**Содержание**

[ВВЕДЕНИЕ 10](#_Toc167978843)

[ГЛАВА 1 Аналитическая часть 12](#_Toc167978844)

[1.1 Обзор и анализ 12](#_Toc167978845)

[1.2 Анализ требований 14](#_Toc167978846)

[1.3 Анализ спецификации ИС. Постановка задачи 16](#_Toc167978847)

[1.4 Анализ и обоснование выбора технологии проектирования 23](#_Toc167978848)

[1.5 Обоснование среды разработки 36](#_Toc167978849)

[Глава 2 Проектная часть 39](#_Toc167978850)

[2.1 Концептуальная модель информационных потоков данных 39](#_Toc167978851)

[2.2 Функциональная модель информационных потоков данных 40](#_Toc167978852)

[2.3 Логическая схема взаимодействия отдельных частей БД 41](#_Toc167978853)

[2.4 Физическая схема взаимодействия отдельных частей БД 42](#_Toc167978854)

[2.5 Функциональная модель 43](#_Toc167978855)

[2.6 Схема технологического процесса 44](#_Toc167978856)

[2.7 Схема взаимосвязи программных модулей 45](#_Toc167978857)

[2.6 Подробное описание проектирования и разработки приложения 48](#_Toc167978858)

[Глава 3 Разработка документации 51](#_Toc167978859)

[3.1 Руководство пользователю 51](#_Toc167978860)

[3.2 Руководство программисту 56](#_Toc167978861)

[3.3. Тестирование системы 59](#_Toc167978862)

[ГЛАВА 4 Технико-экономическое обоснование разработки программного продукта, который должен содержать следующие подразделы 61](#_Toc167978863)

[4.1 Определение суммарных затрат на разработку 61](#_Toc167978864)

[4.2 Расчет ориентировочной цены 70](#_Toc167978865)

[ЗАКЛЮЧЕНИЕ 72](#_Toc167978866)

[СПИСОК ЛИТЕРАТУРЫ 73](#_Toc167978867)

# ВВЕДЕНИЕ

В современном мире бизнеса, где каждый конкурентный шаг и каждая операция направлены на повышение эффективности, создание и внедрение системы управления отношениями с клиентами (CRM) для компании ОсОО «Сэм экспресс», специализирующейся на перевозке товаров между странами, становится ключевым фактором успеха. Эта CRM-система позволит автоматизировать и упростить процессы взаимодействия с клиентами, включая управление контактами, обработку заказов, анализ поведения клиентов и обеспечение высокого уровня обслуживания.

Главная цель CRM-системы для компании ОсОО «Сэм экспресс» заключается в улучшении взаимодействия с клиентами, повышении уровня удовлетворенности заказчиков и увеличении объема перевозок. Это достигается за счет автоматизации рутинных процессов, предоставления персонализированных услуг и анализа данных для принятия обоснованных решений.

Задачи CRM-системы включают управление контактами, обработку заказов, анализ поведения клиентов, обслуживание клиентов, маркетинг и продажи. Эти задачи решаются путем автоматизации процессов, предоставления персонализированных услуг и анализа данных для принятия обоснованных решений.

Дополнительные аспекты разработки CRM-системы для компании ОсОО «Сэм экспресс» включают в себя интеграцию с другими системами, разработку мобильных и веб-приложений, обеспечение безопасности и конфиденциальности данных, аналитику и отчетность, обучение и поддержку, масштабируемость и гибкость, а также тестирование и внедрение.

Внедрение CRM-системы для компании ОсОО «Сэм экспресс» требует комплексного подхода, включающего планирование, разработку, тестирование и поддержку. Учитывая современные технологические требования и особенности рынка, CRM-система может стать ключевым фактором успеха компании, улучшая отношения с клиентами и повышая эффективность бизнес-процессов.

Выбор технологий для разработки CRM-системы зависит от особенностей бизнеса, функциональных требований, бюджета и доступных ресурсов. Сегодня широко используются технологии, такие как JavaScript (JS) и React для разработки пользовательского интерфейса и обработки клиентской части системы, а также SQL и Node.js для работы с базами данных и серверной логикой. Эти технологии обеспечивают высокую производительность, гибкость и масштабируемость системы.

Объектом исследования является сама CRM-система, разработанная и используемая компанией ОсОО «Сэм экспресс» для управления отношениями с клиентами. Предметом исследования является комплекс программных и аппаратных средств, который обеспечивает эффективное взаимодействие и обмен информацией между компанией и ее клиентами.

Целью проекта является создание и внедрение CRM-системы, которая будет способствовать улучшению отношений с клиентами, повышению уровня удовлетворенности заказчиков и увеличению объема перевозок. Основная задача заключается в автоматизации и оптимизации процессов взаимодействия с клиентами, что позволит компании ОсОО «Сэм экспресс» оставаться конкурентоспособной на рынке.

# ГЛАВА 1 Аналитическая часть

## 1.1 Обзор и анализ

Обзор и анализ существующих систем управления перевозками товаров между странами для ОсОО «Сэм экспресс» показывает, что выбор подходящей системы зависит от нескольких ключевых факторов, включая функциональность, масштабируемость, интеграцию с существующими системами, безопасность и поддержку разработчика. Ниже представлен обзор некоторых популярных систем:

1. SAP Transportation Management:

SAP TM предлагает комплексное решение для управления всем жизненным циклом транспортных операций. Она обеспечивает высокую степень автоматизации и интеграции с другими системами SAP, что делает ее привлекательным выбором для крупных предприятий. Однако, высокая стоимость внедрения и сложность настройки могут быть препятствием для небольших и средних компаний.

2. Oracle Transportation Management (OTM):

OTM предоставляет широкий спектр функциональности для управления транспортными операциями, включая планирование, исполнение и отслеживание грузов. Она обладает гибкой архитектурой, что позволяет легко интегрировать ее с другими системами и настраивать под уникальные потребности компании.

3. Cargowise:

Cargowise представляет собой облачное решение для управления логистическими процессами, включая перевозку товаров между странами. Она обладает интуитивным интерфейсом и широким набором функций, таких как отслеживание грузов, управление документами и таможенное оформление. Однако, некоторые компании могут оказаться ограниченными в функциональности и масштабируемости.

Выбор технологий для разработки информационной системы управления перевозками:

Для разработки пользовательского интерфейса и клиентской части информационной системы JavaScript (JS) и React представляют собой отличный выбор. Они обеспечивают высокую производительность, модульность и удобство в использовании, что делает разработку и поддержку системы более эффективными.

Для серверной разработки и управления данными SQL и Node.js являются надежным решением. SQL обеспечивает эффективное управление данными в базах данных, а Node.js позволяет создавать масштабируемые и быстрые веб-приложения.

Выбор подходящей информационной системы для ОсОО «Сэм экспресс» требует тщательного анализа бизнес-процессов, потребностей и бюджета компании. Использование современных технологий, таких как JS, React, SQL и Node.js, поможет создать эффективное и гибкое решение, способное удовлетворить потребности компании и ее клиентов.

Дополнительные аспекты разработки информационной системы включают:

- Интеграцию с транспортными платформами и таможенными сервисами.

- Внедрение аналитики для оптимизации маршрутов и снижения затрат.

- Разработку мобильных приложений для отслеживания грузов и взаимодействия с клиентами.

- Обеспечение безопасности данных и соответствие требованиям международных стандартов.

- Обучение сотрудников и предоставление технической поддержки для эффективного использования системы.

Внедрение информационной системы управления перевозками требует комплексного подхода, начиная с планирования и заканчивая поддержкой в эксплуатации. Успешное внедрение такой системы поможет ОсОО «Сэм экспресс» оптимизировать свои логистические процессы и повысить удовлетворенность клиентов.

## 1.2 Анализ требований

Анализ требований к созданию информационной системы для обеспечения перевозки товаров между странами в компании ОсОО «Сэм экспресс» направлен на определение основных функций и технических характеристик, необходимых для эффективной организации логистики и управления клиентскими отношениями. Ниже представлен подробный обзор требований к информационной системе:

1. Управление контактами и заказами:

Обеспечение возможности создания, редактирования и удаления контактной информации клиентов и партнеров.

Реализация системы отслеживания заказов с момента оформления до доставки, включая управление статусами заказов и возвратами.

Классификация и категоризация клиентов и партнеров для удобства поиска и анализа.

2. Система обслуживания клиентов и партнеров:

Механизм для обработки запросов и жалоб клиентов и партнеров, а также предоставления поддержки через различные каналы связи.

Управление обращениями и решение возникающих проблем с доставкой или обслуживанием.

3. Маркетинг и аналитика:

Инструменты для организации маркетинговых кампаний, включая анализ результатов и поведения клиентов.

Анализ данных о перевозках и взаимодействии с клиентами для выявления тенденций и оптимизации логистических стратегий.

4. Безопасность и конфиденциальность данных:

Обеспечение безопасности и конфиденциальности информации о заказах и клиентах с помощью аутентификации, контроля доступа и шифрования данных.

Применение мер защиты от утечки и несанкционированного доступа к информации.

5. Интеграция с другими системами:

Возможность интеграции с другими информационными системами компании для обмена данными и синхронизации информации.

6. Отчетность и аналитика:

Генерация отчетов о перевозках, клиентском поведении и эффективности логистических операций.

Анализ данных для выявления трендов и оптимизации бизнес-процессов.

7. Пользовательский интерфейс:

Интуитивно понятный и удобный интерфейс для сотрудников и клиентов, поддержка работы на различных устройствах и платформах.

8. Масштабируемость и гибкость:

Гибкая архитектура системы, способная адаптироваться к изменяющимся потребностям компании и росту объема данных.

9. Поддержка и обучение:

Обеспечение обучения сотрудников по использованию системы и предоставление технической поддержки для решения проблем и вопросов.

Анализ требований к информационной системе для ОсОО «Сэм экспресс» позволяет выявить ключевые функции и характеристики, необходимые для создания эффективной системы управления перевозками и клиентскими отношениями. Важно учитывать, что специфика требований может различаться в зависимости от особенностей бизнеса и потребностей клиентов. При разработке системы рекомендуется провести дополнительный анализ требований с учетом конкретных целей и задач компании.

## 1.3 Анализ спецификации ИС. Постановка задачи

### 1.3.1 Цель и назначение создания и модернизации модулей или сервисов информационной системы. Функциональные требования

Цель и назначение создания и модернизации модулей или сервисов информационной системы перевозки товаров между странами ОсОО «Сэм экспресс»

Цель создания и модернизации

Цель создания и модернизации модулей или сервисов информационной системы перевозки товаров между странами ОсОО «Сэм экспресс» заключается в повышении эффективности, безопасности и удобства выполнения логистических задач, а также в улучшении опыта клиентов и оптимизации бизнес-процессов. Основные цели включают:

1. Улучшение опыта клиентов:

Оптимизация интерфейса и процессов для удобства заказчиков.

Обеспечение точного и оперативного отслеживания товаров в пути.

Предоставление персонализированных рекомендаций и уведомлений о статусе доставки.

2. Повышение эффективности управления:

Автоматизация логистических процессов, включая управление складом и маршрутизацию.

Улучшение аналитики для принятия обоснованных решений и оптимизации затрат.

Интеграция с внешними системами, такими как системы отслеживания грузов и управления заказами.

3. Обеспечение безопасности:

Защита конфиденциальности данных о грузах и финансовых операциях.

Обновление систем безопасности для соответствия международным стандартам и законодательству.

4. Расширение функциональности:

Внедрение новых возможностей, таких как мобильные приложения для отслеживания грузов.

Диверсификация методов оплаты и расширение географии доставки.

Назначение модулей и сервисов

Назначение модулей и сервисов информационной системы перевозки товаров между странами ОсОО «Сэм экспресс» включает следующие аспекты:

1. Модуль управления грузами:

Регистрация, отслеживание и управление грузами.

Организация маршрутов и оптимизация доставки.

2. Модуль отслеживания и уведомлений:

Предоставление клиентам возможности отслеживать перемещение груза в реальном времени.

Отправка уведомлений о статусе доставки.

3. Модуль управления заказами:

Прием, обработка и управление заказами клиентов.

Автоматизация процессов от покупки до доставки.

4. Модуль аналитики и отчетности:

Сбор данных о производительности и эффективности доставки.

Генерация отчетов для принятия стратегических решений.

5. Модуль обратной связи и поддержки клиентов:

Предоставление каналов связи для обратной связи и поддержки клиентов.

Обработка запросов на возврат или обмен грузов.

6. Модуль безопасности и контроля:

Мониторинг безопасности грузов в пути.

Обеспечение защиты данных о грузах и транзакциях.

7. Модуль маркетинга и продвижения:

Разработка и проведение маркетинговых кампаний для привлечения новых клиентов и удержания существующих.

Управление рекламными материалами и контентом на веб-сайте.

Анализ эффективности маркетинговых стратегий и реакция на рыночные тренды.

8. Модуль управления складом и инвентаризацией:

Организация хранения и отслеживание товаров на складах.

Учет поступления и отгрузки товаров.

Автоматизация инвентаризации и оптимизация запасов.

9. Модуль управления транспортными средствами:

Планирование и координация маршрутов доставки.

Отслеживание транспортных средств и управление логистикой.

Оптимизация использования ресурсов и снижение затрат на доставку.

10 Модуль управления документами и таможенными формальностями:

Обработка необходимых документов для международной перевозки грузов.

Соблюдение таможенных требований и управление импортно-экспортными процедурами.

Мониторинг и контроль за исполнением таможенных процедур.

11 Модуль аудита и контроля качества:

Проведение аудита процессов доставки и обслуживания клиентов.

Мониторинг качества услуг и выявление потенциальных улучшений.

Реализация мер по повышению качества обслуживания и удовлетворенности клиентов.

12. Модуль управления рисками и страхованием:

Оценка и управление рисками, связанными с международной перевозкой грузов.

Подбор и оформление страховых полисов для защиты от потенциальных финансовых потерь.

Реагирование на чрезвычайные ситуации и оказание помощи клиентам в случае возникновения проблем.

Эти модули и сервисы информационной системы ОсОО «Сэм экспресс» совместно работают для обеспечения эффективной и безопасной перевозки товаров между странами, удовлетворения потребностей клиентов и обеспечения успешного функционирования бизнеса.

Таким образом, создание и модернизация модулей и сервисов информационной системы перевозки товаров между странами ОсОО «Сэм экспресс» направлены на улучшение качества услуг, повышение эффективности и обеспечение удовлетворения потребностей клиентов и партнеров компании.

### 1.3.2. Анализ требований к информационно-коммуникационным средствам для решения задачи. Нефункциональные требования.

Нефункциональные требования играют важную роль в обеспечении эффективной работы информационной системы перевозки товаров между странами для ОсОО «Сэм экспресс». Они определяют характеристики системы, которые влияют на её производительность, надежность, доступность и безопасность. Рассмотрим основные нефункциональные требования к этой информационной системе.

1. Производительность и масштабируемость:

Производительность: Система должна обеспечивать быструю обработку запросов и минимальное время загрузки страниц.

Масштабируемость: Информационная система должна быть способной масштабироваться для обеспечения стабильной работы при росте количества запросов и объема данных.

2. Надежность и доступность:

Надежность: Система должна быть устойчива к сбоям и иметь механизмы автоматического восстановления после них.

Доступность: Система должна быть доступна пользователям в любое время с минимальным временем недоступности для технических работ.

3. Безопасность:

Конфиденциальность данных: Личная и финансовая информация пользователей должна быть надежно защищена.

Защита от атак: Система должна быть защищена от различных видов кибератак.

4. Удобство использования и эргономичность:

Интерфейс пользователя: Пользовательский интерфейс должен быть интуитивно понятным и легким в использовании.

Многоязычность и локализация: Система должна поддерживать несколько языков и легко адаптироваться к различным регионам.

5. Совместимость и интеграция:

Кроссбраузерность: Система должна корректно работать на различных веб-браузерах и устройствах.

Интеграция с внешними системами: Система должна поддерживать интеграцию с внешними сервисами.

6. Поддержка и обслуживание:

Документированность: Полная документация по системе должна быть доступна как для разработчиков, так и для пользователей.

Техническая поддержка: Должна быть доступна оперативная техническая поддержка для решения проблем пользователей.

7. Мобильность:

Мобильная оптимизация: Система должна быть оптимизирована для работы на мобильных устройствах.

Мобильные приложения: Должна быть возможность использования мобильных приложений для расширения функциональности.

8. Экономическая эффективность:

Затраты на обслуживание: Система должна минимизировать затраты на обслуживание и развитие.

Возврат инвестиций: Система должна обеспечивать высокий возврат инвестиций через повышение эффективности и улучшение пользовательского опыта.

Эти нефункциональные требования также обеспечивают следующие аспекты:

9. Гибкость и расширяемость:

Система должна быть гибкой и легко расширяемой для внесения изменений в соответствии с растущими потребностями бизнеса и технологическими требованиями.

10. Аналитика и отчетность:

Поддержка аналитики: Система должна предоставлять возможности для сбора данных и анализа производительности, что позволит оптимизировать процессы и принимать обоснованные решения.

11. Согласованность и целостность данных:

Согласованность данных: Информация, хранящаяся в системе, должна быть согласованной и актуальной во всех её частях и модулях.

12. Экологическая устойчивость:

Минимизация экологического воздействия: Система должна разрабатываться с учетом минимизации использования ресурсов и энергии для снижения негативного воздействия на окружающую среду.

13. Соответствие законодательству:

Соблюдение нормативных требований: Система должна соответствовать законодательным и регуляторным требованиям, включая защиту персональных данных и соблюдение прав потребителей.

14. Обучение и развитие персонала:

Обучение пользователей: Система должна предоставлять ресурсы и обучение для пользователей, чтобы они могли эффективно использовать все её функциональные возможности.

15. Инновации и технологический прогресс:

Внедрение новых технологий: Система должна быть готова к внедрению новых технологий и инноваций для постоянного улучшения своей функциональности и производительности.

Эти нефункциональные требования содействуют созданию и поддержанию информационной системы перевозки товаров между странами для ОсОО «Сэм экспресс» на высоком уровне, обеспечивая её гибкость, безопасность, и экономическую эффективность.

## 1.4 Анализ и обоснование выбора технологии проектирования

### 1.4.1 Язык программирования JavaScript

JavaScript (JS) - это высокоуровневый, интерпретируемый язык программирования, который широко используется для разработки веб-приложений и динамических веб-страниц. Он был создан для придания интерактивности и динамического поведения веб-страницам. Вот подробное описание основных характеристик и возможностей JavaScript:

1. Синтаксис и типы данных:

- JavaScript[[5]](#_СПИСОК_ЛИТЕРАТУРЫ) имеет синтаксис, похожий на другие языки программирования, такие как C, C++ и Java. Это делает его относительно простым для изучения и использования.

- В языке поддерживаются различные типы данных, включая числа, строки, булевы значения, массивы, объекты, null и undefined. Кроме того, JavaScript предоставляет возможности работы с регулярными выражениями и датами.

2. Объектно-ориентированное программирование (ООП):

- JavaScript является прототипно-ориентированным языком, где объекты создаются с использованием функций-конструкторов или литералов объектов.

- Язык поддерживает наследование, инкапсуляцию и полиморфизм, что является ключевыми принципами объектно-ориентированного программирования.

- С помощью прототипов JavaScript позволяет создавать цепочки наследования и повторно использовать код.

3. Функции:

- Функции в JavaScript являются объектами первого класса. Они могут быть назначены переменным, переданы как аргументы и возвращены из других функций.

- JavaScript поддерживает как именованные функции, так и анонимные функции, а также возможность определения функций внутри других функций (замыкания).

- Функции в JavaScript могут быть вызваны с помощью различных методов, включая вызов в качестве конструктора с использованием оператора "new".

4. Управление потоком выполнения:

- JavaScript предлагает различные конструкции для управления потоком выполнения кода, включая условные операторы (if-else, switch) и циклы (for, while, do-while).

- Он также поддерживает обработку исключений с использованием оператора try-catch, что позволяет перехватывать и обрабатывать ошибки и исключения.

5. Взаимодействие с HTML и CSS:

- JavaScript позволяет взаимодействовать с элементами HTML и стилями CSS на веб-странице, изменять их содержимое, атрибуты и стили.

- С помощью JavaScript можно обрабатывать события, добавлять и удалять элементы, изменять классы и анимировать элементы, что делает веб-страницы более интерактивными и динамичными.

6. Библиотеки и фреймворки:

- JavaScript имеет богатую экосистему библиотек и фреймворков, которые облегчают разработку и расширяют возможности языка.

- Некоторые популярные библиотеки и фреймворки включают React, Angular, Vue.js, Node.js, Express.js и многие другие.

- Эти инструменты предоставляют готовые решения для разработки веб-приложений, управления состоянием, маршрутизации и других задач.

JavaScript является мощным языком программирования, который предлагает широкий набор возможностей для разработки веб-приложений. Его гибкость, простота использования и широкая поддержка делают его одним из наиболее популярных языков программирования в мире. JavaScript применяется в различных сферах, включая веб-разработку, серверное программирование, разработку мобильных приложений и интернета вещей, что подчеркивает его важность и актуальность в современном мире программирования.

Консоль разработчика: Консоль разработчика веб-браузера (например, Chrome DevTools) является незаменимым инструментом для отладки JavaScript-кода. Она позволяет проверять значения переменных, исправлять ошибки, анализировать производительность и многое другое.

Сборщики модулей: Сборщики модулей, такие как webpack и Rollup, позволяют разбивать JavaScript-код на модули и объединять их в один оптимизированный файл. Это упрощает организацию и структурирование кода, а также улучшает производительность и скорость загрузки приложения.

Фреймворки и библиотеки: Существуют множество фреймворков и библиотек JavaScript, которые облегчают разработку и предоставляют готовые решения для различных задач. Некоторые из популярных фреймворков включают React.js, Angular.js, Vue.js, Express.js (для серверной разработки) и Node.js.

ECMAScript: ECMAScript (ES) - это стандарт, на основе которого разрабатывается JavaScript. Новые версии ECMAScript, такие как ES6, ES7, ES8 и т.д.вошли значительные улучшения в язык, включая новые возможности, синтаксический сахар, улучшенную поддержку модулей и другие функции. Эти обновления позволяют разработчикам писать более чистый и эффективный код.

WebAssembly: WebAssembly (Wasm) - это двоичный формат, который позволяет запускать высокопроизводительный код на веб-платформе. JavaScript может взаимодействовать с WebAssembly-модулями, предоставляя возможность создавать более мощные и быстрые веб-приложения.

Progressive Web Apps: Progressive Web Apps (PWA) объединяют возможности веб-сайтов и мобильных приложений. JavaScript играет важную роль в разработке PWA, позволяя создавать оффлайн-функциональность, отправлять уведомления и обеспечивать более плавный и реактивный пользовательский интерфейс.

### 1.4.2. Cтандартный язык разметки HTML

HTML[[6]](#_СПИСОК_ЛИТЕРАТУРЫ) (HyperText Markup Language) является стандартным языком разметки для создания веб-страниц. Он используется для описания структуры и содержания веб-документов. HTML-документ состоит из различных элементов, которые определяют типы содержимого и их отображение в браузере.

Основные компоненты HTML:

1. Элементы:

Элементы HTML представляют собой строительные блоки, используемые для разметки различных частей веб-страницы. Каждый элемент обозначается с помощью тега, который состоит из открывающего и закрывающего тегов. Между этими тегами находится содержимое элемента.

Пример элемента:

```html

<p>Это абзац текста.</p>

2. Атрибуты:

Атрибуты HTML добавляют дополнительные сведения к элементам и задают им различные свойства. Атрибуты указываются в открывающем теге элемента и содержат имя и значение, разделенные знаком равенства.

Пример использования атрибута:

```html

<a href="https://www.example.com">Это ссылка на примерный сайт</a>

3. Текстовое содержимое:

HTML позволяет размечать текстовое содержимое веб-страницы. Вы можете использовать различные теги для выделения текста, создания заголовков, создания списков и т.д.

Примеры текстового содержимого:

```html

<h1>Заголовок первого уровня</h1>

<p>Это абзац текста.</p>

<ul>

<li>Элемент списка 1</li>

<li>Элемент списка 2</li>

</ul>

4. Изображения:

HTML позволяет вставлять изображения на веб-страницу с помощью тега `<img>`. Вы можете указать источник изображения, его размеры и альтернативный текст.

Пример вставки изображения:

```html

<img src="image.jpg" alt="Описание изображения" width="300" height="200">

5. Ссылки:

HTML предоставляет тег `<a>`, который используется для создания ссылок. Вы можете указать URL-адрес ссылки и текст, который будет отображаться как ссылка.

Пример создания ссылки:

```html

<a href="https://www.example.com">Это ссылка на примерный сайт</a>

6. Таблицы:

HTML позволяет создавать таблицы для представления структурированных данных. Вы можете использовать теги `<table>`, `<tr>` и `<td>` для определения структуры таблицы и размещения данных в ячейках.

Пример создания таблицы:

```html

<table>

<tr>

<td>Ячейка 1</td>

<td>Ячейка 2</td>

</tr>

<tr>

<td>Ячейка 3</td>

<td>Ячейка 4</td>

</tr>

</table>

Это всего лишь обзор основных компонентов HTML. Язык HTML имеет богатый набор тегов и атрибутов, которые позволяют создавать разнообразные веб-страницы с различным содержимым и взаимодействием. Это язык разметки, который является основой для создания веб-страниц и веб-приложений.

### 1.4.3 Язык программирования CSS

CSS (Cascading Style Sheets) [[8]](#_СПИСОК_ЛИТЕРАТУРЫ)- это язык стилей, используемый для определения внешнего вида и форматирования веб-страниц. Он позволяет разработчикам управлять представлением элементов HTML, определяя их цвет, размеры, шрифты, расположение и другие визуальные атрибуты. Вот подробное описание основных аспектов языка CSS:

1. Синтаксис CSS:

- CSS использует селекторы для выбора элементов HTML, к которым будет применяться стиль.

- Стили в CSS определяются через свойства и значения.

- Каждое правило CSS состоит из селектора, за которым следует блок с объявлениями стилей.

2. Селекторы:

- Тип селектора: выбирает все элементы определенного типа. Например, селектор `p` выбирает все абзацы на странице.

- Классовый селектор: выбирает все элементы с определенным классом. Например, селектор `.highlight` выбирает все элементы с классом "highlight".

- Идентификаторный селектор: выбирает элемент с определенным идентификатором. Например, селектор `#header` выбирает элемент с идентификатором "header".

- Комбинированные селекторы: позволяют комбинировать различные условия для выбора элементов. Например, селектор `ul li` выбирает все элементы `<li>`, которые находятся внутри `<ul>`.

3. Свойства и значения:

- CSS предоставляет широкий набор свойств, которые позволяют управлять внешним видом элементов.

- Свойства определяют различные атрибуты элементов, такие как цвет, размеры, шрифты, отступы, границы и фон.

- Каждое свойство имеет соответствующее значение, которое определяет, каким образом свойство будет применяться к элементу.

CSS является важным инструментом для создания привлекательных и функциональных веб-страниц. Он позволяет разработчикам полностью контролировать внешний вид элементов, создавать современные дизайны и обеспечивать адаптивность для различных устройств.

### 1.4.4 Язык программирования React

Функциональные требования [[11]](#_СПИСОК_ЛИТЕРАТУРЫ)к CRM-системе для интернет-магазина этно-бренда “Aisanat”, с учетом использования Next.js для разработки, направлены на обеспечение эффективного управления отношениями с клиентами и повышения уровня удовлетворенности клиентов. Вот переформулированное описание функциональных требований:

1.Управление контактами и лидами:

Система должна предоставлять возможность создания, редактирования и удаления контактных данных клиентов.

Должна быть реализована система управления лидами, позволяющая отслеживать их жизненный цикл от первого контакта до продажи.

Необходимо обеспечить возможность классификации и категоризации клиентов для удобного поиска и организации.

2.Управление продажами и обслуживанием клиентов:

Система должна поддерживать процесс отслеживания продаж, включая создание и управление заказами, отслеживание статуса выполнения заказов и управление возвратами.

Должна быть реализована система обслуживания клиентов, позволяющая обрабатывать запросы и жалобы клиентов, а также предоставлять поддержку через различные каналы связи.

3.Маркетинг и аналитика:

Система должна предоставлять инструменты для проведения маркетинговых кампаний, включая управление кампаниями, отслеживание результатов и анализ поведения клиентов.

Необходимо обеспечить возможность анализа данных о продажах, клиентском поведении и взаимодействии с клиентами для выявления трендов и оптимизации маркетинговых стратегий.

4.Безопасность и конфиденциальность данных:

Система должна обеспечивать безопасность и конфиденциальность данных клиентов, включая аутентификацию пользователей, контроль доступа и шифрование данных.

Должны быть применены меры защиты от несанкционированного доступа и утечки данных.

5.Интеграция с другими системами:

CRM-система должна иметь возможность интеграции с другими информационными системами предприятия, такими как системы управления контентом (CMS), системы электронной коммерции (e-commerce) и системы управления проектами (PM).

Необходимо обеспечить возможность обмена данными между системами для автоматического обновления информации и предотвращения дублирования данных.

6.Отчетность и аналитика:

Система должна предоставлять возможность генерации различных отчетов и аналитических данных по продажам, клиентскому поведению и взаимодействию с клиентами, таких как статистика по количеству продаж, срокам исполнения и другим параметрам.

Пользователи должны иметь доступ к настраиваемым панелям управления и инструментам для анализа данных.

7.Интерфейс пользователя:

Интерфейс CRM-системы должен быть интуитивно понятным и привлекательным, обеспечивая удобство использования, легкость навигации и доступ к основным функциям системы.

Важно обеспечить адаптивность системы для работы на различных устройствах и платформах, включая мобильные устройства и планшеты.

8.Масштабируемость и производительность:

CRM-система должна быть способна обрабатывать большой объем данных и поддерживать одновременную работу нескольких пользователей.

Необходимо обеспечить высокую производительность системы и масштабируемость для возможности расширения функциональности и обработки растущей нагрузки.

9.Архивирование и сохранность данных:

Система должна обеспечивать архивирование данных и сохранность информации в соответствии с требованиями законодательства и внутренними политиками предприятия.

Важно предусмотреть резервное копирование данных и механизмы восстановления системы в случае сбоев или потери информации.

10.Поддержка и обучение:

После внедрения CRM-системы необходимо предоставить обучение для сотрудников, чтобы они могли эффективно использовать систему и получать максимальную пользу от нее.

Необходима поддержка пользователей и техническая поддержка для решения возникающих проблем и вопросов.

Использование Next.js для разработки CRM-системы позволяет создать высокопроизводительное и адаптивное веб-приложение с серверным рендерингом и статической генерацией страниц, что обеспечивает лучшую производительность и SEO. Next.js также поддерживает API-маршруты, что упрощает создание API для интеграции с другими системами и обеспечивает легкую разработку и поддержку/

### 1.4.5 Язык структурированных запросов SQД

Язык программирования SQL[[14]](#_СПИСОК_ЛИТЕРАТУРЫ) (Structured Query Language) является стандартизированным языком запросов, разработанным для работы с реляционными базами данных. Он обеспечивает возможность создания, изменения и управления данными в базах данных.

SQL состоит из различных компонентов и конструкций, позволяющих выполнять различные операции с данными. Вот основные компоненты SQL:

1. Команды DDL (Data Definition Language):

- `CREATE DATABASE`: Создает новую базу данных.

- `CREATE TABLE`: Создает новую таблицу в базе данных.

- `ALTER TABLE`: Изменяет структуру существующей таблицы.

- `DROP TABLE`: Удаляет таблицу из базы данных.

2. Команды DML (Data Manipulation Language):

- `INSERT INTO`: Вставляет новые данные в таблицу.

- `UPDATE`: Обновляет существующие данные в таблице.

- `DELETE FROM`: Удаляет данные из таблицы.

- `SELECT`: Извлекает данные из таблицы.

3. Команды DQL (Data Query Language):

- `SELECT`: Извлекает данные из одной или нескольких таблиц.

- `FROM`: Определяет, из какой таблицы извлекать данные.

- `WHERE`: Устанавливает условия для фильтрации данных.

- `GROUP BY`: Группирует данные по определенным столбцам.

- `HAVING`: Фильтрует данные, основываясь на агрегатных функциях.

- `ORDER BY`: Сортирует данные по указанным столбцам.

- `JOIN`: Объединяет данные из нескольких таблиц.

4. Команды DCL (Data Control Language):

- `GRANT`: Предоставляет определенным пользователям или ролям права доступа к базе данных или таблице.

- `REVOKE`: Отзывает права доступа у пользователей или ролей.

- `COMMIT`: Фиксирует изменения, внесенные в базу данных.

- `ROLLBACK`: Откатывает изменения, внесенные в базу данных.

- `SAVEPOINT`: Устанавливает точку сохранения для возможности отката изменений к этой точке.

5. Команды TCL (Transaction Control Language):

- `BEGIN TRANSACTION`: Начинает новую транзакцию.

- `COMMIT`: Фиксирует изменения, внесенные в базу данных в рамках текущей транзакции.

- `ROLLBACK`: Откатывает изменения, внесенные в базу данных в рамках текущей транзакции.

SQL также поддерживает использование различных операторов и функций для обработки данных:

- Операторы: `AND`, `OR`, `NOT`, `LIKE`, `IN`, `BETWEEN`, `IS NULL`, `UNION`, `INTERSECT`, `EXCEPT` и другие.

- Функции: `COUNT`, `SUM`, `AVG`, `MAX`, `MIN`, `UPPER`, `LOWER`, `CONCAT`, `SUBSTRING`, `DATE`, `TIME` и другие.

SQL позволяет создавать индексы для оптимизации производительности запросов, а также использовать представления (views), хранимые процедуры (stored procedures) и триггеры (triggers) для упрощения и автоматизации операций с базой данных.

Важно отметить, что SQL является декларативным языком, что означает, что вы описываете результат, который хотите получить, а не указываете конкретные шаги для его достижения. SQL-запросы могут быть сложными и мощными, позволяя работать с большим объемом данных и выполнять сложные операции.

SQL широко применяется во множестве областей, включая веб-разработку, анализ данных, бизнес-интеллект и администрирование баз данных. Он является одним из самых популярных языков программирования в области работы с базами данных, и знание SQL является важным навыком для разработчиков, аналитиков и администраторов баз данных.

### 1.4.6 Система для управления базами данных PostgreSQL

Язык программирования PostgreSQL, часто называемый PL/pgSQL (Procedural Language/PostgreSQL), представляет собой процедурный язык, разработанный специально для работы с базой данных PostgreSQL. PL/pgSQL предоставляет расширенные возможности для создания хранимых процедур, функций и триггеров, позволяя разработчикам эффективно управлять данными и выполнением запросов на сервере PostgreSQL.

Основная цель PL/pgSQL - предоставить разработчикам удобный и мощный инструментарий для написания процедурных элементов базы данных. PL/pgSQL имеет синтаксис, схожий с языком программирования SQL, что облегчает его освоение разработчиками, знакомыми с SQL.

Процедуры, функции и триггеры, написанные на PL/pgSQL, компилируются и выполняются на сервере PostgreSQL, что обеспечивает высокую производительность и эффективность при работе с данными. Компиляция PL/pgSQL происходит во время создания объекта базы данных и сохраняет скомпилированный код для последующего использования.

Еще одним важным аспектом PL/pgSQL является его интеграция с базовым языком SQL. Разработчики могут включать SQL-код непосредственно в свои процедуры и функции, что облегчает выполнение сложных запросов и манипуляцию данными внутри процедурных элементов.

Благодаря широким возможностям PL/pgSQL разработчики могут создавать сложные бизнес-логики на стороне сервера, выполнять вычисления и преобразования данных, обрабатывать исключения и выполнять множество других операций непосредственно на базе данных PostgreSQL.

В целом, PL/pgSQL является мощным языком программирования, который позволяет разработчикам полностью использовать возможности базы данных PostgreSQL. Он обеспечивает высокую производительность, гибкость и удобство в написании процедурных элементов, делая PostgreSQL одной из наиболее распространенных и популярных систем управления базами данных.

## 1.5 Обоснование среды разработки

WebStorm по критериям технического, программного и технологического обеспечения

### 1.5.1 Техническое обеспечение

Техническое обеспечение WebStorm включает требования к аппаратному обеспечению и операционной системе, на которой может работать это IDE (Integrated Development Environment). Основные аспекты технического обеспечения:

- Процессор: Минимум 2 ГГц (рекомендуется многоядерный процессор для лучшей производительности).

- Оперативная память: Минимум 4 ГБ (рекомендуется 8 ГБ и более для работы с большими проектами).

- Место на диске: Минимум 2,5 ГБ свободного места (включая место для кэша и временных файлов).

- Дисплей: Разрешение минимум 1024x768 (рекомендуется Full HD для комфортной работы).

- Операционные системы: Поддерживаются Windows 7/8/10/11 (64-бит), macOS 10.13 и новее, Linux (современные дистрибутивы с поддержкой Gnome или KDE).

### 1.5.2. Программное обеспечение

WebStorm предоставляет мощный набор инструментов и функциональных возможностей для разработки, ориентированных на веб-разработку. Ключевые аспекты программного обеспечения:

- Редактор кода: Поддержка HTML, CSS, JavaScript, TypeScript и других веб-технологий. Включает подсветку синтаксиса, автодополнение кода, и функции рефакторинга.

- Интеграция с VCS: Поддержка Git, Mercurial, SVN и других систем контроля версий. Включает удобный интерфейс для управления изменениями и слияниями.

- Отладка: Мощные инструменты отладки для JavaScript, Node.js и других платформ, включая точечные останова, пошаговое выполнение кода, и инспекцию переменных.

- Тестирование: Интеграция с популярными тестовыми фреймворками, такими как Jest, Mocha, Karma, и Jasmine.

- Пакетные менеджеры: Поддержка npm, Yarn, и других пакетных менеджеров для удобного управления зависимостями проектов.

- Инструменты для разработки: Встроенная поддержка для Docker, Webpack, и других современных инструментов и технологий.

### 1.5.3. Технологическое обеспечение

Технологическое обеспечение WebStorm связано с его возможностями для интеграции и поддержки современных технологий и стандартов веб-разработки:

- Поддержка современных фреймворков: WebStorm поддерживает Angular, React, Vue.js и другие популярные фреймворки, предоставляя специальные инструменты и плагины для повышения продуктивности.

- Кросс-платформенность: IDE работает на всех основных операционных системах (Windows, macOS, Linux), что обеспечивает удобство использования для разработчиков с разными предпочтениями.

-Расширяемость: WebStorm поддерживает плагины и позволяет разработчикам создавать свои собственные расширения для улучшения функциональности IDE.

- Облачные возможности: Поддержка облачных сервисов и инструментов для разработки, включая интеграцию с CI/CD пайплайнами и возможность работы с удаленными репозиториями.

- Обновления и поддержка: Регулярные обновления от компании JetBrains обеспечивают актуальность IDE, включая поддержку новых стандартов, языков и фреймворков.

Таким образом, WebStorm предоставляет все необходимое для комфортной и эффективной разработки веб-приложений, сочетая современные технологии с мощными инструментами и широкими возможностями настройки.

# Глава 2 Проектная часть

## 2.1 Концептуальная модель информационных потоков данных

Создание информационной системы обеспечения перевозки товаров между странами для ОсОО «Сэм экспресс» является важным проектом, направленным на оптимизацию управления перевозками и повышение эффективности бизнес-процессов. Основные компоненты концептуальной модели информационных потоков включают следующее:

1. Сбор информации о грузах и клиентах: Система получает запросы и данные о грузах от клиентов, включая детали перевозки и контактную информацию.

2. Регистрация и обработка заявок: Процесс включает в себя регистрацию новых клиентов и заявок на перевозку в системе, а также обработку этих заявок в соответствии с предварительно собранной информацией.

3. Маршрутизация и управление заявками: Система определяет оптимальные маршруты для перевозки и управляет заявками, направляя их на обработку соответствующим сотрудникам или подразделениям.

4. Обработка заявок: Сотрудники компании приступают к обработке заявок, координируя процесс перевозки и обеспечивая необходимые операции по управлению грузами.

5. Обновление статуса заявок: В процессе перевозки сотрудники систематически обновляют информацию о текущем статусе грузов и выполненных действиях.

6. Уведомления и взаимодействие: Система информирует клиентов о статусе их грузов через уведомления и обеспечивает возможность взаимодействия сотрудников и клиентов в рамках системы.

7. Анализ и отчетность: Система собирает данные для анализа процесса перевозок и создания отчетов, предоставляя руководству информацию о работе компании.

Выбор SQL и Node.js для разработки данной системы обеспечивает надежное хранение данных и эффективную серверную логику, что создает прочную основу для ее функционирования. Перед началом программной реализации необходимо создать модель приложения, которая может быть представлена в виде диаграмм на языке UML.

## 2.2 Функциональная модель информационных потоков данных

При разработке информационной системы для обеспечения перевозки товаров между странами ОсОО «Сэм экспресс», диаграмма вариантов использования становится ключевым инструментом для ясного представления функциональности системы. Этот графический образ позволяет наглядно отобразить взаимодействие между участниками (актерами) и самой системой в рамках различных сценариев использования. На диаграмме отражаются разнообразные действия, которые могут выполнять пользователи, а также их взаимодействие с системой.

Применительно к созданию информационной системы для перевозки товаров между странами ОсОО «Сэм экспресс», диаграмма вариантов использования будет служить ключевым инструментом для наглядного анализа того, как пользователи будут взаимодействовать с системой в различных ситуациях, таких как оформление заказа, отслеживание грузов, обработка таможенной документации и т. д. Это не только важный инструмент для визуализации функциональности системы, но и для последующего анализа и проектирования информационной системы для перевозки товаров между странами ОсОО «Сэм экспресс».

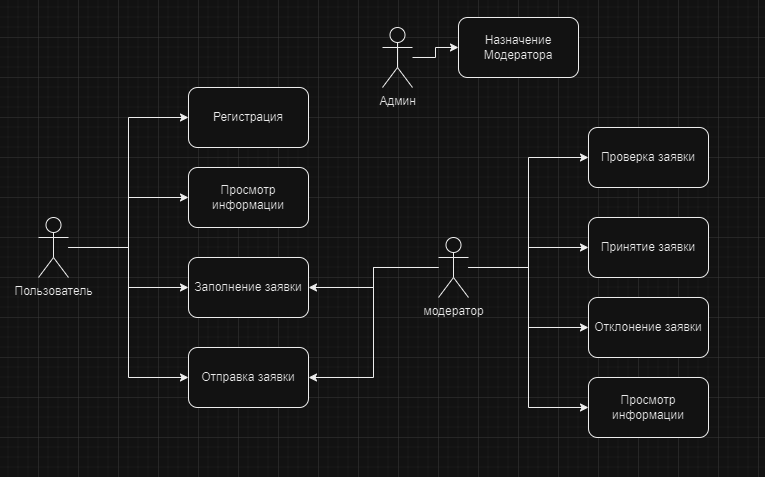


Рисунок 1 — Диаграмма вариантов использования ИС. [[19]](#_СПИСОК_ЛИТЕРАТУРЫ)

## 2.3 Логическая схема взаимодействия отдельных частей БД

Диаграмма состояний является визуальным представлением функционирования информационной системы обеспечения перевозки товаров между странами ОсОО «Сэм экспресс». Она отображает различные состояния, в которых может находиться система, а также демонстрирует переходы между этими состояниями в ответ на различные события или внутренние условия.

Эта диаграмма позволяет наглядно анализировать, как система реагирует на различные события и условия, а также понять поток обработки товаров и изменение состояний системы со временем. Этот инструмент полезен для проектирования и оптимизации информационной системы ОсОО «Сэм экспресс», обеспечивая эффективное управление перевозками и повышение качества обслуживания клиентов.

Диаграмма состояний также упрощает процесс анализа и оптимизации работы информационной системы, позволяя выявить возможные узкие места и улучшить процессы перевозки товаров. Она помогает не только понять текущее состояние системы, но и предвидеть её поведение в ответ на различные сценарии использования, что делает её важным инструментом как на этапе разработки, так и в процессе эксплуатации системы.

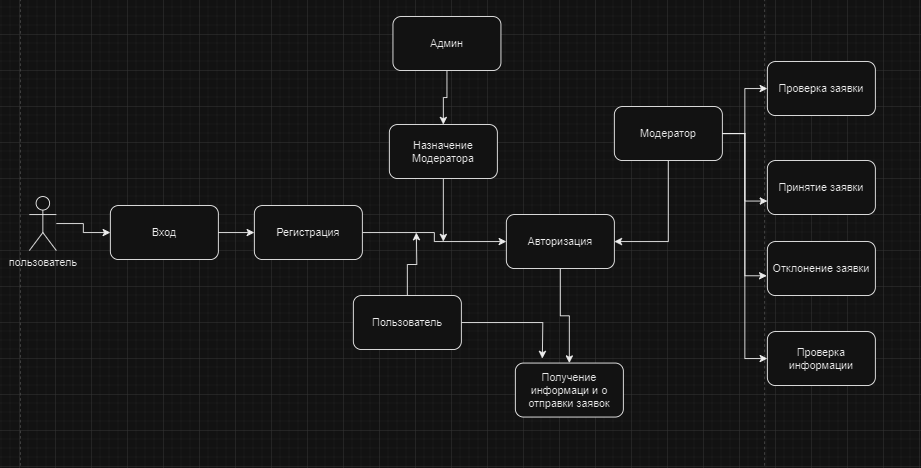


Рисунок 2 — Диаграмма состояний. [[18]](#_СПИСОК_ЛИТЕРАТУРЫ)

## 2.4 Физическая схема взаимодействия отдельных частей БД

ER-диаграмма является ключевым инструментом при разработке информационной системы для перевозки товаров между странами в компании ОсОО «Сэм экспресс». Она применяется для визуализации структуры базы данных, выявления сущностей, их атрибутов и взаимосвязей.

На данной диаграмме будут представлены следующие компоненты:

1. Заказы: Элементы, которые являются основными объектами в системе и содержат информацию о заказах от клиентов.

2. Клиенты: Лица, которые размещают заказы и взаимодействуют с информационной системой.

3. Товары: Классификация товаров, которые представлены в информационной системе.

4. Платежи: Информация о проведенных оплатах заказов.

5. Доставка: Детали о процессе доставки заказов клиентам.

6. Другие компоненты: Возможно, дополнительные сущности, такие как категории товаров, сотрудники, статусы заказов и прочее.

Взаимосвязи между данными сущностями определяются на основе процессов и логики работы информационной системы. Например, заказ может быть связан с определенным клиентом, определенным товаром и его статусом. Также могут быть дополнительные атрибуты, описывающие связи между сущностями.

ER-диаграмма предоставляет наглядное представление о структуре данных в системе, что помогает разработчикам и аналитикам лучше понять взаимосвязи между компонентами информационной системы для перевозки товаров. Этот инструмент является необходимым для проектирования и оптимизации базы данных, что критически важно для эффективного функционирования системы.

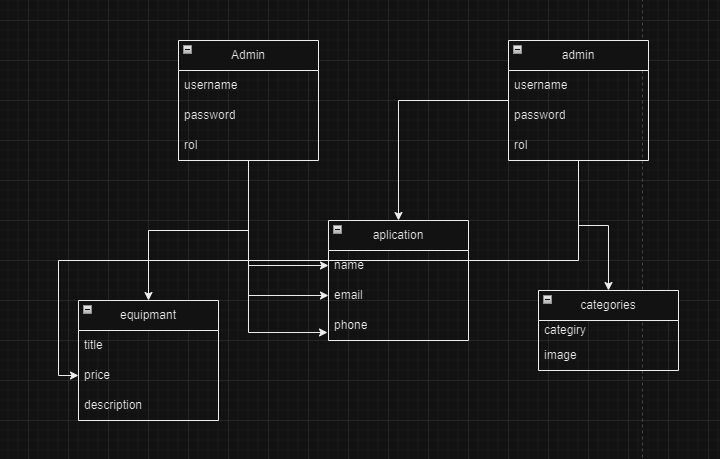


Рисунок 3 — Диаграмма сущностей.

## 2.5 Функциональная модель

На основе данных выше была составлена контекстная диаграмма, показанная на рисунке 4.

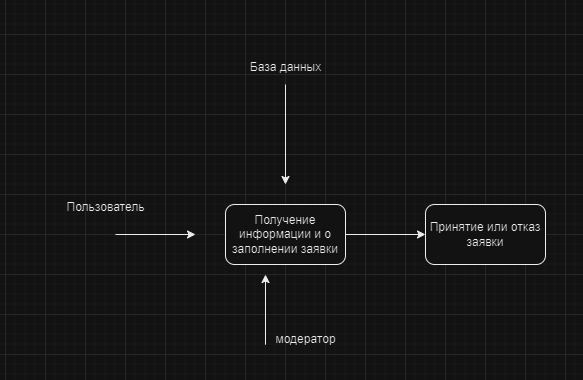


Рисунок 4 — Контекстная диаграмма IDEF0. [[19]](#_СПИСОК_ЛИТЕРАТУРЫ)

## 2.6 Схема технологического процесса

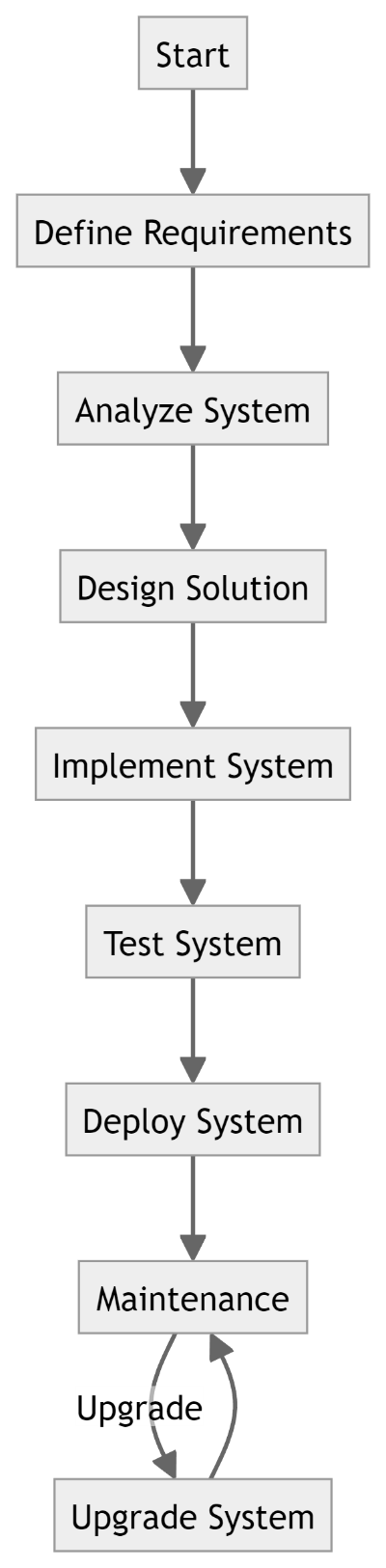


Рисунок 5 — Контекстная диаграмма IDEF1.

На схеме IDEF1 изображена информационная система обеспечения перевозки товаров между странами для ОсОО «Сэм экспресс». Эта схема включает следующие элементы и их взаимосвязи:

SHIPPER (Отправитель): Представляет клиента, отправляющего товары через систему.

SHIPMENT (Отправление): Представляет сам отправленный груз. Связан с отправителем и содержит информацию о перевозимых товарах.

PRODUCT (Товар): Отражает товары, которые перевозятся через систему. Каждый товар связан с определенным отправлением и относится к определенной категории.

CATEGORY (Категория): Представляет категорию товаров, включающую различные подкатегории.

SUBCATEGORY (Подкатегория): Отображает подкатегорию товаров, которая входит в определенную категорию.

Таким образом, схема показывает взаимосвязи между отправителями, отправлениями, товарами, категориями и подкатегориями в информационной системе обеспечения перевозки товаров между странами ОсОО «Сэм экспресс».

## 2.7 Схема взаимосвязи программных модулей

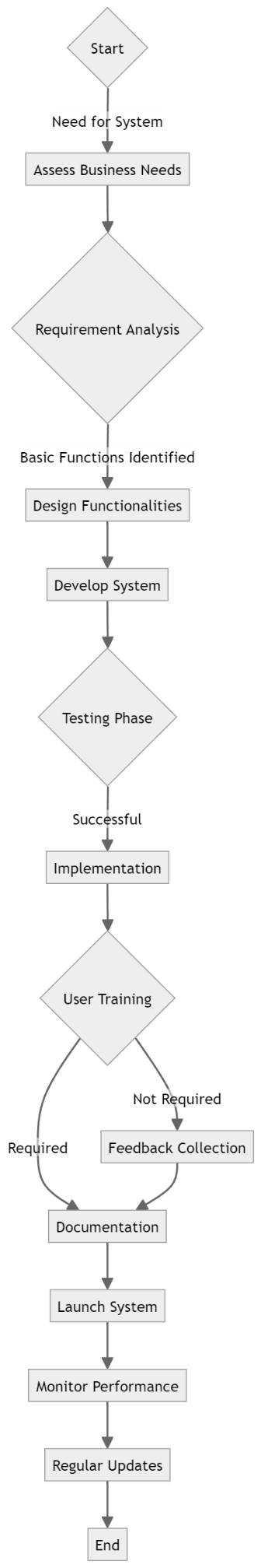


Рисунок 6 — Контекстная диаграмма IDEF2.

Этот чертеж IDEF2 описывает информационную систему, управляющую перевозкой товаров между странами в компании ОсОО «Сэм экспресс». В нем изображены следующие элементы и их взаимосвязи:

- `Клиент` (Customer): Устанавливает связь "один-ко-многим" с `Заказом` (Order), что означает, что один клиент может делать несколько заказов, но каждый заказ связан только с одним клиентом.

- `Заказ` (Order): Связан с `Товаром` (Item), `Оплатой` (Payment), `Инвентарем` (Inventory) и `Техническим отделом` (Technical Department). Заказ включает в себя один или несколько товаров, оплату, используемый инвентарь и информацию для технического отдела.

- У каждой сущности есть атрибуты, описывающие их характеристики, такие как идентификаторы, названия, даты и другие.

Этот чертеж представляет структуру данных и связи между различными элементами в информационной системе обеспечения перевозки товаров между странами в компании ОсОО «Сэм экспресс». Он может быть использован для проектирования и разработки такой системы, а также для понимания ее основных компонентов и взаимосвязей.

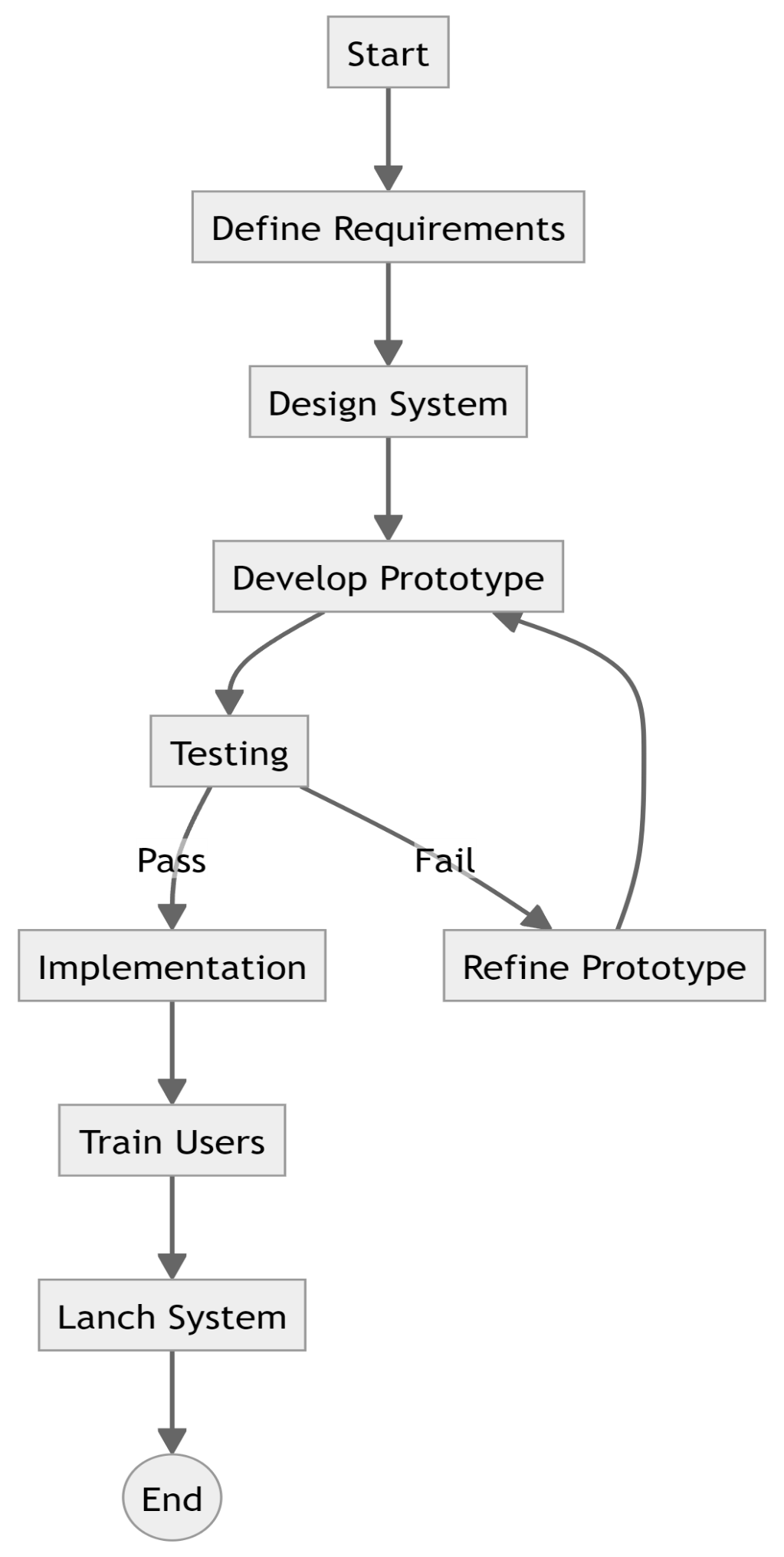


Рисунок 7 — Контекстная диаграмма IDEF3.

Диаграмма контекста (Context Diagram) является инструментом высокого уровня, который иллюстрирует информационную систему в контексте ее взаимодействия с внешними сущностями. Она позволяет визуализировать связи системы с внешними актерами, другими системами или окружающей средой.

В рамках разработки информационной системы для перевозки товаров между странами ОсОО «Сэм экспресс», контекстная диаграмма будет отображать процесс перевозки и взаимодействие с внешними актерами или системами. Этот инструмент поможет понять роль системы в более широком контексте и выделить основные внешние сущности, с которыми она взаимодействует.

## 2.6 Подробное описание проектирования и разработки приложения

Проектирование и разработка информационной системы обеспечения перевозки товаров между странами для ОсОО «Сэм экспресс» является ключевым этапом в оптимизации процессов перевозки и управления бизнесом. Давайте рассмотрим этот процесс более детально.

1. Анализ потребностей

В начале процесса проводится анализ требований клиентов и основных бизнес-процессов компании. Этот этап включает в себя:

- Определение текущих проблем в существующих процессах перевозки товаров.

- Выявление ожиданий и требований персонала компании к новой информационной системе.

- Идентификацию необходимых функциональностей для обеспечения эффективной перевозки.

- Планирование интеграции с уже существующими информационными системами компании.

2. Проектирование архитектуры

На этом этапе определяется структура и компоненты будущей информационной системы. Ключевые аспекты включают:

- Интерфейс пользователя для управления заказами и отслеживания перевозок.

- Механизм аутентификации и авторизации пользователей для обеспечения безопасного доступа к системе.

- Система управления данными о заказах и перевозках.

- Модуль уведомлений для оперативного информирования о статусах заказов и изменениях.

- База данных для хранения информации о клиентах и их заказах.

- Возможность интеграции с другими системами через API для обмена данными.

3. Разработка

На этом этапе осуществляется фактическая реализация информационной системы, включая:

- Создание удобного пользовательского интерфейса, учитывающего потребности пользователей.

- Настройка базы данных для эффективного хранения информации о заказах и перевозках.

- Разработка бизнес-логики для обработки заказов и управления данными.

- Создание модуля уведомлений для оперативного информирования клиентов.

- Тестирование каждого компонента системы для выявления и исправления ошибок.

4. Внедрение и обучение

После завершения разработки системы происходит ее внедрение в рабочую среду компании. Этот процесс включает:

- Установку и настройку необходимого оборудования и программного обеспечения.

- Обучение сотрудников компании работе с новой информационной системой.

5. Поддержка и сопровождение

Важным шагом является последующая поддержка и развитие системы, включая:

- Оказание поддержки пользователям и решение возникающих проблем.

- Внесение изменений и улучшений в систему на основе обратной связи от пользователей.

- Регулярное обновление системы для обеспечения безопасности и функциональности.

Это общий обзор процесса проектирования и разработки информационной системы для компании ОсОО «Сэм экспресс». Каждый этап требует внимательного планирования и выполнения, а также сотрудничества с заинтересованными сторонами для успешного внедрения и использования системы.

# Глава 3 Разработка документации

## 3.1 Руководство пользователю

Пользовательский путь для веб-приложения информационной системы обеспечения перевозки товаров между странами ОсОО «Сэм экспресс»:

1. Вход в систему:

Пользователь аутентифицируется через логин и пароль, одноразовые пароли или биометрические методы.

После успешной аутентификации открывается персонализированная главная страница.

2. Главная страница:

Пользователь получает обзор системы и персональные уведомления.

Также отображается список задач и проектов, требующих внимания.

3. Управление контактами и клиентским взаимодействием:

Пользователь может просматривать и редактировать информацию о контактах, управлять взаимодействием через различные каналы связи.

Возможности включают создание новых контактов, добавление заметок, назначение задач и отслеживание взаимодействия.

4. Уведомления и напоминания:

Пользователь получает уведомления о новых сообщениях, задачах и других важных событиях.

Можно настроить предпочтения по типу и частоте уведомлений.

5. Поиск и фильтрация:

Возможность поиска и фильтрации контактов и задач по различным параметрам, таким как статус, дата создания и категория.

6. Аналитика и отчетность:

Получение отчетов о клиентском взаимодействии, продажах и маркетинговых кампаниях.

7. Настройки и профиль пользователя:

Пользователь может настроить уведомления, язык интерфейса, а также управлять профилем и учетными данными.

8. Поддержка и обратная связь:

Возможность обращения за помощью и обратной связи с администраторами системы.

Дополнительные шаги в пользовательском пути:

9. Коммуникация и совместная работа:

Пользователи могут обмениваться сообщениями, документами и обсуждать задачи в реальном времени.

10. Интеграция с другими системами:

Возможность интеграции с внешними системами для синхронизации данных и автоматизации процессов.

11. Мобильное приложение:

Разработка мобильного приложения для доступа к системе с любого устройства.

12. Обучение и поддержка:

Предоставление обучающих материалов и технической поддержки для пользователей.

13. Безопасность и защита данных:

Обеспечение безопасности данных и конфиденциальности информации пользователей.

14. Аудит и контроль доступа:

Ведение журналов аудита и контроль доступа к данным.

15. Непрерывное улучшение:

Разработка новых функций и улучшений на основе обратной связи пользователей и изменений в бизнес-процессах.

16. Определение бизнес-потребностей:

Понимание конкретных потребностей бизнеса в управлении данными и процессами перевозки товаров между странами.

17. Исследование рынка программ:

Оценка различных информационных систем на рынке, анализ их функционала, отзывов пользователей, цен и возможностей интеграции.

18. Учет масштаба бизнеса:

Выбор информационной системы, соответствующей размеру компании и планам по расширению.

19. Консультация с экспертами:

Получение профессиональных рекомендаций по выбору информационной системы от IT специалистов или консультантов.

20. Безопасность данных:

Обеспечение высокого уровня защиты конфиденциальной информации и данных о перевозках товаров.

21. Поддержка клиентов:

Уверенность в доступности качественной технической поддержки для оперативного решения вопросов и проблем пользователей.

22. Интеграция с другими инструментами:

Проверка возможности интеграции информационной системы с другими программными продуктами, используемыми в бизнесе перевозок.

23. Обучение персонала:

Организация обучения сотрудников по работе с информационной системой для максимального использования её функционала.

24. Стоимость реализации и обслуживания:

Расчет всех затрат на внедрение информационной системы, а также стоимости обновлений и технической поддержки.

25. Тестирование перед окончательным выбором:

Проведение тестирования нескольких информационных систем на практике для оценки их соответствия бизнес-требованиям и удобства использования.

Эти шаги помогут не только в выборе подходящей информационной системы для ОсОО «Сэм экспресс», но и в успешном внедрении и использовании системы для оптимизации процессов перевозки товаров между странами.

### 3.1.1 Требования к программному и аппаратному обеспечению

При разработке информационной системы для обеспечения перевозки товаров между странами ОсОО «Сэм экспресс» необходимо учесть следующие требования к программному и аппаратному обеспечению.

1. Программное обеспечение:

Разработка пользовательского интерфейса: Использование React и Redux для создания интерфейса с поддержкой динамической загрузки данных и управления состоянием.

Управление базой данных: Применение SQL (например, PostgreSQL) для хранения информации о перевозках, клиентах и других связанных данных.

Разработка серверной части: Использование Node.js или других языков программирования для обработки запросов, работы с базой данных и выполнения бизнес-логики.

Аутентификация и авторизация: Внедрение механизмов для безопасной аутентификации и управления доступом пользователей.

2. Аппаратное обеспечение:

Серверы: Поддержка выделенных серверов или облачных ресурсов для обработки данных и хранения информации о перевозках.

Клиентские устройства: Обеспечение совместимости с различными устройствами, такими как компьютеры, планшеты и мобильные телефоны.

3. Сетевая инфраструктура:

Стабильное сетевое соединение: Гарантирование высокоскоростного соединения между клиентами и серверами.

Защита данных: Применение шифрования и протоколов безопасности для защиты конфиденциальности информации.

4. Масштабируемость:

Горизонтальное масштабирование: Возможность расширения системы для учета увеличения объема перевозок и клиентов.

Высокая доступность: Использование кластеров серверов для обеспечения непрерывной работы системы.

5. Интеграция:

Взаимодействие с другими системами: Поддержка интеграции с другими системами для эффективного согласования процессов.

6. Удобство использования:

Понятный интерфейс: Разработка интерфейса, который будет интуитивно понятен пользователям и соответствует потребностям компании «Сэм экспресс».

Управление заказами: Обеспечение возможности управления заказами, отслеживания статусов, получения уведомлений и формирования отчетов.

## 3.2 Руководство программисту

### 3.2.1 Проектирование базы данных

Структура информационной системы обеспечения перевозки товаров между странами для ОсОО «Сэм экспресс»:

1. База данных:

- Таблица "Заказы":

- `id` (INT): уникальный идентификатор заказа (Primary Key).

- `наименование` (VARCHAR): наименование заказа.

- `описание` (TEXT): описание заказа.

- `статус` (VARCHAR): текущий статус заказа (например, "новый", "в обработке", "завершен").

- `ответственный` (INT): идентификатор сотрудника, ответственного за выполнение заказа.

- Таблица "Сотрудники":

- `id` (INT): уникальный идентификатор сотрудника (Primary Key).

- `имя` (VARCHAR): имя сотрудника.

- `должность` (VARCHAR): должность сотрудника.

- `email` (VARCHAR): адрес электронной почты сотрудника.

2. Структура папок для проекта на React:

- `src`

- `components`

- `Заказы`

- `СписокЗаказов.js`

- `ДеталиЗаказа.js`

- `ФормаЗаказа.js`

- `Сотрудники`

- `СписокСотрудников.js`

- `ДеталиСотрудника.js`

- `ФормаСотрудника.js`

- `services`

- `api.js`

- `заказы.js`

- `сотрудники.js`

- `App.js`

- `index.js`

Эта структура предусматривает базу данных для хранения информации о заказах и сотрудниках, где каждая таблица имеет свои поля для хранения соответствующих данных. Структура папок для проекта на React организована таким образом, чтобы компоненты были разделены по функциональности. Каждая папка компонента содержит файлы, отвечающие за отображение списка, деталей и формы соответствующей сущности. Папка `services` содержит файлы для взаимодействия с сервером (API) и обработки запросов к заказам и сотрудникам.

### 3.2.3 Развертывание системы

Для успешного создания информационной системы обеспечения перевозки товаров между странами ОсОО «Сэм экспресс» необходимо последовательно выполнить следующие этапы:

1. Анализ потребностей и планирование:

Провести тщательный анализ требований к информационной системе, учитывая специфику перевозки товаров между странами и ожидания клиентов.

Разработать детальный план действий, определяющий этапы, ресурсы и временные рамки проекта.

2. Выбор платформы и функционала:

Определить наилучшую информационную платформу и необходимый функционал, учитывая потребности ОсОО «Сэм экспресс» и особенности перевозки товаров между странами.

3. Установка и настройка информационной системы:

Установить и сконфигурировать выбранное программное обеспечение, включая базы данных и пользовательские интерфейсы.

Обеспечить интеграцию с другими системами, если это необходимо.

4. Разработка и тестирование:

Разработать информационную систему, включая базу данных клиентов, систему управления заказами и отчетность.

Провести тестирование системы на соответствие требованиям и исправить выявленные ошибки.

5. Обучение персонала:

Организовать обучение сотрудников ОсОО «Сэм экспресс» по использованию новой информационной системы.

Предоставить поддержку и консультации персоналу в процессе внедрения и использования системы.

6. Развертывание и внедрение:

Осуществить развертывание информационной системы в компании, следуя разработанному плану.

Провести пилотное внедрение и собрать обратную связь от пользователей для дальнейших улучшений.

7. Мониторинг и поддержка:

Настроить систему мониторинга работы информационной системы для оперативного выявления и устранения возможных проблем.

Обеспечить техническую поддержку пользователей и регулярно обновлять систему с учетом потребностей и обратной связи клиентов.

Каждый из этих этапов включает в себя ряд действий и задач, необходимых для успешного создания информационной системы обеспечения перевозки товаров между странами ОсОО «Сэм экспресс».

## 3.3. Тестирование системы

Переформулированный текст для темы "Информационная система обеспечения перевозки товаров между странами ОсОО «Сэм экспресс»":

Тестирование информационной системы, обеспечивающей перевозку товаров между странами в рамках ОсОО «Сэм экспресс», представляет собой ключевой этап для обеспечения ее качества и надежности. Этот процесс включает разнообразные виды проверок, направленных на оценку соответствия системы требованиям и ее бесперебойную работу.

Основные направления тестирования:

1. Функциональное тестирование:

Проверка функциональности управления заказами: тестирование создания, изменения, отслеживания и закрытия заказов.

Управление клиентской базой: тестирование операций добавления, редактирования, удаления и поиска информации о клиентах.

Мониторинг взаимодействий: проверка записи и отслеживания контактов с клиентами, включая телефонные звонки, встречи и электронную почту.

Классификация клиентов: тестирование возможностей сегментации клиентской базы по различным критериям (местоположение, предпочтения, история покупок).

2. Нефункциональное тестирование:

Оценка производительности: проверка времени отклика системы и ее стабильности при различных нагрузках.

Проверка надежности: тестирование устойчивости системы к сбоям и восстановления данных.

Тестирование безопасности: исследование на уязвимости, такие как SQL-инъекции, XSS, CSRF, и проверка механизмов аутентификации.

3. Тестирование пользовательского интерфейса:

Оценка удобства использования: проверка интуитивности интерфейса для конечных пользователей.

Тестирование кроссбраузерности и кроссплатформенности: проверка работы системы в разных браузерах и на различных устройствах.

4. Интеграционное тестирование:

Интеграция с внутренней системой управления перевозками: проверка обмена данными между системой и внутренней платформой.

Взаимодействие с внешними сервисами: тестирование интеграции с внешними платежными системами и сервисами доставки.

5. Регрессионное тестирование:

Проверка воздействия обновлений на работу системы и выявление возможных ошибок после изменений.

6. Приемочное тестирование:

Проверка соответствия системы требованиям заказчика и утверждение сценариев использования перед запуском.

7. Тестированию масштабируемости:

Оценка способности системы масштабироваться для обработки увеличения объема данных и пользовательской нагрузки со временем.

8. Тестированию резервного копирования и восстановления:

Проверка эффективности механизмов резервного копирования и возможности быстрого восстановления данных в случае аварийных ситуаций.

9. Тестированию совместимости:

Проверка совместимости системы с различными операционными системами, базами данных и другими необходимыми компонентами.

Эти аспекты дополняют основные этапы тестирования и помогают обеспечить полную функциональность, надежность и безопасность информационной системы ОсОО «Сэм экспресс» в процессе перевозки товаров между странами.

Процесс тестирования включает планирование, подготовку данных, проведение тестов, анализ результатов и устранение дефектов. Тестирование информационной системы ОсОО «Сэм экспресс» гарантирует высокое качество и соответствие требованиям бизнеса и пользователей, обеспечивая успешную интеграцию системы в процессы перевозки товаров.

# ГЛАВА 4 Технико-экономическое обоснование разработки программного продукта, который должен содержать следующие подразделы

## 4.1 Определение суммарных затрат на разработку

### 4.1.1 Расчет прямых затрат

К прямым[[2]](#_СПИСОК_ЛИТЕРАТУРЫ) затратам относятся затраты на материалы и комплектующие, необходимые для выполнения данной разработки и полностью включаемые в ее стоимость.

Таблица 1 - Расчет прямых затрат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Элемент затрат | Цена за шт., сом. | Количество, шт. | Общая стоимость, сом. |
| 1 | Набор канцелярских принадлежностей | 20 | 4 | 80 |
| 2 | Бумага A4 (1 пачка) | 210 | 1 | 210 |
|  | | | Итого: | 290 |

### 4.1.2 Расчет затрат на основную заработную плату

Затраты на основную заработную плату разработчиков определяют как произведение среднедневной заработной платы на трудоемкость разработки по всем работам, выраженную в днях.

Таблица 2 Расчет затрат на основную заработную плату

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Разработчики | Заработная плата, сом./дн. | Шифры выполняемых работ | Общая продолжительность работ, дн. | Основная заработ-ная плата, сом. |
| 1 | Руководитель работы | 150 | 1-2, 4-5, 10-11, 13-14 | 23 | 3450 |
| 2 | Студент | 100 | 1-2, 2-3, 3-4, 5-6, 6-7, 7-8, 8-9, 9-10, 10-11, 11-12, 12-13, 13-14, 14-15 | 55 | 5500 |
|  | | | | Итого: | 8950 |

### 4.1.3 Отчисления на социальные нужды

Из начисленной заработной платы в пользу работников выплачивается 18%, из которых:

15 % направляются в Пенсионный фонд;

2 % в Фонд обязательного медицинского страхования;

1 % в профсоюзный фонд .

Отчисления на социальные нужды (СоцФ) вычисляется по формуле (4.1):

СоцФ = ЗП \* КСоцФ, (4.1)

где ЗП – начисленная заработная плата в пользу работников, сом;

КсоцФ – процент отчислений на социальные нужды от начисленной заработной платы в пользу работников (17,25%);

СоцФ = 8950\*0,1725 = 1543,88 сом.

### 4.1.4 Затраты на содержание и эксплуатацию ПК

Используемые ПК[[3]](#_СПИСОК_ЛИТЕРАТУРЫ) предназначены не только для выполнения данной разработки, поэтому учитываются только текущие затраты на содержание и эксплуатацию.

Расчет стоимости используемых ПК.

Таблица 3 Затраты на содержание и эксплуатацию ПК

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Оборудование | Цена за шт., сом. | Количество, шт. | Общая стоимость, сом. |
| 1 | Системный блок: CPU Intel Core i3-10100 (3.6-4.3GHz), 6MB, LGA1200, Cache L3,EMT64,4 Cores + 8 Threads,Tray,Comet Lake  Клавиатура: Genius KB-320e,  Мышь: A4Tech BT-630,  Монитор: .ACER 19.5  Колонки: Genius SP-S110 2.0  Суммарная мощность: 500 Вт | 39463 | 1 | 39463 |
| 2 | Системный блок: Сpu Intel i3-10100 (3.6-4.3GHz), 6MB, LGA1200, Cache L3,EMT64,4 Cores + 8 Threads,Tray,Comet Lake  Клавиатура: Genius SlimStar 110,  Мышь: Genius NetScroll 120,  Монитор: LG 17”  Колонки: Genius SP-Q06S 2.0  Суммарная мощность: 400 Вт | 19501 | 1 | 19501 |
|  | | | Итого: | 36131 |

В стоимость часа работы ПК включают следующие затраты:

Амортизационные отчисления, приходящиеся на 1 час работы ПК (А), вычисляется по формуле (4.3):

А = Фп \* На / Фв, (4.2)

Где

Фп – первоначальная стоимость ПК, сом;

На – норма амортизации;

Фв – фонд времени работы ПК за год.

Нормы амортизации (На), вычисляются по формуле:

На = [(Фп – Фл) / (Фп \* Tсл)]\*100 , (4.3)

где

Фп – первоначальная стоимость ПК, сом;

Фл – ликвидационная стоимость, сом;

Tсл – срок службы оборудования (10 лет).

Первоначальная стоимость ПК1 (Фп): 39463 сом.

Первоначальная стоимость ПК2 (Фп): 19501 сом.

Ликвидационная стоимость (Фл): 2000 сом.

На= [(39463-2000)/( 39463\*10)]\*100 = 9,5%

На= [(19501-2000)/( 19501\*10)]\*100 = 9,0%

Фонд времени работы ПК (Фв), вычисляются по формуле:

Фв = Чг – (Вг \* Чд + Вм \* 12), (4.4)

где

Чг – число рабочих часов в году (1985 час.);

Вг – время профилактики ежегодное (6 суток);

Чд – число рабочих часов в день (8 час.);

Вм – время профилактики ежемесячное (5 час.);

Фв = 1985 – (6 \* 8 + 5 \* 12) = 1877 (час.)

Расчет амортизационных отчислений (А), по формуле:

А= (39463\*0,095)/1887 = 1,99 сом/ч. (4.5)

А= (19501\*0,09)/1887 = 0,93 сом/ч.

2) Затраты на электрическую энергию, приходящиеся на 1 час работы ПК (Сэл), вычисляется по формуле (5.6):

Сэл = W \* Цэл, (4.6)

где

W – установленная мощность оборудования, кВт/ч (ПК 1=0.5 , ПК 2=0.4);

Цэл - тариф на 1 кВт/ч для прочих непромышленных потребителей:

1,327 сом (без учёта налогов);

1,5 сом (с учетом налогов).

Сэл = 0,5\*1,5 = 0,75 сом/ч.

Сэл = 0,4\*1,5 = 0,6 сом/ч.

3) Затраты на текущий ремонт и профилактику, приходящиеся на 1 час работы ПК (Спр), вычисляется по формуле (4.7):

Cпр = (Фп \* Кзп) / Фв, (4.7)

где

Фп – первоначальная стоимость ПК, сом;

Кзп – процент затрат на текущий ремонт и профилактику от стоимости ПК (5%);

Фв – фонд времени работы ПК за год (1877 час.).

Cпр = 39463\*0,05/1877 = 1,05 сом / ч.

Cпр = 19501\*0,05/1877 = 0,52 сом / ч.

4) Затраты на материалы, необходимые для обеспечения эксплуатации, приходящиеся на 1 час работы ПК (Смэ) , вычисляется по формуле (4.8):

Cмэ = (Фп \* Кзм) / Фв, (4.8)

где

Фп – первоначальная стоимость ПК, сом;

Кзм – процент затрат на материалы, необходимые для обеспечения эксплуатации от стоимости ПК (5%);

Фв – фонд времени работы ПК за год (1877 час.).

Cмэ = 39463\*0,05/1877 = 1,05 сом / ч.

Cмэ = 19501\*0,05/1877 = 0,52 сом / ч.

Затраты в целом, на содержание и эксплуатацию ПК (Спк), вычисляется по формуле (4.9):

Спк = Д \* Ч \* (А + Сэл + Cпр + Cмэ), (4.9)

где

Д - количество дней использования;

Ч - количество часов в день использования;

А - амортизационные отчисления, приходящиеся на 1 час работы, сом;

Сэл - затраты на электрическую энергию, приходящиеся на 1 час работы, сом;

Cпр - затраты на текущий ремонт и профилактику, приходящиеся на 1 час работы, сом;

Cмэ - затраты на материалы, необходимые для обеспечения эксплуатации, на 1 час работы, сом.

Таблица 4 Затраты на 1 час работы оборудований разработчиками

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № | Разработчики | | Затраты, приходящиеся на 1 час работы оборудований, сом | | | | Затраты в целом, сом. |
| Дни | Часы | Амортизационные отчисления | Электрическая энергия | Текущий ремонт и профилактика | Материалы для эксплуатации |
| 1 | 23 | 4 | 1,99 | 0,75 | 1,05 | 1,05 | 444,04 |
| 2 | 55 | 8 | 0,93 | 0,6 | 0,52 | 0,52 | 1126,79 |
|  | | | | | | Итого: | 1570,84 |

Спк= 23 \* 4 \* (1,023 + 0,75 + 0,565 + 0,565 ) = 266,99 сом.

Спк= 55 \* 8 \* (0,689 + 0,6 + 0,398 + 0,398 ) = 917,27 сом.

### 4.1.5 Амортизация нематериальных активов

Нематериальные активы [[1]](#_СПИСОК_ЛИТЕРАТУРЫ)- Лицензионное программное обеспечение далее в тексте программное обеспечение (ПО), используемые на ПК предназначены не только для выполнения данной разработки, поэтому учитываются только текущие затраты на амортизацию нематериальных активов (НМА), по формуле (4.10):

НМА = (Фна \* Тисп) / (Тна \* Фв), (4.10):

где

Фна - стоимость лицензии ПО, сом;

Tисп - время использования ПО для выполнения данной разработки, час;

Tна - срок действия лицензии, год (если бессрочная: Tна = 5);

Фв - фонд времени работы ПК (1877 час.).

Таблица 5 Амортизация нематериальных активов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Состав нематериальных активов | Лицензия | | Тисп, час | НМА, сом |
| Цена, сом | Срок, год |
| 1 | Microsoft Office 2007 | 43200 | Бессрочная | 389 | 1790,6 |
|  | | | | Итого: | 1790,6 |

НМА 1 = (43200\*389)/(5\*1877) = 1790,6 сом.

### 4.1.6 Накладные расходы

В их состав входят затраты на ремонт и содержание помещений, расходы на освещение, кондиционирование воздуха, расходы средства связи, расходы на охрану помещения.

Примем размер накладных расходов как 100% от основной заработанной платы. Таким образом, сумма накладных расходов составит 8950 сом.

Суммарные затраты на разработку:

Таблица 6 Накладные расходы

|  |  |  |
| --- | --- | --- |
| № | Элемент затрат | Сумма, сом. |
| 1 | Прямые затраты | 290,00 |
| 2 | Основная заработная плата | 8950,00 |
| 3 | Отчисления на социальные нужды | 1543,88 |
| 4 | Содержание и эксплуатация ПК | 1570,82 |
| 5 | Амортизация нематериальных активов | 1790,6 |
| 6 | Накладные расходы | 8950,00 |
|  | Итого: | 23095,3 |

## 4.2 Расчет ориентировочной цены

Поскольку потенциальная емкость рынка для данной разработки рассчитана, то ориентировочную цену (Цпо), без учета транспортных затрат, рассчитаем по формуле (4.11):

Цпо = C / n + Cдоп + П, (4.11)

где

С - суммарные затраты на разработку, сом;

n - потенциальное число потребителей;

Cдоп - дополнительные затраты, возникающие при продаже разработки каждому потребителю, это упаковка, DVD-диск (50 сом);

П - прибыль, рассчитываемая в процентах от (C /n + Cдоп) (25%).

Цпо = 23095,3/ 50 + 50 + ((23095,3/ 50 + 50)\*0,25) = 689,88 сом.

Вывод: данный проект является эффективным для потребителя и может приносить экономическую выгоду.

# ЗАКЛЮЧЕНИЕ

При разработке информационной системы для обеспечения перевозки товаров между странами ОсОО «Сэм экспресс» были сформулированы основные цели, направленные на оптимизацию логистических процессов и улучшение общего управления бизнесом.

Созданная информационная система предоставляет современные инструменты для эффективного контроля и управления перевозками, автоматизации процессов отслеживания грузов, мониторинга их статуса и обратной связи с клиентами. Это существенно сокращает время выполнения оперативных задач и повышает общий уровень обслуживания клиентов.

Внедрение информационной системы также способствует улучшению аналитики в области перевозок, оптимизации маршрутов и повышению лояльности клиентов за счет персонализированного подхода.

Учитывая вышеизложенные факторы, можно заключить, что создание и внедрение информационной системы для обеспечения перевозки товаров является ключевым этапом в развитии ОсОО «Сэм экспресс». Ожидается, что это приведет к увеличению эффективности перевозок, улучшению взаимодействия с клиентами и укреплению позиций компании на рынке международных перевозок.

Предложенная информационная система также способствует повышению эффективности управления складскими запасами и планированию перевозок, что позволяет снизить издержки и оптимизировать процессы доставки товаров. Интеграция аналитических инструментов обеспечивает возможность более точного прогнозирования спроса и адаптации бизнес-стратегии в соответствии с требованиями рынка.

Кроме того, создание информационной системы позволяет собирать и анализировать данные о перевозках, что помогает выявить особенности и потребности клиентов, а также предоставлять персонализированные рекомендации и услуги, увеличивая тем самым удовлетворенность клиентов и конкурентоспособность компании.

Общая цель создания информационной системы для ОсОО «Сэм экспресс» заключается в повышении качества обслуживания клиентов, увеличении прибыли и укреплении лидирующих позиций на рынке международных перевозок.

# СПИСОК ЛИТЕРАТУРЫ

1. Советов Б.Я., Цехановский В.В. Информационные технологии: Учебник для вузов. – 3-е изд., стер. – М.: Высшая школа, 2006. – 263 с.6 ил.
2. Рыжиков Ю.И. Вычислительные методы. – СПб.: БХВ-Петербург, 2007. – 400 с.: ил.
3. Беляев А.А., Коротков Э.М. Системология организации: Учеб- ник / Под ред. д-ра экон. наук, проф. Э.М. Короткова. – М.: ИНФРАМ, 2000. – 182 с. С четырьмя и более авторами
4. Eric Newell «Mastering Microsoft Dynamics 365 Implementations»   
   John Wiley & Sons Limited

**Электронный ресурс**

1. "JS для начинающих" - https://learn.javascript.ru/
2. "HTMLBook" - <https://htmlbook.ru/>
3. "CSS гид по Flexbox" - https://css-tricks.ru/css-flexbox-guide/
4. Учебник по HTML и CSS, включающий примеры кода, правила оформления и подробное описание различных элементов и свойств.
5. Интерактивный учебник по JavaScript для начинающих разработчиков.
6. "React документация" - https://ru.reactjs.org/
7. Официальная документация React на русском языке. Включает примеры кода, руководства и рекомендации по разработке на React.
8. "SQL с нуля" - https://sql-ex.ru/
9. Подробное руководство по использованию Flexbox в CSS для создания гибкого и адаптивного макета.
10. "PostgreSQL документация" - https://postgrespro.ru/docs/postgresql
11. "React документация" - https://ru.reactjs.org/
12. "Хороший код" - https://goodcode.io/
13. Блог с статьями, советами и примерами кода по различным аспектам разработки, включая JavaScript, React, SQL и другие технологии.
14. "Хекслет" - <https://ru.hexlet.io/>
15. "Системный анализ и проектирование" [Попов Виктор Юрьевич](https://www.hse.ru/org/persons/24487003), [Сизых Наталья Васильевна](https://www.hse.ru/org/persons/192085683). https://www.hse.ru/edu/courses/292688483
16. "HTML Academy" - https://htmlacademy.ru/
17. Образовательная платформа, предлагающая интерактивные курсы по HTML, CSS и JavaScript, включая курсы по разработке интерфейсов и адаптивного дизайна.
18. The Unified Modeling Language Resource Site (<http://www.uml-diagrams.org/>).
19. Справочник по веб-технологиям, включающий описание основных концепций, терминов и инструкций по использованию различных технологий.
20. "Next.js для начинающих" - https://www.udemy.com/course/nextjs-for-beginners/
21. "Next.js документация" - https://nextjs.org/docs