Техническое задание

Проект: Заказы компании перевозок с участием морских портов

1. Термины и определения

1.1. Общие термины

Система – программное приложение, предназначенное для управления заказами на перевозку грузов с участием морских портов.
Компания перевозок – организация, оказывающая услуги транспортировки грузов.
Заказ – запрос клиента на организацию перевозки груза.
Груз – материальный объект, подлежащий транспортировке.
Морской порт – объект инфраструктуры, предназначенный для приёма и отправки судов.
Транспорт – судно, грузовик или иное средство, используемое для перевозки.
Маршрут – путь следования груза от начального до конечного пункта, включая промежуточные порты.
CRUD – набор основных операций: Create (создание), Read (чтение), Update (обновление), Delete (удаление).

1.2. Бизнес-термины

Клиент – физическое или юридическое лицо, заказывающее перевозку.
Оператор – сотрудник компании, управляющий заказами.
Логистика – процесс планирования, реализации и контроля эффективного движения и хранения товаров.
Событие в порту – запись о входе/выходе/разгрузке/погрузке судна или груза.
Отчёт – структурированный документ с информацией о деятельности, заказах, грузах и т.д.

1.3. Технические термины

PostgreSQL – реляционная система управления базами данных (СУБД).
MongoDB – документоориентированная система управления базами данных (СУБД).
БД – база данных.
API – интерфейс программирования приложений.
Распределённая система – система, состоящая из компонентов, расположенных на разных узлах сети.
Репозиторий Git – удалённое хранилище исходного кода проекта.

1.4. Другие термины

TBD – To Be Defined (будет определено позже).
UML – унифицированный язык моделирования.

2. Общие положения

2.1. Назначение документа

Настоящий документ определяет требования к системе управления заказами компании перевозок с участием морских портов. Он служит основой для:

- проектирования структуры базы данных (реляционной и документоориентированной),
- разработки приложения,
- тестирования,
- сдачи проекта.

2.2. Цели создания Системы
Автоматизация учёта и управления заказами на перевозку.
Интеграция с морскими портами для отслеживания статуса грузов.
Поддержка распределённой архитектуры с использованием разных типов СУБД (PostgreSQL и MongoDB).
Обеспечение гибкости и масштабируемости для роста бизнеса.
Обеспечение безопасности и целостности данных.
2.3. Основные функциональные возможности Системы

- Управление заказами: создание, просмотр, изменение, удаление.
- Управление клиентами: хранение информации о клиентах, контактных данных.
- Управление транспортом: информация о судах, грузовиках, их загрузке и маршрутах.
- Управление портами: информация о портах, их географическом положении, доступности.
- Управление маршрутами: построение и хранение информации о маршрутах перевозки.
- Управление грузами: информация о весе, типе, статусе, владельце.
- Отслеживание событий в портах: вход/выход судна, погрузка/разгрузка груза.
- Генерация отчётов: статистика по заказам, грузам, портам, транспорту.
- Интеграция данных: связь между PostgreSQL и MongoDB через внешние ключи.

2.4. Использование Технического Задания
ТЗ используется командой разработчиков как основной источник требований.
Все спорные или неоднозначные моменты уточняются и фиксируются в истории изменений.

3. Функциональные требования

3.1. Диаграммы Вариантов Использования
(Описание диаграммы:)
Диаграммы Вариантов Использования (Use Case Diagram) отображают взаимодействие между актёрами (например, Клиент, Оператор, Система) и основными функциями системы.

	Описание
Клиент	Физическое или юридическое лицо, заказывающее перевозку.
Оператор	Сотрудник компании, управляющий заказами и грузами.
Порт	Внешняя система, передающая данные о событиях (разгрузка).
Система	Ядро приложения, обеспечивающее хранение и обработку данных.

3.2. Описание Вариантов Использования
3.2.1. ВИ «Зарегистрироваться как Клиент»

3.2.1.1. Описание ВИ
Клиент должен иметь возможность зарегистрироваться в системе, указав свои контактные данные.

3.2.1.2. Предусловия
Пользователь не зарегистрирован в системе.
Имя, email и телефон указаны корректно.
3.2.1.3. Основной поток действий
Пользователь выбирает действие "Зарегистрироваться".
Система отображает форму регистрации (см. п. 4.1 Форма «Регистрация»).
Пользователь вводит:

- Имя
- Email
- Телефон
- Пароль
- Пользователь подтверждает регистрацию.
- Система проверяет корректность данных.
- Система сохраняет клиента в clients (PostgreSQL).
- Система отправляет подтверждение по email.
- Система выводит сообщение об успешной регистрации.

3.2.1.4. Альтернативные потоки действий
3.2.1.4.1. Некорректные данные
Если данные введены некорректно, система выводит сообщение об ошибке и возвращается к шагу 3.
3.2.1.4.2. Email уже зарегистрирован
Система выводит сообщение "Пользователь с таким email уже существует".
3.2.1.5. Бизнес-правила
Email должен быть уникальным.
Пароль должен соответствовать требованиям безопасности.
3.2.2. ВИ «Создать заказ»
3.2.2.1. Описание ВИ
Клиент или Оператор должен иметь возможность создать новый заказ на перевозку груза.

3.2.2.2. Предусловия
Пользователь (Клиент или Оператор) авторизован в системе.
Существует клиент, транспорт, маршрут, груз.
3.2.2.3. Основной поток действий
Пользователь выбирает действие "Создать заказ".
Система отображает форму заказа.
Пользователь вводит:

- Клиента
- Груз
- Маршрут (откуда, куда)
- Предпочтительную дату отправки
- Пользователь подтверждает создание.
- Система проверяет корректность данных.
- Система сохраняет заказ в orders (PostgreSQL).

- Система возвращает ID заказа.

3.2.2.4. Альтернативные потоки действий
3.2.2.4.1. Некорректные данные
Если данные введены некорректно, система выводит сообщение об ошибке и возвращается к шагу 3.
3.2.2.5. Бизнес-правила
Заказ не может быть создан без указания груза и маршрута.
Клиент должен быть зарегистрирован в системе.
3.2.3. ВИ «Отследить груз»
3.2.3.1. Описание ВИ
Пользователь должен иметь возможность отследить текущее местоположение и статус груза.

3.2.3.2. Предусловия
Груз создан и связан с заказом.
События в портах (в MongoDB) существуют.
3.2.3.3. Основной поток действий
Пользователь вводит ID груза.
Система находит груз в cargo (PostgreSQL).
Система запрашивает события по cargo_id в tracking_events (MongoDB).
Система отображает хронологию событий: дата, порт, статус (в пути, разгружен и т.д.).
3.2.3.4. Альтернативные потоки действий
3.2.3.4.1. Груз не найден
Система выводит сообщение "Груз не найден".
3.2.3.5. Бизнес-правила
Только авторизованный пользователь может отслеживать груз.
Статус груза обновляется при наступлении события в порту.
3.2.4. ВИ «Создать событие в порту»
3.2.4.1. Описание ВИ
Система должна получать события от портов и сохранять их в port_logs (MongoDB).

3.2.4.2. Предусловия
Событие отправлено внешней системой (порт).
ID груза и порта существуют.
3.2.4.3. Основной поток действий
Внешняя система отправляет событие (например, "вход", "разгрузка") с ID груза и порта.
Система проверяет существование груза и порта.
Система сохраняет событие в port_logs (MongoDB).
Система обновляет статус груза (если применимо).
3.2.4.4. Альтернативные потоки действий
3.2.4.4.1. Груз не найден
Система отклоняет событие и логирует ошибку.
3.2.4.5. Бизнес-правила
Событие может быть только одного типа за раз (например, "вход" или "разгрузка").
Событие должно быть связано с существующим грузом.

4. Требования к формам

4.1. Форма «Заказ»

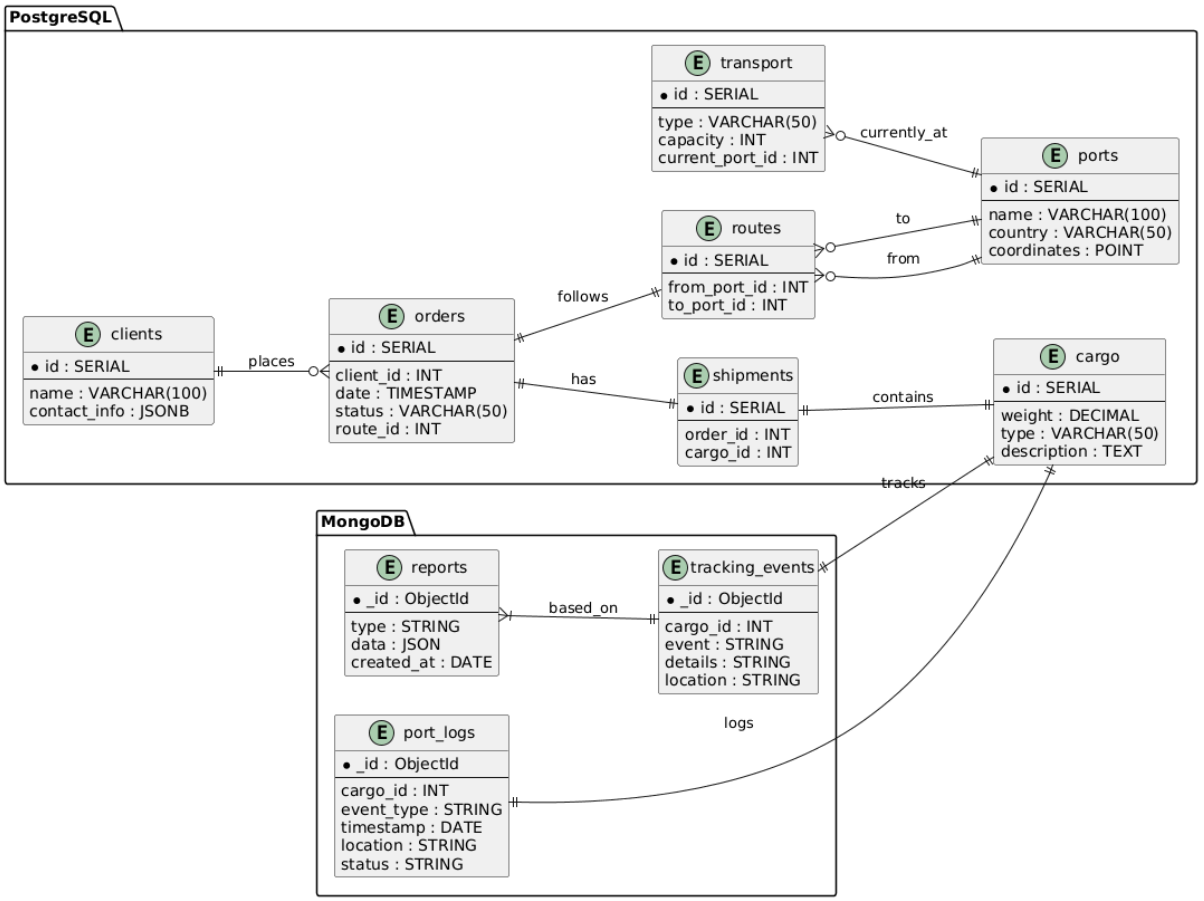
№	Название	Тип	Описание
1	ID заказа	Число	Уникальный идентификатор (автоинкремент)
2	Клиент	Ссылка	ID клиента в PostgreSQL
3	Груз	Ссылка	ID груза в PostgreSQL
4	Маршрут	Ссылка	ID маршрута в PostgreSQL
5	Дата создания	Дата	Автоматически заполняется
6	Статус	Строка	В обработке, в пути, доставлен

4.2. Форма «Груз»

№	Название	Тип	Описание
1	ID груза	Число	Уникальный идентификатор
2	Вес	Десятичное	В килограммах
3	Тип	Строка	Опасный, хрупкий, стандартный
4	Описание	Текст	Произвольное описание
5	Владелец	Ссылка	ID клиента

5. Модель данных
5.1. Общие положения
В настоящем разделе описывается логическая структура базы данных системы управления заказами компании перевозок с участием морских портов. Структура включает сущности, их атрибуты и связи между ними. Система использует распределённую архитектуру, включающую:

Реляционную базу данных (PostgreSQL) — для хранения структурированных данных с жёсткими связями (например, заказы, клиенты, транспорт).
Документоориентированную базу данных (MongoDB) — для хранения гибких, часто изменяющихся данных (например, события в портах, отчёты).
5.2. UML-диаграмма сущностей и связей
На диаграмме представлены основные сущности системы и их взаимосвязь.
Диаграмма выполнена в нотации UML (Use Case Diagram).



Описание основных сущностей и их связей.

Сущность	**Описание**
clients	Информация о клиентах: имя, контактные данные.
orders	Заказы на перевозку: дата, статус, клиент, маршрут.
cargo	Грузы: вес, тип, описание.
shipments	Связь между заказом и грузом.

routes	Маршруты: начальный и конечный порт.
ports	Морские порты: название, страна, координаты.
transport	Транспорт: тип, вместимость, текущий порт.

tracking_events	События отслеживания груза: тип события, детали, местоположение.
port_logs	Логи событий в портах: вход/выход, разгрузка, статус.
reports	Отчёты: тип, данные в формате JSON, дата создания.

Связь	Описание
orders.client_id→clients.id	Один клиент может сделать много заказов.
orders.id→shipments.order_id	Один заказ может включать один или несколько грузов.
shipments.cargo_id→cargo.id	Один груз может быть связан с одним заказом.
routes.from_port_id→ports.id	Маршрут начинается в одном порту.
routes.to_port_id→ports.id	Маршрут заканчивается в одном порту.
cargo.id→tracking_events.cargo_id	Один груз может иметь много событий отслеживания.
cargo.id→port_logs.cargo_id	Один груз может проходить через несколько портов.

6. Нефункциональные требования

6.1. Интерфейс пользователя
Веб-интерфейс должен быть интуитивно понятным и удобным.
Поддержка основных браузеров: Chrome, Firefox, Safari, Edge.
6.2. Требования к производительности
Система должна отображать любую форму не дольше 5 секунд.
Система должна обрабатывать до 100 одновременных запросов.
Время отклика API не должно превышать 2 секунд.
6.3. Требования к безопасности
Система должна использовать аутентификацию и авторизацию.
Доступ к данным должен быть ограничен по ролям.
Все пароли должны храниться в зашифрованном виде.

7. Требования к приемке-сдаче проекта

Комплект поставки:
Техническое задание
Исходный код (на Git)
SQL и JS-скрипты
Документация
Примеры запросов и CRUD-операций
Приемо-сдаточные испытания проводятся по согласованному графику.
Исполнитель и Заказчик подписывают акт приёмки.