество камер видеонаблюдения. В одном помещении может быть установлено сразу несколько таких камер, что позволяет записывать видеоматериал с нескольких ракурсов. Отдел диспетчеризации целиком и полностью влияет на качество и контроль за всем производственным и трудовым процессом, именно поэтому, в нём расположено столь же крупное количество мониторов, на которых в реальном времени отображается запись с этих камер видеонаблюдения, дарующих возможность наблюдать за большинством процессов в компании.

Так же, в приведенном выше отделе диспетчеризации используется SCADA-система которая способна выполнять большое количество задач по контролю за всеми этапами производства продукции и играет важнейшую роль в её формировании. Эта система представляет собой программный пакет предназначенный для сбора, обработки, отображения и архивирования информации о процессе производства продукции.

Для удобного контроля и пользования приведенной системы, потребовалось расположить столь же массивное скопление мониторов, как и в случае с мониторами, на которых отображается видеонаблюдение. Каждый монитор отображает собственный сектор и показатели микроклимата.

Например, один из них показывает какая температура у каждого холодильника, хранящего препараты и медикаменты в нужном секторе в реальном времени, или какое на данный момент содержание частиц в приготовленном для осуществления производства продукции воздухе. В случае отклонения от заданных показаний, установленных в документальном порядке, диспетчеру отображается визуальное и довольно громкое звуковое сопровождение о возникшей проблеме. После уведомления о неполадке диспетчер связывается с держателем процесса — человеком, который ответственен за руководство и контроль сектора, в котором возникло отклонение, после этого вписывает в журнал все данные об отклонении. В случае, если проблему не удаётся решить первоначальными способами, то диспетчер при помощи SCADA-системы имеет возможность внести некоторые корректировки для стабилизации процесса и устранения отклонения в производственных помещениях.

Также при каждом начале работы одного из этапов производства, держатели процесса связываются с диспетчером для получения разрешения на проведение этапа производства и уточняют, какие показатели микроклимата в этом помещении на данный момент.

Для связи с диспетчером может использоваться корпоративный телефон, а также рация, способная передавать связь по нескольким каналам и давать возможность общаться сразу нескольким людям в любой момент времени.

## **1.2 Классификация информационных систем**

Информационные системы (ИС) играют важную роль в современном мире, обеспечивая поддержку бизнес-процессов, управления данными, коммуникации и принятия решений. Важность ИС требует тщательного изучения их классификации, чтобы понять разнообразие и назначение различных систем. В этой статье рассматриваются основные подходы к классификации информационных систем, включая типы информационных систем, функциональные области, архитектуры, уровни интеграции, модели распространения и модели лицензирования.

### **1) Классификация по функциональному пространству**

#### **Оперативная информационная система:**

**Транзакционная система (TPS):** обеспечивает выполнение рутинных операций, таких как обработка заказов, бухгалтерский учет и управление запасами. Примеры: система обработки продаж, банковская система.

**Система управления производственными процессами (MES):** предназначена для управления и мониторинга производственных операций. Пример: Система управления производственной линией.

#### **Информационная система управления (MIS):**

Они обеспечивают промежуточную управленческую поддержку, предоставляя отчетную и аналитическую информацию для принятия решений. Примеры: система финансового управления, система управления персоналом.

#### **Система поддержки принятия решений (DSS):**

Они используют модели данных, аналитические инструменты и методы для принятия решений. Примеры: система управления проектами, система финансового планирования.

#### **Информационная система администратора (EIS):**

Они сосредоточены на стратегическом планировании и управлении и предоставляют информацию высокого уровня для руководителей высшего звена и менеджеров. Примеры: информационные панели, системы бизнес-аналитики.

### **Система управления взаимоотношениями с клиентами (CRM):**

Именно эта система была выбрана мной для разработки. Она предназначен для управления взаимодействием с клиентами, включая продажи, маркетинг и услуги. Например: Salesforce, HubSpot.

### **Система управления ресурсами предприятия (ERP):**

Интегрируйте ключевые бизнес-процессы, такие как финансы, производство, продажи и человеческие ресурсы, в единую систему. Примеры: SAP, Oracle ERP.

### **Система управления цепочками поставок (SCM):**

Они обеспечивают управление цепочками поставок и логистикой от планирования до доставки. Пример: Jda, Manhattan Partners.

### **Информационно-управляющая система (КМС):**

Они обеспечивают создание, распространение и использование знаний внутри организации. Примеры: SharePoint, слияние.

## **2) Классификация по архитектуре**

### **централизованный:**

Процессы обработки данных и управления сосредоточены в одном центральном месте. Пример: Центральная банковская система.

развертывается.:

Данные и операции распределяются по нескольким узлам, обеспечивая отказоустойчивость и масштабируемость. Примеры: система управления филиалами, глобальная логистическая система.

### **Клиент-серверные ИС:**

Система разделена на клиентскую и серверную части, сервер обеспечивает обработку данных, а клиент предоставляет пользовательский интерфейс. Пример: Корпоративная база данных, клиент-серверное приложение.

### **Облако (SaaS):**

Доступ к системе осуществляется через Интернет, а данные хранятся в облаке. Примеры: рабочая область Google, Microsoft Office 365.

### **3) Классификация по уровню интеграции и интегрированный:**

Они объединяют различные компоненты и процессы в единую систему, чтобы обеспечить согласованность данных и процессов. Пример: ERP-система.

#### **Модульные ИС:**

Они состоят из отдельных модулей, которые могут работать независимо или интегрированно. Пример: CRM-система с отдельным модулем для продаж, маркетинга и обслуживания.

#### **Сервисные системы (SOA):**

Они основаны на концепции сервис-ориентированной архитектуры, в которой компоненты взаимодействуют через стандартные интерфейсы и протоколы. Примеры: веб-сервисы, микросервисы.

### **4) Классификация по модели распределения**

#### **Внутри компании:**

Они устанавливаются и запускаются на серверах компании. Пример: традиционная ERP-система.

#### **Облако (SaaS):**

Оно доступно в Интернете и предлагается по подписке. Примеры: Salesforce, рабочая область Google. Гибрид:

Оно сочетает в себе элементы локальных и облачных решений, обеспечивая гибкость и возможность выбора наилучшей модели. Пример: Гибридные облачные решения.

### **5) Классификация по типу лицензии**

#### **Зарегистрированный:**

Они разработаны коммерческой компанией и продаются по лицензии. Примеры: Microsoft Dynamics, Oracle ERP.

#### **С открытым исходным кодом:**

Исходный код доступен для модификации и использования. Примеры: Odoo, SugarCRM.

## **6) Классификация по размеру и типу бизнеса**

### **Для малого и среднего бизнеса:**

Разворачивают систему, которая проста в использовании и имеет базовые функции. Примеры: Zoho CRM, QuickBooks.

### **Для средних компаний:**

Система с расширенными функциями и параметрами настройки. Примеры: Microsoft Dynamics365, NetSuite.

### **Для крупных компаний:**

Полнофункциональная система с глубокими возможностями настройки и интеграции. Примеры: SAP, Oracle ERP.

## **7) Классификация по отраслям**

### **Отраслевые ИС:**

Индивидуальные решения, адаптированные к конкретным отраслям. Пример: Система здравоохранения, розничная система.

### 1.3 Обзор аналогов информационных систем

CRM-системы широко используются компаниями в различных отраслях промышленности для управления взаимоотношениями с клиентами и повышения эффективности продаж. Вот несколько популярных российских CRM-систем:

#### 1) Битрикс24:

**Bitrix24** - популярная CRM-система, предоставляющая широкий спектр инструментов для управления взаимоотношениями с клиентами, коммуникаций и организаций бизнес-процессов. Вот некоторые из основных функций и возможностей bitrix24crm:

**Управление базами данных клиентов:** Bitrix24 позволяет хранить информацию о клиентах, контактах, транзакциях, компаниях и других организациях в одной базе данных. Легко отслеживайте контакты с клиентами, историю взаимодействия и другие данные.

**Управление торговлей и продажами:** Bitrix24CRM предоставляет инструменты для создания и отслеживания транзакций, управления этапами продаж, прогнозирования выручки и анализа эффективности продаж.

**Маркетинг и коммуникации:** Bitrix24 имеет такие функции, как создание маркетинговых кампаний, рассылка информационных бюллетеней по электронной почте, управление социальными сетями и интеграция с различными каналами связи.

**Задачи и проекты:** Bitrix24 включает инструменты для управления задачами, проектами и совместной работой в команде. Вы можете создавать задачи, делегировать задачи, отслеживать прогресс и оценивать результаты.

**Отчетность и анализ:** Эта система предоставляет возможность создавать отчеты о продажах, маркетинговых кампаниях, деятельности клиентов и других показателях эффективности.

**Интеграция:** Bitrix24 имеет множество интеграций с другими службами и приложениями, что упрощает работу с существующими инструментами в вашем бизнесе.

**Bitrix24** доступен в виде облачного решения и автономной установки на серверах компании. Платформа предлагает различные тарифные планы с различными характеристиками в зависимости от потребностей бизнеса.

**Bitrix24** - это мощная CRM-система с множеством полезных функций, но у нее также есть некоторые **недостатки**, поэтому **она не была выбрана мной:**

**Сложность использования:** Для некоторых пользователей запуск Bitrix24 может быть затруднен из-за его многочисленных функций и настроек. Вам нужно потратить время на изучение и освоение платформы.

**Стоимость:** Некоторые компании могут посчитать, что цена тарифного плана Bitrix24 слишком высока, особенно если вам не нужны все предлагаемые функции. Для малого бизнеса это может быть не самый экономичный вариант.

**Потребность в технической поддержке:** Установка и настройка автономной версии Bitrix24 на сервере требует определенных технических навыков. Это может быть непросто для компаний, в которых нет ИТ-специалистов.

**Интеграция с другими системами:** Несмотря на доступность интеграции, некоторые пользователи могут столкнуться с ограничениями при интеграции Bitrix24 с другими внутренними или сторонними системами.

**Ограничения бесплатной версии:** Бесплатная версия Bitrix24 имеет ограничения на количество пользователей, доступные функции и объем хранимых данных. Для крупных компаний этого может быть недостаточно.



**Зависимости от подключения к Интернету:** Bitrix24 - это облачный сервис, поэтому для его использования вам необходимо определенное подключение к Интернету. В случае проблем с Интернетом это может быть проблемой.

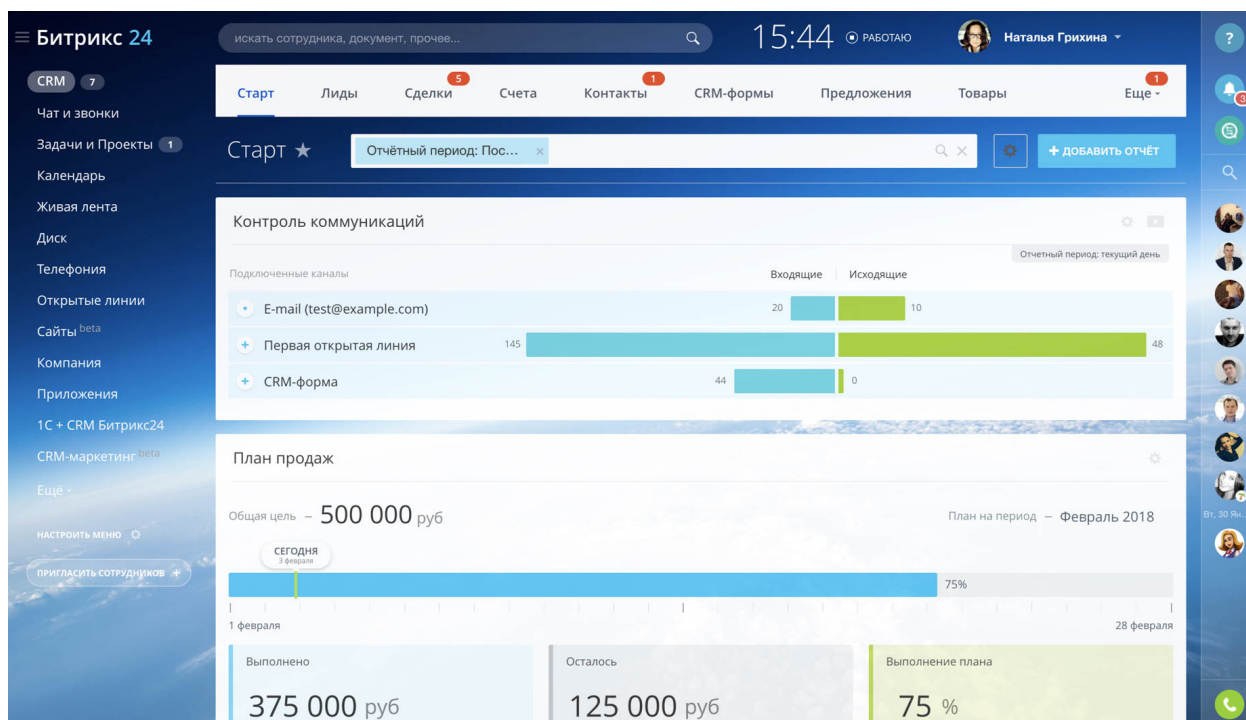


Рисунок 1.1 – Интерфейс Bitrix24

## 2) AmoCRM:

**AmoCRM** - популярная CRM-система, разработанная российской компанией amoCRM. Он предназначен для автоматизации управления продажами, взаимодействия с клиентами и улучшения работы отдела продаж. Некоторые важные особенности российской CRM-системы Amocrm включают в себя::

**Удобный интерфейс:** amoCRM предлагает интуитивно понятный и простой в использовании интерфейс, который делает систему более эффективной в использовании.

**Управление воронкой продаж:** эта система позволяет создавать и настраивать воронки продаж, отслеживать этапы транзакций и контролировать движение по воронке.

**Возможности автоматизации:** amoCRM предлагает широкий спектр функций автоматизации процессов, включая отправку уведомлений, создание задач и работу с шаблонами.

**Интеграция:** платформа поддерживает интеграцию с различными службами и приложениями, такими как почтовые службы, социальные сети, телефония и другие инструменты.

**Отчетность и анализ:** amoCRM предоставляет возможность создавать отчеты о продажах, анализировать эффективность отделов продаж и отслеживать ключевые показатели.

**Мобильное приложение:** amoCRM предлагает мобильное приложение, которое позволяет запускать систему из любого места для удобства пользователей

**Безопасность данных:** компания обеспечивает высокий уровень защиты пользовательских данных, делая использование системы безопасным для бизнеса

**amoCRM** является популярным выбором для многих компаний как в России, так и за рубежом благодаря своей функциональности, простоте использования и гибкости настройки в соответствии с различными потребностями бизнеса.

**amoCRM** - это мощная CRM-система, **но у нее также есть некоторые недостатки**

**Сложность конфигурации:** Первоначальная установка Amosm может быть трудоемкой и трудоемкой, особенно если вам необходимо создавать сложные рабочие процессы или интегрироваться с другими системами.

**Стоимость:** Для некоторых компаний стоимость использования amoCRM может быть очень высокой, особенно если вам нужны расширенные функции или большое количество пользователей.

**Отсутствие определенных функций:** Определенные отрасли октября и определенные бизнес-процессы могут не включать все функции, необходимые для amoCRM, и могут потребовать дополнительной настройки или интеграции.

**Интеграция с другими системами:** некоторым компаниям могут потребоваться октябрьские дополнительные усилия или ресурсы для интеграции amoCRM со своими существующими системами.

**Ограничение разрешений пользователей:** Обратите внимание, что некоторые пользователи могут быть немного ограничены в настройке разрешений пользователей в amoCRM по сравнению с другими CRM-системами.

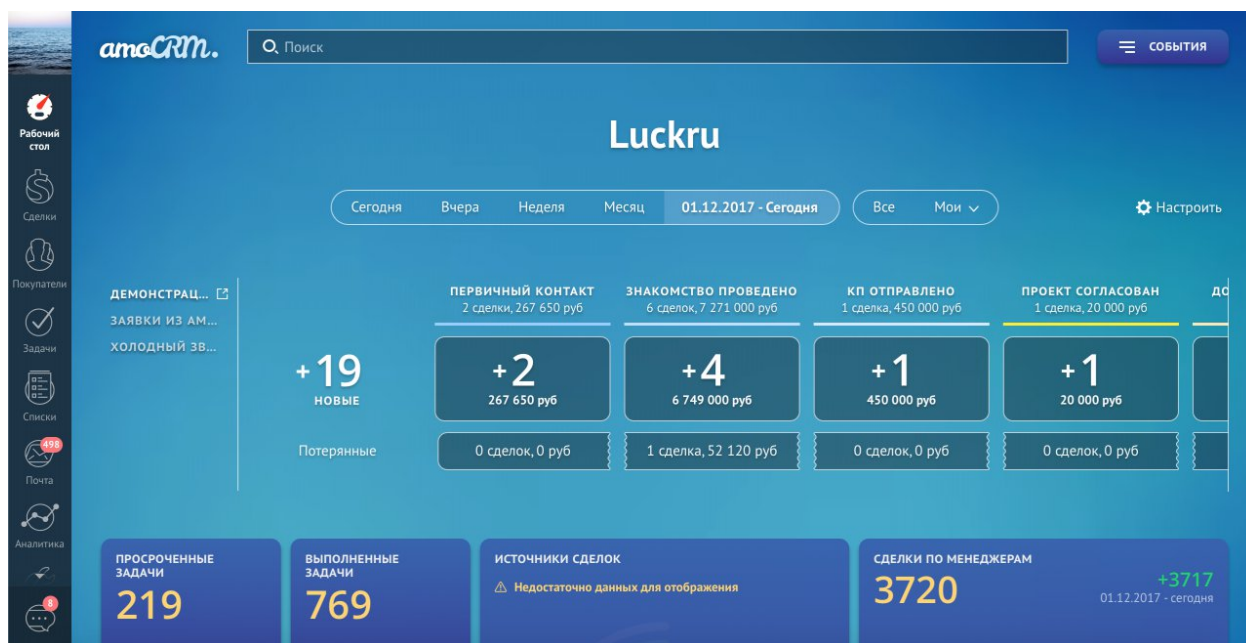


Рисунок 1.2 – Интерфейс AmoCRM

### 3) RetailCRM:

**RetailCRM** - это облачная CRM-система, разработанная специально для розничной торговли. Он предлагает комплексное решение для управления

взаимоотношениями с клиентами, заказами, хранением, маркетингом и аналитикой для розничных компаний.

Вот некоторые ключевые особенности и преимущества **retailcrm::**

**Многоканальный:** с помощью этой платформы вы можете интегрировать несколько каналов продаж, таких как интернет-магазины, розничные магазины и колл-центры, в одну систему, чтобы удобно управлять всеми заказами и данными о клиентах.

**Управление заказами:** retailCRM предоставляет инструменты для управления заказами с момента размещения до доставки и возврата. Это помогает оптимизировать процесс обработки заказов и повысить удовлетворенность клиентов.

**Интеграция с другими сервисами:** эта платформа может быть легко интегрирована с различными сервисами и платформами, такими как платежные системы, логистические компании и службы доставки.

**Аналитика и отчетность:** retailCRM предоставляет обширные возможности для анализа данных, отчетности и прогнозирования продаж, чтобы помочь вам принимать обоснованные бизнес-решения.

**Персонализация общения с клиентами:** эта система позволяет создавать персонализированные маркетинговые кампании с клиентами на основе предпочтений и истории покупок.

**Складской учет:** retailCRM включает инструменты для учета запасов, управления поставками и оптимизации запасов.

**retailCRM** широко используется розничными компаниями для улучшения процесса продаж, повышения эффективности и улучшения обслуживания клиентов.

**Retailcrm** имеет много преимуществ для розничных компаний, но у этой платформы также есть некоторые недостатки, и поэтому она также потерпела неудачу в процессе отбора:

**Сложность реализации:** Некоторым компаниям может быть сложно внедрить retailcrm, что может потребовать времени и ресурсов для обучения персонала и настройки системы в соответствии с конкретными потребностями бизнеса.

**Высокая стоимость:** использование retailCRM может быть дорогостоящим для малого бизнеса из-за высокой стоимости обучения, интеграции с другими услугами и обучения персонала.

**Необходимость тщательного планирования:** Успешное использование RETAILCRM требует тщательного планирования и надлежащей подготовки данных, в противном случае могут возникнуть проблемы с точностью и полнотой информации.

**Ограниченная гибкость:** благодаря функциональности функции retailCRM некоторые пользователи могут столкнуться с ограничениями при настройке систем для своих собственных бизнес-процессов.

**Зависимости от подключения к Интернету:** retailCRM - это облачная платформа, поэтому для нее требуется стабильное подключение к Интернету. Это может быть проблемой в случае сбоя сети.

**Недостаточная поддержка:** в редких случаях пользователи могут столкнуться с проблемами, связанными с поддержкой разработчиков retailCRM, что может замедлить процесс решения.

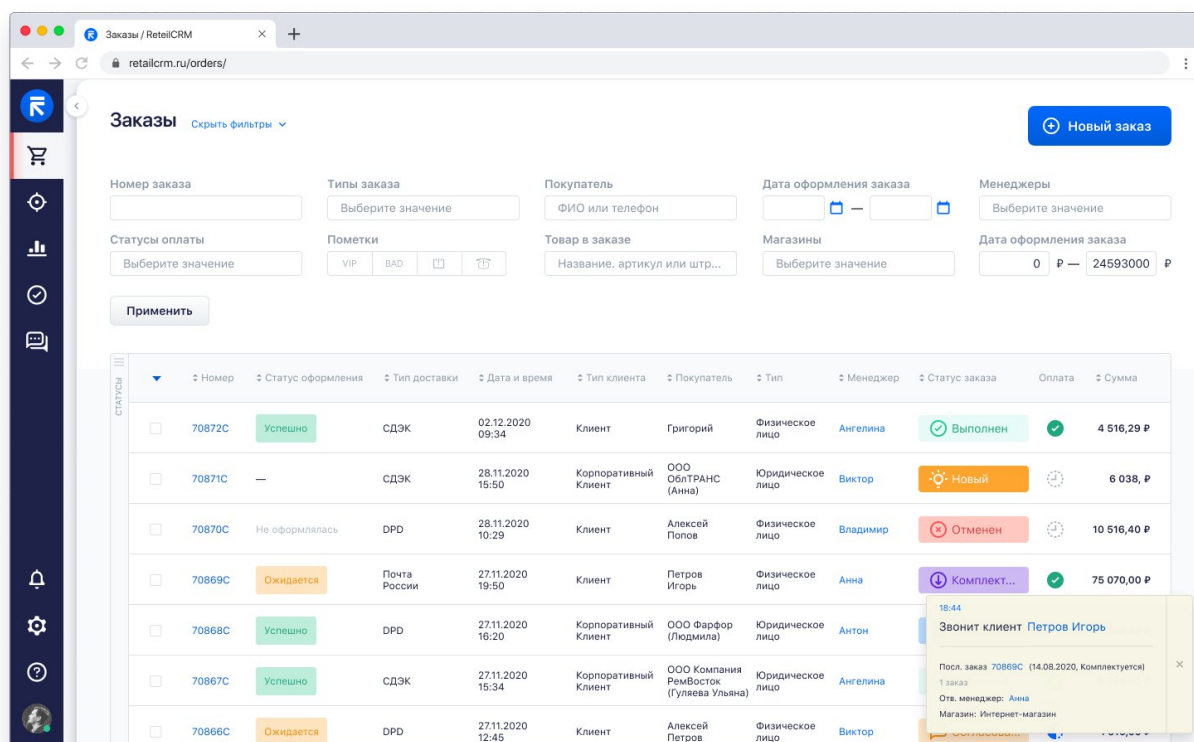


Рисунок 1.3 – Интерфейс RetailCRM

#### 4) Terrasoft CRM:

**Terrasoft CRM** - это платформа управления взаимоотношениями с клиентами (CRM), предназначенная для автоматизации маркетинга, продаж и обслуживания клиентов. Вот некоторые ключевые особенности Terrasoft crm:

**Маркетинг:** Эта платформа предоставляет инструменты для создания маркетинговых кампаний, управления маркетинговыми активами, анализа результатов и автоматизации маркетинговых процессов.

**Продажи:** Terrasoft CRM помогает вам управлять продажами, отслеживать воронку продаж, управлять контактами и возможностями, а также анализировать данные о продажах.

**Обслуживание клиентов:** Платформа предоставляет инструменты для управления запросами, поддержкой и обслуживанием клиентов, помогая компаниям эффективно реагировать на запросы и проблемы клиентов.

**Анализ:** Terrasoft CRM предоставляет возможности анализа данных, отчетности и панели мониторинга для отслеживания ключевых показателей производительности.

**Интеграция и гибкость:** платформа имеет архитектуру с открытым исходным кодом с гибкими конфигурациями и средствами интеграции с другими системами.

**Облачное и локальное развертывание:** Terrasoft CRM предлагает как облачное, так и локальное развертывание в зависимости от потребностей вашей компании.

**Мобильное приложение:** существует мобильное приложение, которое позволяет сотрудникам работать в CRM из любой точки мира.

**Terrasoft CRM** широко используется компаниями в различных отраслях благодаря своей гибкости, параметрам настройки для конкретных бизнес-потребностей и широкому спектру функций для управления взаимоотношениями с клиентами.

**Terrasoft CRM** имеет множество преимуществ, таких как широкий спектр функций и гибкость настройки, а также некоторые недостатки, которые не были выбраны.:

**Сложность настройки:** настройка и внедрение Terrasoft crm может потребовать больших усилий и ресурсов, особенно для крупных компаний и сложных бизнес-процессов.

**Стоимость:** Для некоторых компаний стоимость лицензирования и внедрения Terrasoft CRM может быть очень высокой, особенно когда требуется масштабирование или октябрьская дополнительная интеграция.

**Потребность в обучении персонала:** с новой CRM-системой вам может потребоваться время и ресурсы для обучения ваших сотрудников, и им может потребоваться аналогичная платформа раньше.

**Интеграция с другими системами:** интеграция Terrasoft CRM с другими системами может быть сложной задачей. Это особенно верно, если у компании уже есть сложная программная экосистема.

**Ограничения масштабируемости:** Некоторые крупные компании могут иметь ограничения в отношении масштабируемости и способности адаптировать платформу к сложным бизнес-процессам.

**Зависимости от поставщика:** Как и в случае с любой другой облачной платформой, использование Terrasoft CRM означает зависимость от поставщиков и их политик обновления и поддержки.

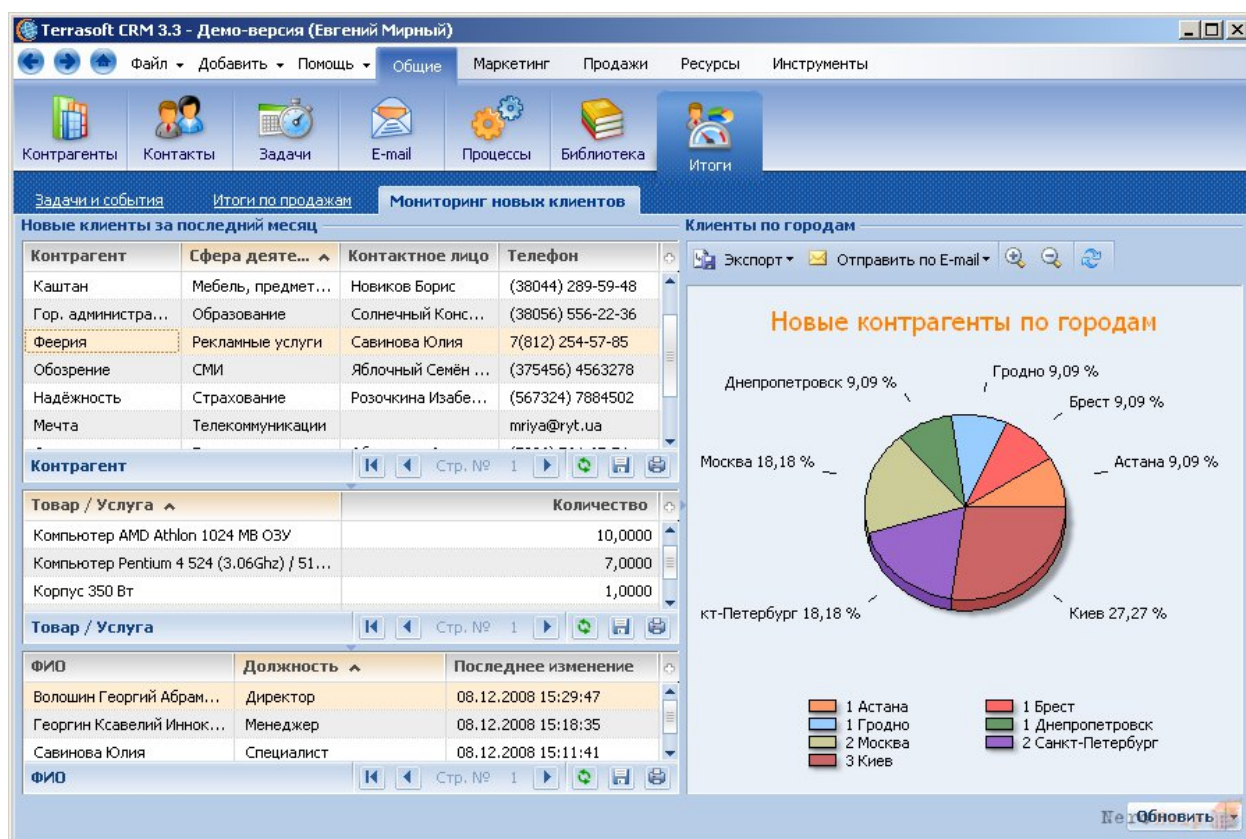


Рисунок 1.4 – Интерфейс Terrasoft CRM

## 5) Croc CRM:

**Croc CRM** - это российская CRM-система, разработанная Срос. Он предназначен для автоматизации управления взаимоотношениями с клиентами (CRM) и охватывает широкий спектр функций для эффективного



управления бизнесом. Некоторые ключевые особенности **Croc crm** включают:

**Модуль управления продажами:** Croc CRM предоставляет инструменты для управления всеми этапами продаж, от привлечения потенциальных клиентов до заключения сделок

**Модуль управления контактами и клиентами:** система позволяет вам управлять информацией о контактах и клиентах, историей взаимодействия, перепиской, задачами и другими данными для эффективного общения.

**Анализ и отчетность:** Croc CRM предоставляет инструменты для анализа данных, отчетности и мониторинга ключевых показателей эффективности компании.

**Интеграция с другими системами:** CRM-система Croc упрощает рабочий процесс, предоставляя возможность интеграции с другими платформами и сервисами.

**Пользовательский интерфейс и настраиваемость:** интерфейс Croc CRM удобен для пользователя, а также предоставляет параметры настройки, адаптированные к потребностям конкретной компании.

**Облачные решения:** Croc CRM предлагает облачное решение, которое использует Интернет для обеспечения доступности данных из любой точки мира.

**Поддержка и обновления:** Croc будет оказывать поддержку пользователям системы и регулярно выпускать обновления для повышения функциональности и безопасности.

**Croc CRM** - популярное решение среди российских компаний для автоматизации процессов декоммунизации клиентской базы и повышения эффективности обслуживания клиентов.

**Croc CRM** - это мощный инструмент для управления взаимоотношениями с клиентами, но у него также есть некоторые недостатки, из-за которых он не выбран:

**Высокая стоимость:** внедрение и использование Croc crm может быть дорогостоящим для малого бизнеса и стартапов из-за высокой стоимости. licensing.In также обучение персонала и системная поддержка.

**Сложность реализации:** Некоторые пользователи считают, что процесс внедрения Croc crm может быть сложным и что необходимо обучать сотрудников и настраивать систему в соответствии с конкретными потребностями компании.

**Ограниченная гибкость:** Croc CRM предлагает настраиваемые функции, но гибкость системы имеет свои пределы при попытке адаптироваться к конкретным бизнес-процессам.

**Интернет-зависимость:** Croc CRM - это облачное решение, поэтому вам необходим постоянный доступ в Интернет для работы вашей системы. Это может привести к проблемам, если нет подключения.

**Недостаточная интеграция с другими системами:** некоторые пользователи могут испытывать ограничения при интеграции Croc crm с другими внутренними или внешними системами, в зависимости от платформы.

**Отсутствие некоторых расширенных функций:** пользователям, которым требуются определенные расширенные функции в области аналитики, автоматизации процессов или маркетинга, Croc CRM предоставляет все необходимые им инструменты

## **1.4 Обоснование выбора программных средств для создания информационной системы**

Инструменты для разработки CRM-системы

### **1) Laravel**

Laravel был выбран в качестве основной платформы для создания CRM-систем по нескольким причинам:

**Удобство и простота:** Laravel - одна из самых популярных фреймворков PHP благодаря интуитивно понятному синтаксису и простоте обучения.<sup>1</sup> Его архитектура и файловая структура способствуют быстрой разработке и облегчают понимание кода даже начинающим разработчикам. Это особенно важно для проектов, где требуется быстрое развертывание и минимальные затраты на обучение

**Мощные инструменты:** Laravel предлагает широкий спектр встроенных инструментов для работы с базами данных, включая осмысленный ORM, удобную маршрутизацию, встроенную аутентификацию и авторизацию, а также мощные функции для тестирования и отладки кода. Эти инструменты могут значительно сократить время разработки и улучшить качество конечного продукта.

**Сообщества и документация:** у Laravel самое активное и дружелюбное сообщество в рамках PHP. Сообщество регулярно выпускает обновления и улучшения, оказывает помощь и поддержку новичкам, а также активно разрабатывает плагины и расширения. Обширная и понятная документация поможет вам быстро найти ответы на новые вопросы и эффективно решать проблемы в процессе разработки.

**Безопасность:** Laravel обеспечивает повышенную безопасность, предоставляя встроенные механизмы для защиты от распространенных уязвимостей в Интернете, таких как внедрение SQL, межсайтовые сценарии (XSS) и подделка межсайтовых запросов (CSRF). Это особенно важно при разработке CRM-системы, в которой безопасность данных клиентов является приоритетом.

## 2) PhpStorm

**PhpStorm** был выбран в качестве основной среды разработки (IDE) из-за следующих преимуществ:

**Интеграция с Laravel:** PhpStorm предоставляет мощные инструменты для работы с PHP и Laravel framework, включая встроенные подсказки, завершение кода и интеграцию с композитором. Это значительно упрощает процесс разработки, отладки и тестирования кода, что особенно важно при создании сложных систем, таких как CRM.

**Автоматизация задач:** PhpStorm предоставляет множество инструментов для автоматизации повседневных задач, таких как рефакторинг кода, обнаружение и исправление ошибок, работа с базами данных и системами контроля версий. Это позволяет ускорить процесс разработки и снизить вероятность ошибок.

**Поддержка других технологий:** PhpStorm поддерживает не только PHP, но и HTML, CSS, JavaScript, SQL и т. Д. Он также поддерживает другие языки программирования и технологии, такие как английский и французский языки программирования. Это делает его универсальным инструментом, который вы можете использовать для улучшения различных аспектов вашей CRM-системы, включая интерфейс и серверную часть.

**Инструменты совместной работы:** PhpStorm включает инструменты для совместной работы, такие как интеграция с системами управления версиями (деки, SVN), поддержка кода и совместное редактирование, чтобы упростить процесс совместной работы и совместного использования кода между разработчиками.

## 3) Figma

**FIGMA** была выбрана для создания макета CRM-системы и планирования функций по следующим причинам:

**Интуитивно понятный интерфейс:** Figma предоставляет интуитивно понятный и простой в освоении интерфейс, который позволяет быстро приступить к работе над проектами и минимизировать время обучения.

**Широкий спектр инструментов:** Figma предлагает полный набор инструментов для создания прототипов, макетов и проектирования интерфейсов. Это упрощает визуализацию идей и концепций, позволяя быстро вносить изменения и корректировки на ранних этапах разработки.

**Создание интерактивных прототипов:** возможность создания интерактивных прототипов позволяет протестировать функциональность и простоту использования вашей системы даже на этапе проектирования. Это поможет вам выявить и устранить потенциальные проблемы до начала разработки, сократив временные и ресурсные затраты.

**Адаптивный дизайн:** Figma позволяет создавать дизайны, которые легко адаптируются к различным устройствам и разрешениям экрана. Это особенно важно для CRM-системы, которая может использоваться на различных устройствах, включая настольные компьютеры, планшеты и смартфоны.

**Совместная работа в режиме реального времени:** Figma позволяет нескольким дизайнерам и разработчикам одновременно работать над проектами, обмениваться идеями и вносить изменения на лету.

#### **4. OpenServer**

**OpenServer** был выбран для локальной разработки по нескольким причинам:

**Простота установки и использования:** OpenServer предоставляет простой и интуитивно понятный интерфейс для настройки и настройки серверных сред. Это позволяет быстро приступить к работе над вашим проектом, избегая сложных настроек и конфигураций

**Поддержка PHP и MySQL:** OpenServer содержит все компоненты, необходимые для работы с вашим проектом Laravel, включая PHP и MySQL.

Это обеспечивает совместимость с платформой и позволяет приложению работать без октября дополнительной настройки.

**Легкость переключения версий PHP:** OpenServer упрощает переключение между различными версиями PHP. Это полезно при работе с различными проектами, для которых требуется определенная версия PHP. Это обеспечивает гибкость и удобство разработки.

**Поддержка дополнительных инструментов:** OpenServer поддерживает другие полезные инструменты, такие как Composer, который обычно используется в Laravel для управления зависимостями. октябре 2018 г. Это делает процесс разработки более эффективным и упрощает управление проектами

**Активное сообщество и регулярные обновления:** OpenServer имеет активное сообщество пользователей, которое предоставляет поддержку и регулярные обновления. Это обеспечивает стабильную работу сервера и возможность быстрого решения проблем в случае их возникновения.

**Собственная разработка:** OpenServer позволяет запускать ваш веб-сайт на локальном сервере без доступа в Интернет, что значительно ускоряет процесс разработки и тестирования и снижает потерю данных.

## **2. Проектная часть**

### **2.1 Проектирование архитектуры информационной системы**

#### **2.1.1 Диаграммы UML**

Проектирование архитектуры информационной системы (ИС) является важным этапом разработки, который определяет структуру, компоненты и взаимодействия различных частей системы. На данном этапе разрабатываются основные схемы и модели, которые послужат руководством для дальнейшей реализации системы. Одним из эффективных инструментов проектирования ИР-архитектур является использование диаграмм UML (унифицированного языка моделирования).

**Унифицированный язык моделирования (UML)** - это стандартизированный язык моделирования, используемый для описания, визуализации, проектирования и документирования компонентов информационной системы. UML предоставляет набор графических представлений и диаграмм для описания как статических, так и динамических аспектов системы.

При проектировании архитектуры информационной системы используются различные типы диаграмм UML, каждая из которых фокусируется на определенных аспектах системы:

##### **1) Диаграмма DFD**

**Диаграмма потоков данных (DFD)** - это графический инструмент, используемый для моделирования и документирования потока данных в информационной системе. DFD позволяет визуализировать, как данные перемещаются по системе, какие процессы обрабатываются и где они хранятся. Этот инструмент особенно полезен для анализа существующих систем

и разработки новых систем. DFD состоит из нескольких важных элементов:

### **Процесс**

**Описание:** Операция - это действие или функция, которая преобразует входные данные в выходные. На рисунке показан процесс в виде круга или эллипса.

**Пример:** Транзакция может быть указана как "Обработка заказа" или "Ввод данных".

### **Поток данных**

**Описание** - Поток данных показывает перемещение данных между транзакциями, внеш декоммунизированными объектами и хранилищами данных. Поток данных представлен стрелками.

**Пример:** поток данных может означать "заказ клиента" или "отчет о продажах".

### **хранилище данных**

**Описание:** Хранилище данных - это место, где данные хранятся постоянно или временно. На рисунке хранилище данных изображено в виде параллельной линии или открытого прямоугольника.

**Пример:** хранилище данных может быть указано как "клиентская база данных" или "Файловая система".

### **Внешние сущности**

**Описание:** Внешние сущности - это источник или получатель данных, который находится за пределами рассматриваемой системы. Они нарисованы в виде прямоугольника.



**Пример:** внешний объект может быть указан как "Клиент" или "Поставщик".

### **Уровень DFD**

Его можно разделить на несколько уровней, чтобы обеспечить различную степень детализации DFD:

#### **Контекстная диаграмма (уровень 0-DFD)**

**Описание:** Контекстная диаграмма представляет наиболее распространенный уровень DFD, который представляет систему как единый процесс и взаимодействует с внешними объектами.

**Цель:** предоставить обзор системы и ее окружения.

#### **Цифра уровня 1 (уровень 1DFD)**

**Описание:** Этот уровень детализации разбивает основной процесс контекстной диаграммы на подпроцессы, которые декоммунизируют основные функции системы и взаимодействие между ними

**Цель:** обеспечить более подробное понимание основных функций системы.

**Диаграмма более низкого уровня (уровень 2, уровень 3 и т. д.)**

**Описание:** Эти диаграммы дают более подробное представление о подпроцессах и показывают небольшие функции и взаимодействия.

**Цель:** разработать конкретные части системы для более глубокого анализа и проектирования.

### **Преимущества использования DFD**

**Простота и наглядность:** в DFD используются простые графические элементы, которые облегчают чтение и понимание для всех участников проекта.

**Улучшенная коммуникация:** DFD способствует лучшему пониманию процессов и потоков данных между разработчиками, пользователями и другими заинтересованными сторонами.

**Выявление проблем и узких мест:** DFD-анализ помогает вам выявлять потенциальные проблемы в вашем потоке данных и процессах и устранять их на ранних этапах разработки.

**Гибкость:** DFD можно использовать для анализа существующих систем и разработки новых, что делает их универсальным инструментом системного анализа и проектирования.

**В моем случае DFD выглядит следующим образом:**

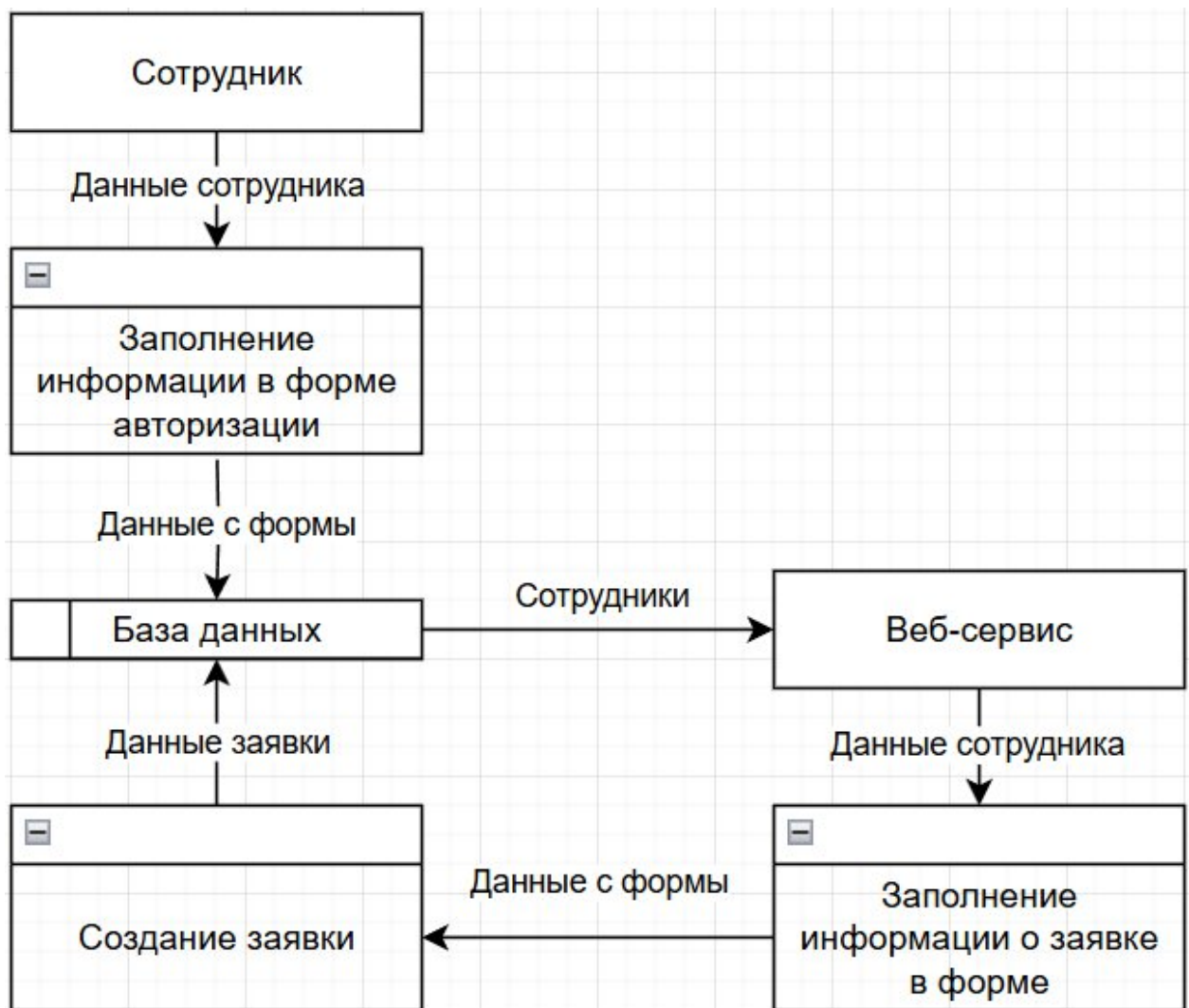


Рисунок 2.1 – DFD диаграмма

## 2) IDEF диаграмма

**IDEF** (интегрированное определение для моделирования функций) - это методология, используемая для моделирования и анализа сложных систем. IDEF содержит множество стандартов, каждый из которых разработан для конкретных аспектов системного анализа и проектирования. Наиболее распространенными являются IDEF0 для функционального моделирования, IDEF1X для информационного моделирования и *idef3* для моделирования процессов. В этой части завершающего проекта мы подробно рассмотрим диаграмму IDEF, ее назначение, структуру, компоненты и примеры реализации.

## **IDEF0: Функциональное моделирование**

### **Цель и применение**

**IDEF0** используется для функционального моделирования и может описывать декоммунизировать функции системы и взаимодействия между ними. Этот методологический подход помогает идентифицировать, анализировать и документировать ключевые функции и их взаимосвязи, которые имеют решающее значение для понимания и улучшения процессов в организации.

### **Функциональный блок**

**Описание:** Представляет собой основную функцию или процесс системы. На рисунке они изображены в виде прямоугольника.

**Пример:** функциональный блок может означать "принятие заказа", "обработка заказа" и "доставка товара".

### **Стрелки**

**Описание:** Декоммунизирует взаимодействие между функциональными блоками. Стрелки могут представлять входные данные, выходные данные, управляющие сигналы и механизмы.

**Пример:** Стрелка может отображать поток данных, такой как информация о заказе, подтверждение заказа и команды управления.

### **Контекст и границы**

**Описание:** Определяет границы рассматриваемой системы. Контекстная диаграмма (верхний уровень) показывает взаимодействие всей системы с внешней средой.

**Пример:** На контекстной диаграмме можно отобразить взаимодействие между системой декоммунизации заказов и клиентами и поставщиками.

### **Декомпозиция**

**Описание:** Функция может быть разбита на более мелкие подфункции, которые способствуют детальному анализу и пониманию процесса.

**Пример:** функцию "Обработка заказа" можно разделить на подфункции: "Проверка доступности продукта", "Подтверждение заказа" и "Отправка уведомлений клиентам".

## **IDEF1X: Информационное моделирование**

### **Цель и применение**

**IDEF1X** используется для информационного моделирования и может определять структуру данных и отношения внутри системы. Этот стандарт используется при разработке реляционных баз данных, которые обеспечивают ясность и точность определения данных.

### **Основные компоненты *idef1x***

#### **Актив**

**Описание.** Представляет объект, который необходимо записать и сохранить в системе данных. Рисунок показан в виде прямоугольника.

**Пример:** объект может представлять "клиент", "заказ" и "продукт".

#### **Атрибуты**

**Описание:** Свойство или свойство объекта. Атрибуты перечислены в прямоугольнике сущности.

**Пример:** Атрибуты сущности "Клиенты" могут включать "Имя", "Адрес" и "Номер телефона".

### **Отношения**

**Описание:** Декоммунизирует взаимодействие между сущностями. Диаграмма рисуется как линия, соединяющая объекты.

**Пример:** Декоммунизация между "клиентом" и "заказом" означает, что у каждого клиента может быть более одного заказа.

## **IDEF3: Моделирование процессов**

### **Цель и применение**

IDEF3 используется для моделирования процессов и может документировать ряд событий и действий в системе. Этот стандарт помогает анализировать и улучшать бизнес-процессы.

### **Основные компоненты idef3**

#### **Узлы транзакций**

**Описание.** Представляет действие или событие в транзакции. Форма рисуется в виде прямоугольника или прямоугольника круглой формы.

**Пример:** Узел транзакции может означать "получить заявку", "обработать платеж" и "отправить товар".

#### **Поток (соединение)**

**Описание:** Отображает порядок и логику процесса. Потоки представлены стрелками, соединяющими узлы процесса.

**Пример:** в потоке может отображаться "Принять запрос", за которым следует "Обработка платежей". Объект

**Описание:** Объекты обработки ввода / вывода, такие как документы, данные и материалы. Рисунок рисуется в виде овала или округлого прямоугольника.

**Пример:** объект может означать "заказ клиента" или "подтверждение оплаты".

### **Преимущества использования диаграмм IDEF**

**Стандартизация и декомбинация:** IDEF обеспечивает стандартизированный подход к моделированию, упрощению взаимодействия и взаимопонимания между разработчиками и пользователями.

**Гибкость и адаптивность:** IDEF можно использовать для моделирования различных аспектов системы, от функциональных процессов до структур данных и последовательностей событий.

**Всесторонняя аналитическая поддержка:** диаграммы IDEF позволяют проводить углубленный анализ сложных систем, выявлять проблемы и разрабатывать оптимальные решения.

**Документация и отчетность:** IDEF предоставляет четкую и структурированную документацию, которая упрощает управление проектами и обслуживание системы.

**IDEF** диаграмма для моего веб-сервиса представлена следующим образом:

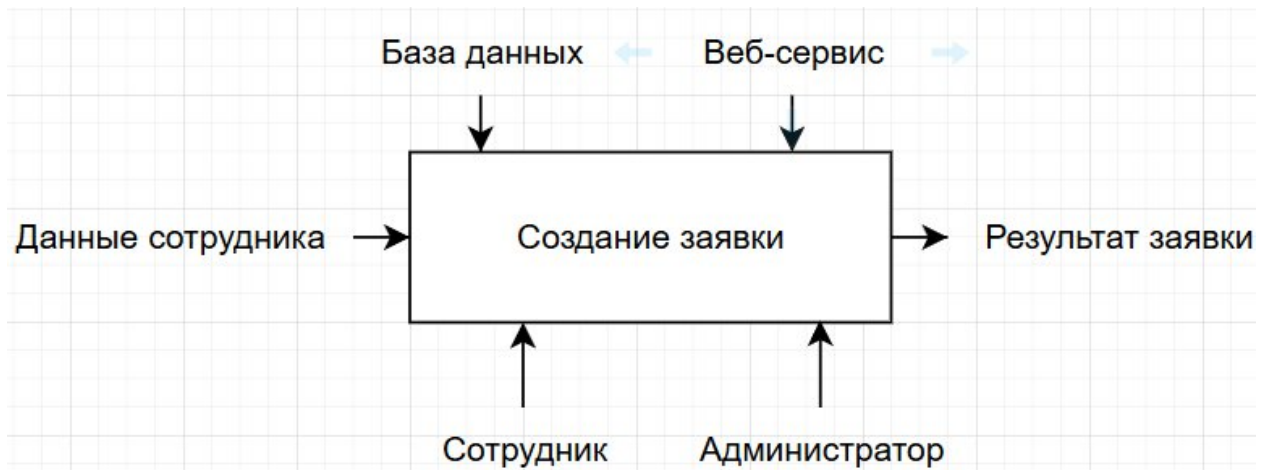


Рисунок 2.2 – IDEF диаграмма

### 3) Диаграмма классов

Диаграммы классов являются одним из основных строительных блоков унифицированного языка моделирования (UML) и широко используются в объектно-ориентированном анализе и проектировании систем. Эти диаграммы позволяют визуализировать структуру системы, показывая классы, их атрибуты, методы и отношения между ними.

#### **Цель диаграммы классов**

**Диаграмма классов предназначена для:**

**Моделирование структуры системы:** Диаграмма классов показывает статическую структуру системы, ее класс, атрибут декоммунизацию и методы, а также отношения между классами.

**Для облегчения разработки и документации:** диаграммы классов используются на различных этапах жизненного цикла разработки программного обеспечения, включая анализ требований, проектирование, внедрение и тестирование.

**Поддержка объектно-ориентированного программирования:** диаграммы классов помогают вам понять и использовать



принципы объектно-ориентированного программирования (ООП), такие как инкапсуляция, наследование и полиморфизм.

## **Основные компоненты диаграммы классов**

### **Класс**

**Описание:** Класс представляет собой основные строительные блоки системы. Каждый класс определяет набор объектов с общими атрибутами и методами.

**Структура:** класс представлен в виде прямоугольника, разделенного на 3 части: имя класса, атрибуты и методы.

### **Атрибуты**

**Описание.** Атрибут определяет свойство класса. Каждый атрибут имеет имя и тип данных.

**Пример:** в классе "Клиент" атрибутами могут быть "имя" (строка), "адрес" (строка) и "номер телефона" (строка).

### **Метод**

**Описание.** Метод описывает поведение класса. Каждый метод имеет имя и параметры.

**Пример:** в классе "Клиент" методами могут быть "создать заказ ()" и "изменить адрес ()".

### **Отношения**

**Ассоциации:** Показывают связи между классами. Ассоциации могут быть однонаправленными или двунаправленными.

**Декоммунизация:** особый тип ассоциации, который показывает взаимосвязь между целым и частью. Агрегация показывает, что 1 класс состоит из экземпляров другого класса.

Составное сложение: более строгий тип сложения, который показывает, что часть строго связана с целым.

Декоммунизация: отображает иерархические отношения между классами. Класс-преемник наследует атрибуты и методы базового класса.

Реализация: используется для отображения взаимосвязи декоммунизации между интерфейсами и классами, реализующими эти интерфейсы.

Типы отношений в диаграммах классов

Ассоциация

Пояснение: Декоммунизация - это связь между 2 классами, которая показывает, как объект одного класса связан с объектом другого класса.

Пример: взаимосвязь между классами "Клиент" и "Заказ" может указывать на то, что клиент может декоммунизировать заказ.

Большинство

Описание: Кратность определяет количество экземпляров 1 класса, которые могут быть связаны с 1 экземпляром другого класса.

Пример: Кратность 1.\* Указывает, что между "Клиентом" и "Заказом" у клиента может быть дека заказов.

Сбор

Пояснение: Декоммунизация - это слабая связь между целым и частью, при которой часть может существовать независимо от целого.

Пример: класс "Команда" может быть объединен с классом "Игрок", который указывает, что команда состоит из игроков.

#### Композитная агрегация (состав)

Описание: Составное декоммунизация - это тесная взаимосвязь между целым и частью, часть не может существовать без целого.

Пример: класс "Дом" может быть конфигурацией класса "Комната", который показывает, что комната не может существовать без дома.

#### Наследие

Описание: Наследование - это отношение "есть", при котором один класс (подкласс) наследует атрибуты и методы другого класса (суперкласс).

Пример: класс Circle может наследовать от класса Shape, чтобы указать, что круг является формой. Приложение

Описание. Реализация - это связь между декоммунизацией интерфейса и классом, реализующим этот интерфейс.

Пример: класс "Квадрат" может реализовать интерфейс "Геометрическая фигура", который показывает, что квадрат является конкретной реализацией геометрической фигуры.

#### Преимущества использования диаграмм классов

Четкость и ясность: Диаграммы классов обеспечивают визуальное представление структуры системы, облегчая понимание и анализ.

Упростите процесс разработки: диаграммы классов помогают разработчикам четко определить требования и системы

проектирования перед написанием кода. Это снижает вероятность ошибок.

Модульность и поддержка повторного использования кода: четкая структура классов упрощает модульное проектирование и упрощает повторное использование кода.

Системная документация: Диаграммы классов служат ключевыми элементами системной документации, облегчая их поддержку и дальнейшее развитие.

## **2.2 Разработка интерфейсов информационной системы**

## **2.3 Разработка клиентской части информационной системы**

## **2.4 Разработка серверной части информационной системы**

## **2.5 Размещение программного продукта в сети Интернет**

## **2.6 Тестирование информационной системы**

### 3. Экономическое обоснование

Разработка веб-сервиса для обработки заявок сотрудников позволит существенно снизить затраты компании в краткосрочной перспективе. Проект включает участие одного программиста, которому потребуется 2 месяца на выполнение всех этапов разработки.

Для оценки экономической эффективности проекта приведена таблица работ (таблица 3.1).

Таблица 3.1 – Этапы разработки информационной системы

| Содержание работы                                     | Количество дней | График работы       |
|---|-----------------|---------------------|
| <b>1 этап – Анализ предметной области</b>             |                 |                     |
| 1.1 Сбор сведений об организации                      | 1               | 22.04.24            |
| 1.2 Классификация информационных систем               | 3               | 23.04.24 - 25.04.24 |
| 1.3 Сбор данных об аналогах информационной системы    | 4               | 26.04.24 - 29.04.24 |
| 1.4 Выбор программных средств                         | 1               | 30.04.24            |
| <b>2 этап – Проектирование информационной системы</b> |                 |                     |
| 2.1 Проектирование архитектуры                        | 6               | 01.05.24 - 06.05.24 |
| <b>3 этап - Программирование</b>                      |                 |                     |
| 3.1 Разработка интерфейсов                            | 8               | 07.05.24 - 14.05.24 |
| 3.2 Разработка клиентской части                       | 12              | 15.05.24 - 26.05.24 |
| 3.3 Разработка серверной части                        | 12              | 27.05.24 -          |

| Содержание работы                     | Количество дней | График работы       |
|---------------------------------------|-----------------|---------------------|
|                                       |                 | 07.06.24            |
| <b>4 этап – Тестирование</b>          |                 |                     |
| 4.1 Тестирование                      | 6               | 08.06.24 - 13.06.24 |
| <b>5 этап – Заключение</b>            |                 |                     |
| 5.1 Анализ результатов тестирования   | 1               | 14.06.24            |
| 5.2 Проведение экономических расчетов | 1               | 15.06.24            |
| 5.3 Оформление пояснительной записки  | 2               | 16.06.24 - 17.06.24 |
| <b>Всего</b>                          | <b>58</b>       |                     |

**Затраты на разработку информационной системы включают:**

**Основная заработная плата:**

- Оклад программиста – 38 000 руб./мес
- Время разработки – 2 месяца
- Основная з/п = 38 000 руб./мес \* 2 мес = 76 000 руб.

**• Страховые взносы:**

- Ставка страховых взносов – 30% от фонда оплаты труда
- Страховые взносы = 76 000 руб. \* 0.3 = 22,800 руб.

**• Прочие прямые затраты:**

- Интернет – 650 руб./мес
- Электроэнергия (расчет в таблице 3.2)

Таблица 3.2 – Расчет электроэнергии

| Наименование оборудования | Кол-во оборудования | Потребление электроэнергии (Вт/ч) | Потребление электроэнергии (кВт/д) |
|---------------------------|---------------------|-----------------------------------|------------------------------------|
| Системный блок            | 1                   | 850                               | 0.850                              |
| Монитор                   | 2                   | 280                               | 0.560                              |
| Освещение                 | 1                   | 35                                | 0.035                              |
| <b>Итого:</b>             |                     |                                   | <b>1.445</b>                       |

- Время разработки и тестирования – 30 дней
- Тариф на электроэнергию – 6.57 руб./кВт\*ч
- Ежедневные затраты = 1.456 кВт/д \* 6.57 руб. = 9,566 руб./день
- Полные затраты = 9,566 руб./день \* 30 дней = 286,98 руб.
- Прочие прямые затраты = 286,98 руб. + 650 руб. \* 2 мес = 1 873,96 руб.

#### 4. Накладные расходы:

- Составляют 27% от основной заработной платы
- Накладные расходы = 76 000 руб. \* 0.27 = 20 520 руб.

Таблица 3.3 – Расчет всех расходов

| Наименование статьи расходов   | Сумма руб.        |
|--------------------------------|-------------------|
| Основная заработная плата      | 76 000            |
| Отчисления по соц. страхованию | 22 800            |
| Прочие прямые затраты          | 1 873,96          |
| Накладные расходы              | 20 520            |
| <b>Итого:</b>                  | <b>100 694,48</b> |

#### Расчет окупаемости информационной системы

Внедрение веб-сервиса позволит сократить затраты на заработную плату сотрудников за счет уменьшения времени на обработку заявок. Планируется сокращение около 25% сотрудников.

- Количество менеджеров – 24 человека

- Заработная плата менеджера – 58 000 руб./мес
- Кол-во сокращенных сотрудников =  $22 * 0.25 = 6$  чел.
- Экономия в месяц = 6 чел. \* 58 000 руб. = 348 000 руб.
- Экономия в год = 348 000 руб. \* 12 мес = 4 176 000 руб.

#### **Годовой экономический эффект**

- $\mathcal{E} = \mathcal{D} - \mathcal{P}$
- $\mathcal{E} = 4\,176\,000 \text{ руб.} - 100\,694,48 \text{ руб.} = 4\,075\,305,52 \text{ руб.}$

#### **Срок окупаемости**

- $\mathcal{O} = \mathcal{P} / \mathcal{D}$
- $\mathcal{O} = 100\,694,48 \text{ руб.} / 4\,176\,000 \text{ руб.} \approx 0.024$  года или 9 дней

В результате, разработка и внедрение веб-сервиса для компании **ЗАО «ФармФирма «Сотекс»** является экономически целесообразной, так как позволит значительно сократить затраты на заработную плату сотрудников и быстро окупится.



## Библиография

1. Справочник JSON [Электронный ресурс]. – [https://schoolsw3.com/jsref/jsref\\_obj\\_json.php](https://schoolsw3.com/jsref/jsref_obj_json.php)
2. Справочник CSS [Электронный ресурс]. – <https://webref.ru/css/>
3. Справочник HTML [Электронный ресурс]. – <https://webref.ru/css/>
4. JSON Server [Электронный ресурс]. – <https://npmjs.com/package/json-server/>
5. Современный учебник JavaScript [Электронный ресурс]. – <https://learn-javascript.ru/>
6. Пакетный менеджер NPM [Электронный ресурс]. – <https://npmjs.com/>
7. GitHub [Электронный ресурс]. – <https://github.com/>
8. Git [Электронный ресурс]. – <https://git-scm.com/>
9. Figma [Электронный ресурс]. – <https://figma.com/>
10. Easings [Электронный ресурс]. – <https://easings.net/ru>
11. StackOverflow форум для разработчиков [Электронный ресурс]. – <https://stackoverflow.com/>
12. Компрессор изображений [Электронный ресурс]. – <https://compressor.io/>
13. Learn Git Branching [Электронный ресурс]. – <https://learngitbranching.js.org/>
14. HTML Book [Электронный ресурс]. – <https://htmlbook.ru/css-cheat-sheet/>
15. Кроссбраузерность [Электронный ресурс]. – <https://caniuse.com/>
16. Google Fonts [Электронный ресурс]. – <https://fonts.google.com/>
17. Валидатор HTML кода [Электронный ресурс]. – <https://validator.w3.org/>
18. Форум с полезной информацией [Электронный ресурс]. – <https://habr.com/ru/>

19. Минификатор кода [Электронный ресурс]. -  
<https://www.minifier.org/>
20. Компрессор SVG изображений [Электронный ресурс]. –  
<https://www.svgomg.net/>