**СОДЕРЖАНИЕ**

[Введение 7](#_Toc168418885)

[1 Обоснование целесообразности создания подсистемы по учёту операций с контейнерами 9](#_Toc168418886)

[1.1 Описание предметной области 9](#_Toc168418887)

[1.1.1 Введение в терминально-складские комплексы 9](#_Toc168418888)

[1.1.2 Особенности работы Автоматизированной системы управления терминально-складскими комплексами 10](#_Toc168418889)

[1.2 Описание текущей работы по учёту операций с контейнерами в АРМ ПКП 12](#_Toc168418890)

[1.3 Обоснование разработки 17](#_Toc168418891)

[2 Требования на разработку подсистемы 19](#_Toc168418892)

[2.1 Требования к системе в целом 20](#_Toc168418893)

[2.1.2 Требования к структуре и функционированию системы 20](#_Toc168418894)

[2.1.2 Требования к численности и квалификации пользователей системы 25](#_Toc168418895)

[2.1.3 Требования к показателям назначения 26](#_Toc168418896)

[2.1.4 Требования к эргономичности и технической эстетике 26](#_Toc168418897)

[2.1.5 Требования к патентной чистоте 26](#_Toc168418898)

[2.2 Требования к функциям (задачам), выполняемым разрабатываемой системой 26](#_Toc168418899)

[2.2.1 Справочная форма «Книга выгрузки контейнеров ГУ-44к» 26](#_Toc168418900)

[2.2.2 Справочная форма «Наличие контейнеров с грузом» 27](#_Toc168418901)

[2.2.3 Справочная форма «Книга завоза контейнеров» 28](#_Toc168418902)

[2.2.4 Справочная форма «Книга вывоза контейнеров» 29](#_Toc168418903)

[2.3 Требования к видам обеспечения 30](#_Toc168418904)

[2.3.1 Требования к математическому обеспечению 30](#_Toc168418905)

[2.3.2 Требования к лингвистическому обеспечению 30](#_Toc168418906)

[2.3.3 Требования к системному программному обеспечению 30](#_Toc168418907)

[3 Разработка программного обеспечения оперативно-аналитической подсистемы по учёту операций с контейнерами 32](#_Toc168418908)

[3.1 Описание функционала 32](#_Toc168418909)

[3.2 Моделирование и визуализация с помощью StarUML 33](#_Toc168418910)

[3.2.1 Диаграмма последовательности (Sequence Diagram) 33](#_Toc168418911)

[3.2.2 Диаграмма классов (Class Diagrams) 36](#_Toc168418912)

[3.3 Программная реализация веб-приложения 37](#_Toc168418913)

[3.3.1 SQL Запросы для формирования справочных форм 39](#_Toc168418914)

[3.4 Пользовательский интерфейс 43](#_Toc168418915)

[3.4.1 Выбор справочной формы 43](#_Toc168418916)

[3.4.2 Фильтры 43](#_Toc168418917)

[3.4.3 Сообщение «Данные не найдены» 45](#_Toc168418918)

[3.4.4 Страница справочной формы «Книга выгрузки контейнеров ГУ-44к» 45](#_Toc168418919)

[3.4.5 Страница справочной формы «Наличие контейнеров с грузом» 46](#_Toc168418920)

[3.4.6 Страница справочной формы «Книга завоза контейнеров» 47](#_Toc168418921)

[3.4.7 Страница справочной формы «Книга вывоза контейнеров» 47](#_Toc168418922)

[4 Оценка разработанной подсистемы 49](#_Toc168418923)

[Заключение 54](#_Toc168418924)

[Список использованных источников 55](#_Toc168418925)

[Приложение А Форма «Книга выгрузки контейнеров ГУ-44К» 56](#_Toc168418926)

[Приложение Б Форма «Наличие контейнеров с грузом» 57](#_Toc168418927)

[Приложение В Форма «Книга завоза контейнеров» 58](#_Toc168418928)

[Приложение ГФорма «Книга вывоза контейнеров» 59](#_Toc168418929)

[Приложение ДИнтерфейс, отвечающий за фильтры. 60](#_Toc168418930)

[Приложение ЕРезультаты проведенных экспериментов. 64](#_Toc168418931)

[Приложение ЖЛистинг кода фронтенда 65](#_Toc168418932)

[Приложение ИЛистинг кода серверной части бэкенда 82](#_Toc168418933)

# **ВВЕДЕНИЕ**

В современном мире транспортно-логистические комплексы становятся всё более сложными и разветвлёнными, что требует эффективного управления и контроля за операциями. Одним из ключевых элементов в этой системе является являются терминально-складские комплексы (ТСК), где осуществляется приём, хранение и отгрузка грузов, включая контейнеры. Для эффективного управления такими комплексами внедряются и разрабатываются Автоматизированные системы управления терминально-складскими комплексами (АСУТСК).

В рамках данного исследования рассматривается обоснование создания оперативно-аналитической подсистемы по учёту операций с контейнерами в составе АСУТСК. Рассмотрено текущее состояние учёта операций с контейнерами в автоматизированном рабочем месте приемосдатчика контейнерной площадки (АРМ ПКП), выявлены проблемы и недостатки данной системы, а также предложено готовое решение в виде оперативно-аналитической подсистемы.

Важным аспектом проекта является анализ средств и инструментов разработки, изучение применения веб-технологий и особенностей создания пользовательского интерфейса. Для выполнения задач будут использованы следующие инструменты: среда разработки «VS Code», инструмент тестирования «Postman» и инструмент моделирования «Star UML». Особое внимание уделяется применению веб-технологий и созданию интуитивно понятного интерфейса.

Практическая часть работы включает разработку интерфейса с использованием инфраструктуры АСУТСК, создание диаграмм классов и последовательности для серверной части, а также разработку самого программного обеспечения. Основные требования включают наличие диаграмм и отлаженное программное обеспечение. Выполнение этой работы обеспечит практический опыт в разработке программных систем, освоение современных инструментов и технологий, а также улучшение навыков проектирования и разработки ПО.

Кроме того, в исследовании уделяется особое внимание производительности подсистемы, отвечающих за обработку и выдачу данных. Одним из ключевых показателей производительности является время формирования справочных форм. Исследуется экспериментальное определение среднего времени обработки операций по формированию отчётов в разработанной подсистеме, будет представлен детальный анализ временных характеристик разработанной подсистемы и будет сделан вывод о её соответствии заданным требованиям.

# **Обоснование целесообразности создания подсистемы по учёту операций с контейнерами**

## **Описание предметной области**

### **Введение в терминально-складские комплексы**

Терминально-складские комплексы (ТСК) – это инфраструктурные объекты, предназначенные для приема, временного хранения, обработки и отправки грузов и контейнеров, включая различные этапы и операции в логистической цепи, связанные с их обработкой [3].

ТСК играют ключевую роль в логистической цепи, обеспечивая эффективное управление потоком грузов на протяжении всего пути от отправителя к получателю. ТСК включают в себя различные элементы, такие как складские помещения, площадки для маневрирования грузовых транспортных средств, терминалы обработки грузов, системы складского учета и контроля, а также инфраструктуру для передачи и обработки данных.

В ТСК выполняются различные операции, такие как выгрузка и загрузка грузов, сортировка, упаковка, размещение на складе, инвентаризация, отслеживание и контроль качества. Эти процессы выполняются с использованием различных технологий и оборудования, включая автоматизированные системы и роботизированные устройства.

Учет операций с контейнерами играет решающую роль в оптимизации процессов в ТСК. Он позволяет отслеживать перемещение контейнеров по складу или терминалу, определять их местоположение, состояние и содержимое. Это обеспечивает более точное планирование и управление складскими запасами, минимизирует потери времени и снижает риски ошибок.

Эффективная система учета также способствует повышению безопасности и обеспечению надежности операций. Путем непрерывного отслеживания перемещения грузов и контроля состояния контейнеров, система учета позволяет оперативно реагировать на потенциальные проблемы, предотвращая утерю или повреждение грузов. Мониторинг сроков хранения грузов и соблюдение норм и стандартов обеспечивают безопасность и качество обслуживания. Анализ данных о перемещении, состоянии и использовании контейнеров помогает выявить узкие места в операционных процессах и принять меры по их устранению, что в конечном итоге повышает эффективность работы комплекса и улучшает общую производительность логистических цепочек.

Постоянное улучшение методов учета операций с контейнерами и технологий на современные решения способствует повышению производительности и конкурентоспособности терминально-складских комплексов. Они становятся более гибкими, адаптивными к изменяющимся рыночным условиям и требованиям клиентов.

Создание современной системы управления ТСК, способные обеспечить автоматизацию, централизацию, интеграцию и оптимизацию всех процессов, а также обеспечить высокий уровень контроля.

ТСК играют ключевую роль в логистической цепи и их эффективное управление является критически важным для обеспечения эффективности и конкурентоспособности всей системы. Разработка подсистемы по учёту операций с контейнерами имеет большое значение для оптимизации работы ТСК и обеспечения их более эффективного функционирования.

### **Особенности работы Автоматизированной системы управления терминально-складскими комплексами**

Автоматизированная система управления терминально-складскими комплексами (АСУ ТСК) играет важную роль в повышении эффективности и оперативности управления железнодорожными терминалами и складскими комплексами.

Эта система представляет собой комплекс программно-аппаратных средств, предназначенных для автоматизации и оптимизации. АСУ ТСК в ОАО РЖД включает широкий спектр режимов, каждый из которых выполняет определенные функции. Эти режимы включают в себя уведомления для оперативной связи с сотрудниками, реестр документов для систематизации и управления документами, паспорта объектов для хранения основной информации о комплексах и оборудовании, а также функции по управлению ценами, клиентами, выполнению работ, анализу финансовой деятельности и другие. АСУТСК также обеспечивает ведение номенклатурно-справочной информации и управление профилями пользователей для эффективного функционирования системы.

Внедрение оперативно-аналитической подсистемы по учету операций с контейнерами и формирование справочных форм, таких как «Наличие контейнеров с грузом», «Книга завоза контейнеров», «Книга вывоза контейнеров» и «Книга выгрузки контейнеров ГУ-44к», позволяет значительно улучшить управление складскими операциями и повысить эффективность всей логистической цепочки.

Такая подсистема способствует повышению оперативности и точности учета всех операций с контейнерами. Формирование справочных форм позволяет оперативно отслеживать наличие контейнеров, их принадлежность, а также осуществлять контроль за грузами в контейнерах.

Это существенно снижает риск ошибок и упрощает процесс управления складскими запасами. Благодаря оперативно-аналитической подсистеме руководство компании получает возможность принимать более обоснованные и эффективные управленческие решения.

Информация, предоставляемая справочными формами, позволяет более точно прогнозировать потребности в ресурсах, планировать закупки новых контейнеров и оптимизировать распределение ресурсов для максимизации эффективности складских операций.

Кроме того, внедрение такой подсистемы значительно повышает прозрачность и контроль за процессами управления складом. Формирование книги завоза, вывоза и выгрузки контейнеров позволяет более детально контролировать каждую операцию с контейнерами, что способствует снижению рисков ошибок или недостаточной информированности. Это особенно важно в условиях динамичной логистической среды, где необходимо оперативно реагировать на изменения.

## **Описание текущей работы по учёту операций с контейнерами в АРМ ПКП**

АРМ ПКП – это автоматизированное рабочее место приемосдатчика контейнерной площадки. АРМ ПКП является составной частью автоматизированной системы управления станциями (АСУ СТ). Разработчик АРМ ПКП – Центр информационных технологий на транспорте ООО «ЦИТТранс»

АРМ ПКП представляет собой программное обеспечение как для отслеживания текущей деятельности контейнерной площадки, выполняемой персоналом приемосдатчиков на станции, так и для анализа работы станции, проводимого администрацией.

АРМ ПКП обеспечивает автоматизацию следующих функций:

* оформление приема;
* оформление без документных контейнеров;
* формление соответствующих технологических документов и ведение станционной отчетности (ПСА формы КЭУ-16, памяток формы ГУ-45, вагонных листов формы ГУ-38 и др.);
* учет работы контейнерной площадки;
* получение необходимых справок.

При работе с АРМ ПКП для каждого рабочего места требуется персональная электроно-вычислительная машина (ПЭВМ). Установкой необходимых компонентов и их настройкой на ПЭВМ занимаются как сами разработчики, так и системные администраторы, которые работают с помощью специальных программ. Так же системный администратор устанавливает серверные настройки, распространяющиеся на работу всего сервера, станции или отдельного рабочего места. Они определяют полномочия пользователя и учитывают особенности выполнения различных технологических операций – это влияет на структуру и состав справочных форм.

Для наглядности опишем процесс установки и запуска АРМ ПКП на диаграмме BPMN [1] (рисунок 1).

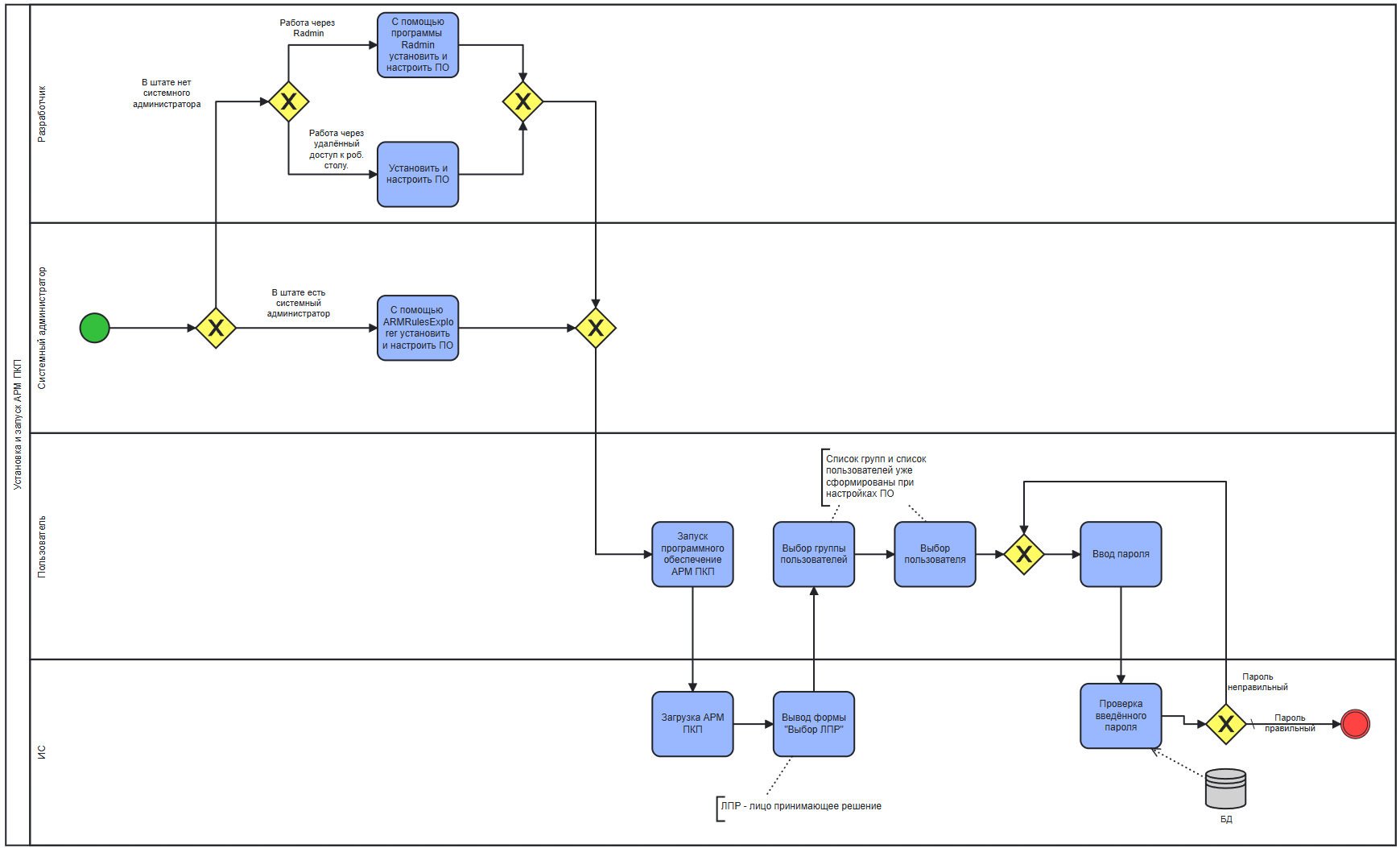


Рисунок 1 – Диаграмма как есть в нотации BPMN. Установка и запуск АРМ ПКП

При входе в систему на экране появится форма «Выбор ЛПР», из общего списка необходимо выбрать группу и пользователя, который в данный момент допущен к работе с программным обеспечением АРМ ПКП. Каждая группа имеет свои полномочия (права) и разрешенные места работы. Местом работы называется набор путей и контейнерных площадок, с которыми может работать пользователь.

Для работы с определенными справочными формами, такими как «Наличие контейнеров с грузом», «Книга завоза контейнеров», «Книга вывоза контейнеров» и «Книга выгрузки контейнеров ГУ-44к», необходима предварительная настройка системы на каждой станции и (или) в рабочем месте. Эту настройку выполняет системный администратор с помощью программы ARMRulesExplorer, а не пользователь напрямую.

При формировании нужной справочной формы, пользователь выбирает узел, где доступен список справочных форм. Для выбора определенных параметров в форме, пользователь может использовать фильтры, предоставленные системой. Эти фильтры могут отличаться для разных справочных форм. Например, пользователь может выбирать значения параметров из списка, использовать календарь для выбора даты и времени, или вводить числовые значения в соответствующие поля.

Далее следует установить требуемые в фильтрах параметры и нажать кнопку «Применить». Если в БД АСУ СТ есть данные, удовлетворяющие введенным параметрам, то появляется экранная форма отчета. Если данных нет- то предупреждение: «Нет данных для отчета».

Пример формирования справочной формы можно посмотреть на рисунке 2.

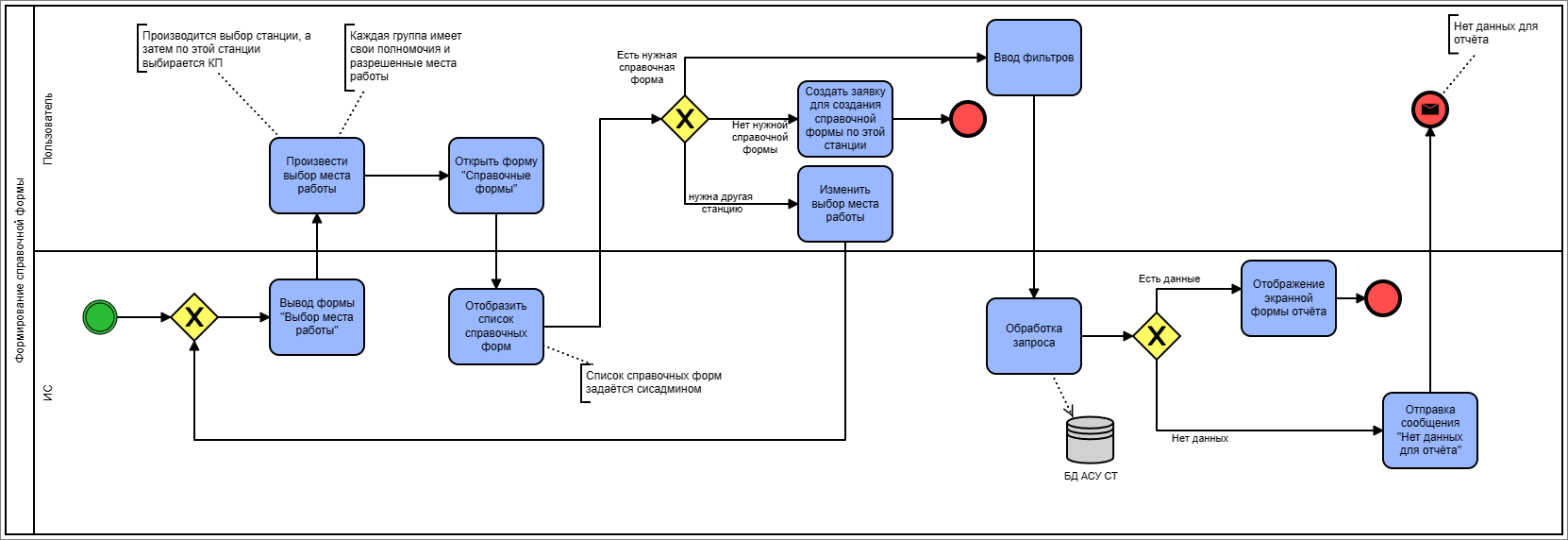


Рисунок 2 – Диаграмма как есть в нотации BPMN. Формирование справочной формы

## **Обоснование разработки**

На сегодняшний день основная часть задач сотрудников ЦМ в ОАО РЖД выполняется в рамках АСУ ТСК. Однако для ведения учета операций с контейнерами и получения необходимых справочных форм сотрудникам ЦМ приходится обращаться к сторонним системам, в частности, к АРМ ПКП.

Еще одним недостатком АРМ ПКП является его децентрализованность. Учет операций с контейнерами ведется только для конкретной станции, а информация о происходящем на других станциях не учитывается. Для получения полной справочной формы приходится объединять данные с нескольких станций.

При анализе работы программы АРМ ПКП и изучении диаграмм BPMN «Как есть» (рисунок 1 и рисунок 2) обнаружены определенные недочеты в бизнес-процессе. Данное приложение функционирует только на персональных ЭВМ и требует установки и настройки со стороны системных администраторов.

Настройка программы осуществляется системными администраторами, которые определяют, какие справочные формы будут доступны на конкретной станции. Это приводит к ситуации, когда пользователь может столкнуться с отсутствием нужной формы или ее неправильным отображением. В таких случаях пользователю приходится обращаться к системному администратору для добавления или настройки необходимой формы.

Кроме того, при входе в систему, пользователь сразу указывает на какой станции он будет работать, что затрудняет доступ к данным с других. Для получения справочной формы по другой станции приходится произвести настройки в отдельной вкладке, а не в самом фильтре.

Эти недостатки сказываются на эффективности и удобстве использования программы АРМ ПКП, замедляя процессы работы сотрудников и создавая дополнительные сложности при учёте операций с контейнерами. Для повышения производительности и удобства работы необходимо рассмотреть возможности улучшения и модернизации данной системы, включая упрощение процесса настройки и расширение функционала для более гибкого управления данными и формами.

АСУ ТСК уже запущена и выполняет ряд задач, аналогичных тем, которые выполняются в АРМ ПКП. Однако АСУ ТСК имеет несколько преимуществ. Она реализована на технологии тонкого клиента, что позволяет работать с ней через интернет-браузер без необходимости установки дополнительного программного обеспечения на рабочих местах пользователей. Нет необходимости в проведении начальных настроек при входе в систему - достаточно ввести логин и пароль, и система автоматически определит доступные пользователю операции и функционал. Это делает процесс работы более гибким и удобным для пользователей, сокращая время на подготовку к использованию системы и устраняя необходимость в установке специализированных приложений.

Разрабатываемая оперативно-аналитическая подсистема учёта операций с контейнерами, входящая в состав АСУ ТСК, нацелена на обеспечение максимального удобства для пользователей. Для этого предусмотрен единый и интуитивно понятный интерфейс с единым графическим дизайном, где основные элементы управления и навигации будут располагаться в одном и том же месте для удобства использования. Особое внимание уделяется упрощению процесса формирования справочных форм: фильтры для создания этих форм будут доступны на одной странице, что позволит пользователям удобно и быстро настраивать необходимую отчётность. Кроме того, предусматривается возможность получения справочных форм не только для отдельной станции, а сразу для всего терминально-складского комплекса.

Представим диаграмму как будет по формированию справочных форм в нотации BPMN (рисунок 3).

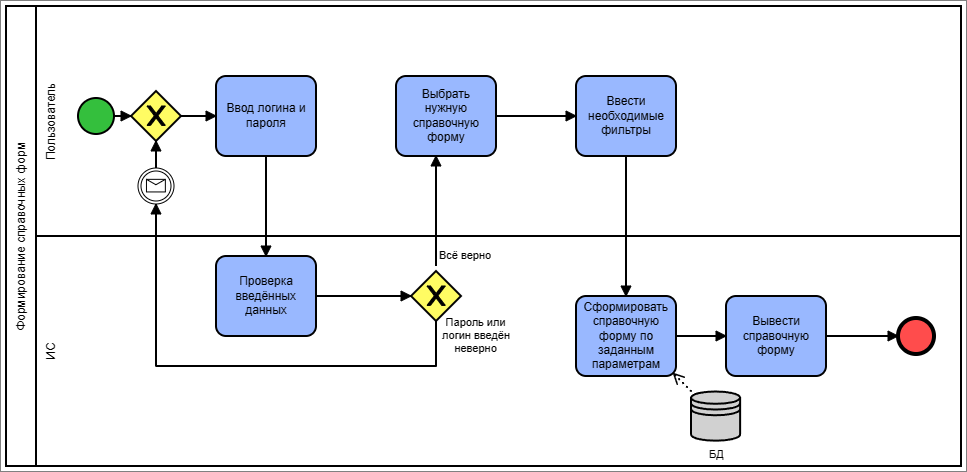


Рисунок 3 – Диаграмма как будет в нотации BPMN. Формирование справочной формы

# **Требования на разработку подсистемы**

Обоснованное и детальное определение требований к информационной системе является ключевым элементом успешного процесса создания системы. Содержание технического задания (ТЗ) на систему регламентируется ГОСТом 34.602–2020, который устанавливает основные аспекты требований к системе. Однако начальная работа по формированию требований начинается еще на стадии предпроектного обследования объекта информатизации, где определяются потребности пользователей относительно создаваемой информационной системы.

В процессе проектирования требования могут претерпевать изменения и уточнения, что является нормальной практикой и может быть необходимо для создания оптимальной системы. Однако, чем более полно и обоснованно сформулированы требования на начальных этапах, тем более успешным, быстрым и экономически эффективным может быть процесс создания системы.

Согласно ГОСТу, требования к автоматизированной системе можно разделить на три категории:

* требования к системе в целом;
* требования к функциям (задачам), выполняемым разрабатываемой системой;
* требования к различным типам обеспечения (информационному, техническому, программному и т. д.);

После обсуждения общих принципов формирования требований к информационной системе согласно ГОСТу 34.602-89, перейдем к разработке специальных требований для оперативно-аналитической подсистемы, ответственной за учет операций с контейнерами в рамках автоматизированной системы управления терминально-складским комплексом (АСУ ТСК).

Разработанная оперативно-аналитическая подсистема по учёту операций с контейнерами в составе АСУ ТСК должна быть реализована в программных средствах в соответствии с требованиями проводимой политики импортозамещения.

## **Требования к системе в целом**

### **Требования к структуре и функционированию системы**

Разработанная оперативно-аналитическая подсистема по учёту операций с контейнерами должна состоять из следующих справочных форм:

* книга выгрузки контейнеров ГУ-44к;
* книга завоза контейнеров;
* книга вывоза контейнеров;
* наличие контейнеров с грузом;

#### **Требования к функциональной структуре системы**

В разработанной подсистеме по учёту операций с контейнерами должны быть реализованы функции в соответствии с Таблицей 1

Таблица 1 – Реализуемые функции

|  |  |  |
| --- | --- | --- |
| Подсистема | Справочная форма | Функции |
| Оперативно-аналитическая подсистема по учёту операций с контейнерами. | Книга выгрузки контейнеров ГУ-44к. | 1. Введение следующих фильтров:  * Дата с – выбор из календаря или ввод вручную; * Дата по – выбор из календаря или ввод вручную; * ДМ – выбор из справочника (обязательный фильтр для выбора); * МЧ – выбор из справочника (обязательный фильтр для выбора); * МОП – выбор из справочника (обязательный фильтр для выбора).  1. Вывод сформированной справочной формы в составе следующих столбцов (Приложение А, таблица А.1.):  * № п/п; * Номер вагона; * Номер отправки; * Сведения о контейнере:   + Код владельца;   + Номер;   + Типоразмер; * Дата и время выгрузки; * Местонахождение контейнера; * Номер документа; * Дата и время закрытия учета; * Примечания. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Подсистема | Справочная форма | Функции |
| Оперативно-аналитическая подсистема по учёту операций с контейнерами. | Наличие контейнеров с грузом. | 1. Введение следующих фильтров:  * Дата с – выбор из календаря или ввод вручную; * Дата по – выбор из календаря или ввод вручную; * ДМ – выбор из справочника (обязательный фильтр для выбора); * МЧ – выбор из справочника (обязательный фильтр для выбора); * МОП – выбор из справочника (обязательный фильтр для выбора); * Наименование груза – выбор из справочника ЕТСНГ.  1. Вывод сформированной справочной формы в составе следующих столбцов (Приложение Б, таблица Б.1.):  * Код груза – отображается код груза по ЕТСНГ; * Наименование – отображается наименование груза по ЕТСНГ; * Кол-во контейнеров – расчет количества контейнеров по данным СК-1. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Подсистема | Справочная форма | Функции |
| Оперативно-аналитическая подсистема по учёту операций с контейнерами. | Книга завоза контейнеров. | 1. Введение следующих фильтров  * Дата с – выбор из календаря или ввод вручную; * Дата по – выбор из календаря или ввод вручную; * ДМ – выбор из справочника (обязательный фильтр для выбора); * МЧ – выбор из справочника (обязательный фильтр для выбора); * МОП – выбор из справочника (обязательный фильтр для выбора); * Состояние контейнера: * Все;   + Груженый;   + Порожний.  1. Вывод сформированной справочной формы в составе следующих столбцов (Приложение В, таблица В.1.):  * № п/п; * Номер контейнера; * Типоразмер; * Дата и время завоза или загрузки; * № КЭУ-16; * Код клиента; * Наименование клиента; * Вес брутто; * Номер автомобиля; * Водитель; * Примечание. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Подсистема | Справочная форма | Функции |
| Оперативно-аналитическая подсистема по учёту операций с контейнерами. | Книга вывоза контейнеров. | 1. Введение следующих фильтров  * Дата с – выбор из календаря или ввод вручную; * Дата по – выбор из календаря или ввод вручную; * ДМ – выбор из справочника (обязательный фильтр для выбора); * МЧ – выбор из справочника (обязательный фильтр для выбора); * МОП – выбор из справочника (обязательный фильтр для выбора); * Состояние контейнера: * Все;   + Груженый;   + Порожний.  1. Вывод сформированной справочной формы в составе следующих столбцов (Приложение Г, таблица Г.1.)  * № п/п; * Номер контейнера; * Типоразмер; * Дата и время вывоза или разгрузки; * № КЭУ-16; * Код клиента; * Наименование клиента; * Вес брутто; * Номер автомобиля; * Примечание. |

#### **Требования к программно-аппаратной архитектуре разрабатываемой системы**

Информационное взаимодействие между компонентами программного обеспечения должно осуществляться по технологии «тонкого клиента» с использованием базы данных АСУ ТСК.

Доступ пользователей к отчетности должен осуществляться по протоколу HTTP посредством Web-браузера.

#### **Перспективы развития и модернизации системы**

Подсистема для АСУ ТСК должна быть разработана как открытая система, позволяющая производить необходимые модернизации и развитие при минимальных трудозатратах. Для этого необходимо при создании ориентироваться на типовые решения, принятые для основных информационных технологий, по техническим средствам, ПО и обмену информацией.

Подсистема должна быть построена с использованием модульного принципа на основании открытых стандартов и предоставлять возможности модернизации и развития для улучшения визуальных и/или функциональных возможностей при минимальных трудозатратах, в том числе:

1) возможности доработки интерфейса системы и/или административных интерфейсов;

2) включение дополнительных функциональных модулей;

3) оптимизацию и горизонтальное масштабирование основных функциональных модулей системы.

### **Требования к численности и квалификации пользователей системы**

Пользователи АСУ ТСК должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне квалифицированного пользователя и свободно осуществлять базовые операции в стандартных Windows.

Пользователи системы должны быть ознакомлены с инструкциями по пользованию оперативно-аналитической подсистемы по учёту операций с контейнерами в составе АСУ ТСК.

### **Требования к показателям назначения**

Требования к временным характеристикам. Формирование справок, выписок не должно превышать 60 секунд.

### **Требования к эргономичности и технической эстетике**

Разрабатываемая оперативно аналитическая подсистема по учёту операций с контейнерами в составе АСУ ТСК должна обеспечивать удобный для пользователей интерфейс, отвечающий следующим требованиям:

* единый графический дизайн для пользовательских интерфейсов с одинаковым расположением основных элементов управления и навигации;
* взаимодействие пользователя с развиваемым программным обеспечением должно осуществляться на русском языке, исключения могут составлять только системные сообщения;
* информация в интерфейсе должна корректно отображаться при разных разрешениях дисплеев.

### **Требования к патентной чистоте**

Установка оперативно-аналитической подсистемы по учёту операций в контейнерах в составе АСУ ТСК не должна предъявлять дополнительных требований к покупке лицензий на программное обеспечение сторонних производителей, кроме программного обеспечения, указанного в разделе 2.3.4.

## **Требования к функциям (задачам), выполняемым разрабатываемой системой**

### **Справочная форма «Книга выгрузки контейнеров ГУ-44к»**

Должен быть разработан интерфейс справочной формы «Книга выгрузки контейнеров ГУ-44к». Данная справочная форма должна состоять из следующих столбцов:

− № п/п;

− Номер вагона;

− Номер отправки;

− Сведения о контейнере:

− Код владельца;

− Номер;

− Типоразмер;

− Дата и время выгрузки;

− Местонахождение контейнера;

− Номер документа;

− Дата и время закрытия учета;

− Примечания.

Справочная форма должна формироваться с 18:01 начальной даты по 18:00 конечной даты.

Также необходимо реализовать следующие фильтры:

− Дата с – выбор из календаря или ввод вручную;

− Дата по – выбор из календаря или ввод вручную;

− ДМ – выбор из справочника (обязательный фильтр для выбора);

− МЧ – выбор из справочника (обязательный фильтр для выбора);

− МОП – выбор из справочника (обязательный фильтр для выбора);

### **Справочная форма «Наличие контейнеров с грузом»**

Должен быть разработан интерфейс справочной формы «Наличие контейнеров с грузом». Данная справочная форма должна состоять из следующих столбцов:

− Код груза – отображается код груза по ЕТСНГ;

− Наименование – отображается наименование груза по ЕТСНГ;

− Кол-во контейнеров – расчет количества контейнеров по данным СК-1.

Также необходимо реализовать следующие фильтры:

− Дата с – выбор из календаря или ввод вручную;

− Дата по – выбор из календаря или ввод вручную;

− ДМ – выбор из справочника (обязательный фильтр для выбора);

− МЧ – выбор из справочника (обязательный фильтр для выбора);

− МОП – выбор из справочника (обязательный фильтр для выбора);

− Наименование груза – выбор из справочника ЕТСНГ.

### **Справочная форма «Книга завоза контейнеров»**

Должен быть разработан интерфейс справочной формы «Книга завоза контейнеров». Данная справочная форма должна состоять из следующих столбцов:

− № п/п;

− Номер контейнера;

− Типоразмер;

− Дата и время завоза или загрузки;

− № КЭУ-16;

− Код клиента;

− Наименование клиента;

− Вес брутто;

− Номер автомобиля;

− Водитель;

− Примечание.

Справочная форма должна формироваться с 18:01 начальной даты по 18:00 конечной даты.

Также необходимо реализовать следующие фильтры:

− Дата с – выбор из календаря или ввод вручную;

− Дата по – выбор из календаря или ввод вручную;

− ДМ – выбор из справочника (обязательный фильтр для выбора);

− МЧ – выбор из справочника (обязательный фильтр для выбора);

− МОП – выбор из справочника (обязательный фильтр для выбора);

− Состояние контейнера:

− Все;

− Груженый;

− Порожний.

### **Справочная форма «Книга вывоза контейнеров»**

Должен быть разработан интерфейс справочной формы «Книга вывоза контейнеров». Данная справочная форма должна состоять из следующих столбцов:

− № п/п;

− Номер контейнера;

− Типоразмер;

− Дата и время вывоза или разгрузки;

− № КЭУ-16;

− Код клиента;

− Наименование клиента;

− Вес брутто;

− Номер автомобиля;

− Примечание.

Справочная форма должна формироваться с 18:01 начальной даты по 18:00 конечной даты.

Также необходимо реализовать следующие фильтры:

− Дата с – выбор из календаря или ввод вручную;

− Дата по – выбор из календаря или ввод вручную;

− ДМ – выбор из справочника (обязательный фильтр для выбора);

− МЧ – выбор из справочника (обязательный фильтр для выбора);

− МОП – выбор из справочника (обязательный фильтр для выбора);

− Состояние контейнера:

− Все;

− Груженый;

− Порожний.

## **Требования к видам обеспечения**

### **Требования к математическому обеспечению**

Подсистема по учёту операций с контейнерами в составе АСУ ТСК должна базироваться на типовых методиках, моделях и алгоритмах, утверждённых ОАО «РЖД».

Разрабатываемые алгоритмы и программы должны обладать свойствами надёжности и устойчивости.

Используемые математические методы должны обеспечивать минимизацию времени расчёта, обмена информацией и используемых ресурсов программно-технического комплекса системы.

### **Требования к лингвистическому обеспечению**

Компоненты системы на стороне back-end должен быть написаны на языке программирования PHP версии не ниже 7 версии с использованием паттерна MVS (Model View Controller).

Языком запросов к реляционной базе данных PostgreSQL должен быть SQL−92 (Structured Query Language) или версией выше.

### **Требования к системному программному обеспечению**

Разрабатываемое программное обеспечение должно быть реализовано в программных средствах в соответствии с требованиями проводимой политики импортозамещения.

Требования к системному (обеспечивающему) ПО сервера приложений:

* docker-compose 1.25 (лицензирования не требуется).

Требования к системному (обеспечивающему) ПО сервера баз данных:

* СУБД PostgreSQL 12 и выше (лицензирования не требуется).

Требования к клиентской части:

* Операционная система Windows 7 или более новая версия;
* Microsoft Offiсe версия 2007 и выше;
* Яндекс.Браузер версии 17.4.1 и выше.

АРМ пользователя должно быть реализовано по технологии тонкого клиента посредством интернет-браузера и не должно требовать отдельной установки на рабочее место пользователя.

# **Разработка программного обеспечения оперативно-аналитической подсистемы по учёту операций с контейнерами**

## **Описание функционала**

В оперативно-аналитической подсистеме по учёту операций с контейнерами в составе АСУ ТСК будут разработаны следующие справочные формы:

* Книга выгрузки контейнеров ГУ-44к;
* Наличие контейнеров с грузом;
* Книга завоза контейнеров;
* Книга вывоза контейнеров.

Основной функционал отчетов был детально описан в разделе «Требования к функциям, выполняемым разрабатываемой системой». В данном разделе рассматриваются такие аспекты, как состав обязательных фильтров, необходимых для формирования справочных форм, возможность множественного выбора в некоторых фильтрах, а также формат ввода дат и других параметров.

Однако, помимо описанных функций в вышеупомянутом разделе, существуют и другие важные аспекты работы системы. Каждый отчет содержит обязательные фильтры, которые пользователи должны заполнить перед формированием отчета. В случае, если пользователь пытается сформировать справочную форму без заполнения этих обязательных фильтров, система должна отображать сообщение: «Не заполнены обязательные фильтры для отчётной формы».

Если при выполнении запроса к базе данных не удается найти соответствующую информацию, система должна выводить сообщение: «Данные не найдены».

Кроме того, необходимо предусмотреть обработку ошибок, которые могут возникнуть на серверной стороне. Например, возможны ситуации, когда не удается вызвать ту или иную функцию. В таких случаях система должна корректно обрабатывать эти ошибки и предоставлять пользователю соответствующую информацию о произошедшей ошибке.

## **Моделирование и визуализация с помощью StarUML**

StarUML – это сложное программное обеспечение, предназначенное для поддержки гибкого и краткого моделирования [2]. В данной работе будем использовать программное обеспечение StarUML для моделирования и визуализации различных аспектов системы. StarUML предоставит нам возможность создавать диаграммы классов, диаграммы последовательностей и другие типы UML-диаграмм, которые помогут структурировать и анализировать архитектуру разрабатываемого программного обеспечения. Мы начнем с создания диаграмм последовательностей для визуализации взаимодействия между компонентами системы во времени, после чего перейдем к разработке диаграмм классов, чтобы определить структуру и отношения между объектами.

### **Диаграмма последовательности (Sequence Diagram)**

Сейчас мы приступим к описанию диаграммы последовательности. Эти диаграммы предназначены для визуализации обменов между объектами системы в хронологическом порядке. Они отображают, какие объекты коммуницируют друг с другом и в каком порядке происходят эти обмены.

Рассмотрим особенности создаваемой оперативно-аналитической подсистемы, уделяя особое внимание архитектуре серверной части кода. Подсистема будет разрабатываться с использованием шаблонного типа паттерна MVC (Model View Controller). На диаграмме (рисунок 4) показано, что при отправке HTTP-запроса на сервер его перехватывает класс Route, который вызывает действие контроллера. Контроллер, в свою очередь, обращается к методу класса Model, который получает данные для справочной формы из базы данных. После получения необходимой информации она отображается пользователю.

На основе данной диаграммы будут разрабатываться все справочные формы в подсистеме, предназначенной для учёта операций с контейнерами. Использование диаграммы последовательности и паттерна MVC обеспечивает чёткую и структурированную архитектуру системы, способствующую эффективной обработке данных и их отображению пользователям.

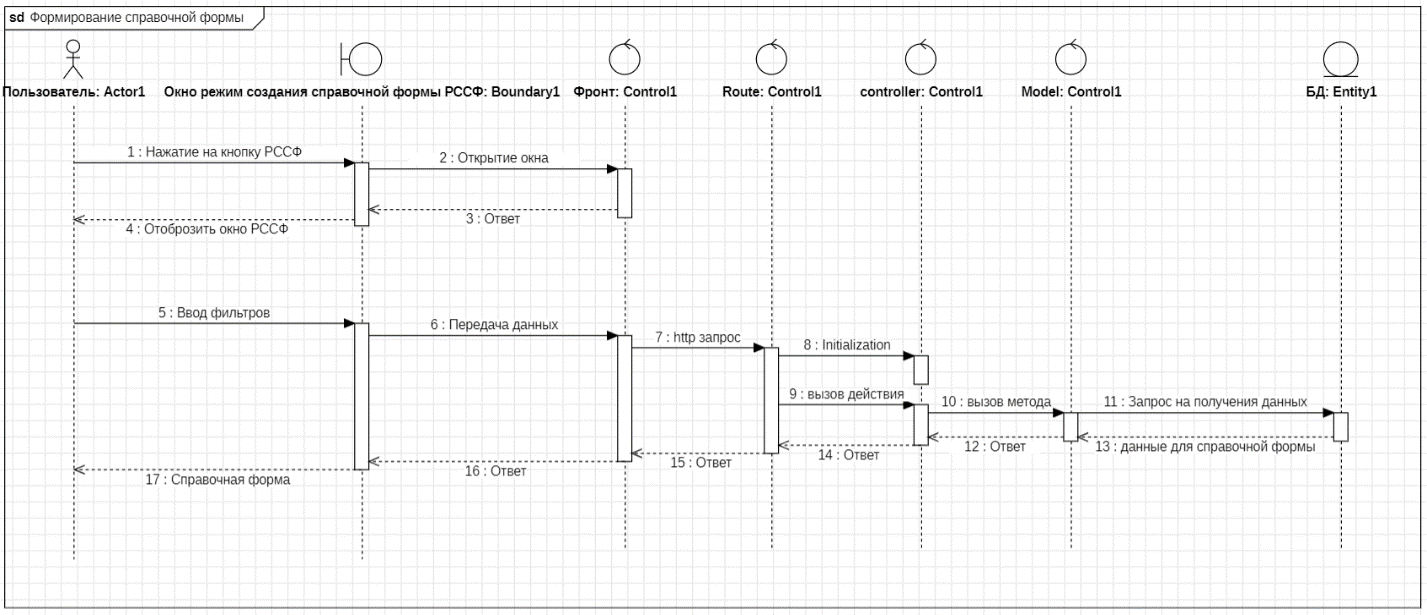


Рисунок 4 – Диаграмма последовательности. Формирование справочной формы

### **Диаграмма классов (Class Diagrams)**

Диаграммы классов представляют собой визуальное отображение структуры оперативно-аналитической подсистемы, выделяя классы, их атрибуты и методы, а также отношения между ними. Эти диаграммы помогают понять, как компоненты системы взаимодействуют друг с другом и как они организованы внутри. Они являются мощным инструментом для проектирования архитектуры системы и обеспечивают понимание её структуры разработчикам и архитекторам.

На рисунке 5 как раз показывается как описываются классы с их атрибутами и методами, которые используются для формирования справочных форм на стороне серверной части – back-end.

В классе Route\_Report атрибутом выступает маршрут (Route), который содержит в себе HTTP-запрос. При отправке данного HTTP-запроса на сервер, вызываются методы данного класса. Метод controller инициализирует класс Controller\_Report, а метод action вызывает действие этого инициализированного класса. Функция auth указывает на включенную аутентификацию – для выполнения действия пользователь должен быть аутентифицирован. Параметр pathType обозначает использование определенного типа пути или маршрута.

Класс Controller\_Report вызывает метод класса Model\_Report через функцию listAction, где в качестве атрибута выступает request, содержащий массив данных, необходимых для формирования отчета. Данный класс является наследником класса Controller.

Model\_Report, в свою очередь, является наследником ModelAbstraction, используя атрибуты и методы класса ConditionHelper. С помощью метода list и атрибута CONDITION\_SCHEMA, который представляет собой ассоциативный массив, реализуется SQL-запрос в базу данных.

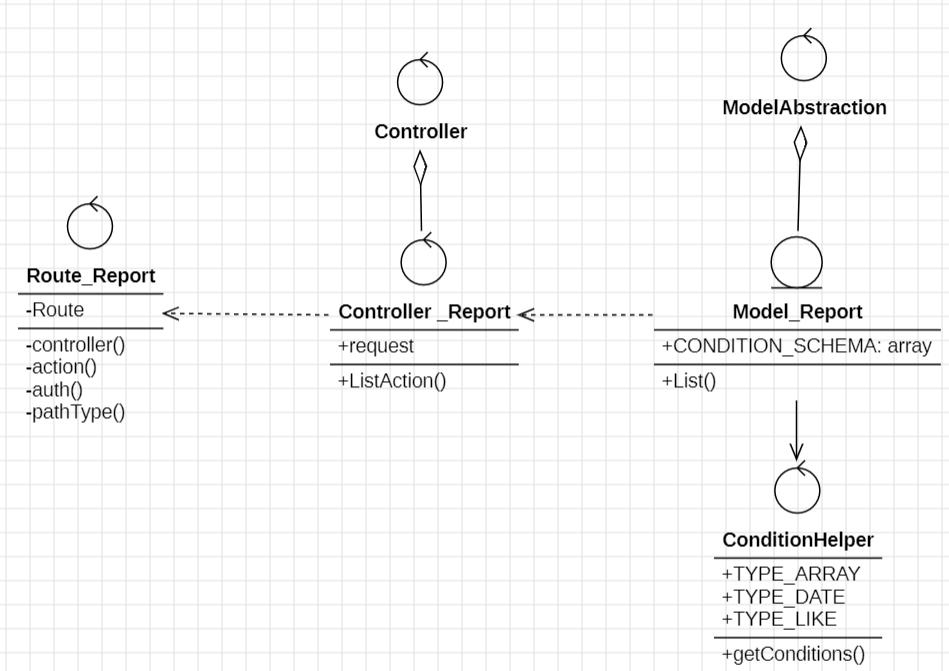


Рисунок 5 − Диаграмма классов. Формирование справочной формы

## **Программная реализация веб-приложения**

Подсистема по учёту операций с контейнерами разделена на следующие составляющие, фронтенд и бэкенд. Разработка программного обеспечения будет производиться на основе архитектуры тонкого клиента, где основная вычислительная нагрузка и управление данными будут осуществляться на серверной стороне, обеспечивая высокую производительность и безопасность.

Бэкенд — это серверная часть программного обеспечения, которая отвечает за обработку логики приложения, управление базами данных и взаимодействие с фронтендом через API. Данное программное обеспечение разработано на языке программирования PHP, который широко используется благодаря своей простоте и эффективности при создании веб-приложений. При разработке использовался паттерн Model-View-Controller (MVC). Для реализации подсистемы по учету операций с контейнерами были созданы диаграммы последовательности и классов, на основе которых был написан соответствующий код. Этот код, демонстрирующий работу системы и взаимодействие её компонентов, можно посмотреть в приложении И.

Фронтенд — это часть приложения, с которой взаимодействуют пользователи, обеспечивая отображение данных и пользовательский интерфейс. В данном проекте интерфейс был написан c использованием React и Redux. Для Автоматизированной системы управления терминально-складским комплексом были уже реализованные необходимые библиотеки, обеспечивающие удобное и эффективное создание интерфейса. Исходный код можно посмотреть в приложении Ж.

В рамках создания оперативно-аналитической подсистемы были задействованы несколько инструментов, каждый из которых играл важную роль в процессе разработки и обеспечивал определенные функциональные возможности.

Postman – это инструмент для тестирования веб-сервисов, который позволяет разработчикам создавать, тестировать, отлаживать и документировать API. С его помощью были проведены тестирование и отладка веб-сервисов, взаимодействующих с оперативно-аналитической подсистемой. Postman обеспечил эффективное взаимодействие с API и проверку корректности передачи данных между различными компонентами системы.

VSCode – это среда разработки с открытым исходным кодом, предоставляющая широкий спектр функциональных возможностей, включая подсветку синтаксиса, автодополнение кода, отладку и интеграцию с системами контроля версий. VSCode использовался для написания и редактирования исходного кода оперативно-аналитической подсистемы, что способствовало повышению производительности разработки и обеспечивало комфортное рабочее окружение.

Docker-compose – это инструмент для определения и запуска многоконтейнерных Docker-приложений. С помощью docker-compose разворачивались и масштабировались контейнеры, включающие компоненты подсистемы по учёту операций с контейнерами в составе АСУТСК, что обеспечивало удобное управление окружением разработки и упрощало развертывание приложения.

Yarn – это пакетный менеджер для управления зависимостями JavaScript. Yarn использовался для установки и управления зависимостями проекта оперативно-аналитической подсистемы.

DBeaver – это инструмент для управления базами данных с открытым исходным кодом. С его помощью осуществлялось подключение к базе данных PostgreSQL, это позволило проводить мониторинг и выполнение SQL-запросов к базе данных АСУТСК.

Эти инструменты в совокупности предоставили полный цикл средств для разработки, тестирования, отладки и управления зависимостями в рамках создания подсистемы по учёту операций с контейнерами, обеспечивая эффективное и продуктивное рабочее окружение.

### **SQL Запросы для формирования справочных форм**

В ходе процесса составления отчетов использовалась разработанная база данных (БД), специально созданная для АСУТСК, с учетом требований предприятия. Данная база данных храниться в системе управления базами данных (СУБД) PostgreSQL. Исходя из информации, сохраненной в БД АСУТСК, составляются справочные формы.

Для более полного и точного понимания структуры и взаимосвязей данных, содержащихся в базе данных АСУТСК, была разработана модель сущность-связь или ER-модель (рисунок 6). Для этого использовалась специализированная программа для проектирования баз данных, такая как ERWin, которая позволяет создавать и визуализировать ER-диаграммы, отображающие сущности, их атрибуты и связи между ними.

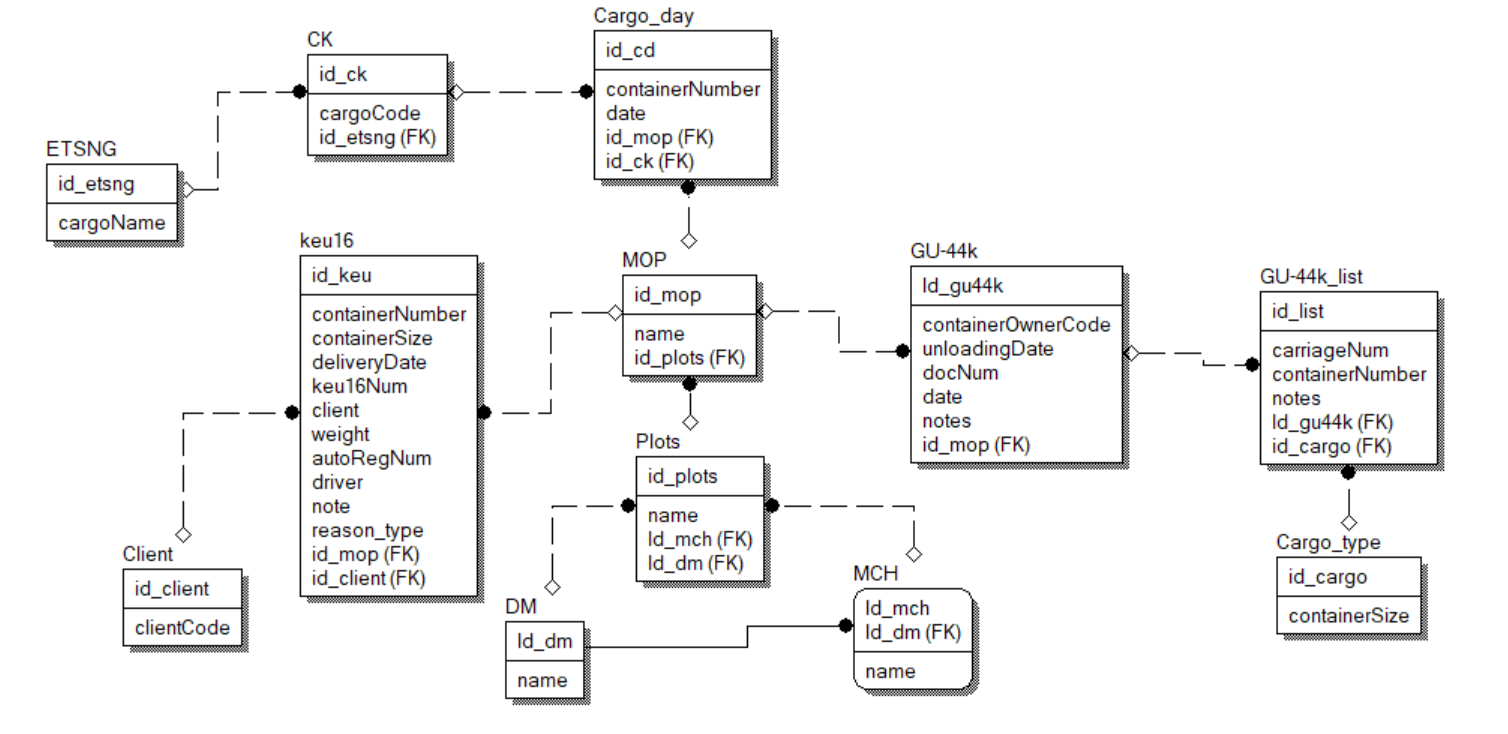


Рисунок 6 – ER-модель

В разработанной подсистеме по учёту операций с контейнерами были реализованы SQL запросы для формирования справочных форм, в соответствии с таблицей 2. Сущности и их атрибуты были переименованы в связи с конфиденциальностью данных.

Таблица 2 – SQL запросы

|  |  |
| --- | --- |
| Справочная форма | SQL запрос |
| Книга выгрузки контейнеров ГУ-44к. | select  row\_number() over() as num  , gl.carriageNum  , g.containerOwnerCode  , gl.containerNumber  , ct.containerSize  , g.unloadingDate  , mop.name as location  , g.docNum  , g.date  , gl.notes  from GU-44k g  inner join GU-44k\_list gl on gl.id\_gu44k = g.id\_gu44k  inner join mop on mop.id\_mop = g.mop\_id  left join Plots p on p.id\_plots = mop.id\_plots  left join DM dm on dm.id\_dm = p.id\_dm  left join MCH mch on mch.id\_mch = dp.id\_mch  left join Cargo\_type ct on ct.id\_cargo = gl.id\_class |
| Наличие контейнеров с грузом. | select  ck.cargoCode,  e.cargoName,  count (distinct cd.containerNumber) as containerCount  from Cargo\_day cd  left join mop on mop.id\_mop = scd.mop\_id  left join Plots p on p.id\_plots = mop.id\_plots  left join DM dm on dm.id\_dm = p.id\_dm  left join MCH mch on mch.id\_mch = dp.id\_mch  left join CK ck on ck.id\_ck = cd.id\_ck  left join ENTSNG e on e.id\_etsng = ck.id\_etsng  group by  ck. ck.cargoCode  , e.cargoName |

Продолжение таблицы 2

|  |  |
| --- | --- |
| Справочная форма | SQL запрос |
| Книга завоза контейнеров. | select  row\_number() OVER(ORDER BY keu.id DESC) num  , k.containerNumber  , k.containerSize  , k.deliveryDate  , k.keu16Num  , c.clientCode  , с.client  , k.weight  , k.autoRegNum  , k.driver  , k.note  from keu16 k  inner join mop on mop.id = k.mop  left join Plots p on p.id\_plots = mop.id\_plots  left join DM dm on dm.id\_dm = p.id\_dm  left join MCH mch on mch.id\_mch = dp.id\_mch  left join client c on c.id\_client = k.id\_client  where k.reason\_type = 2 -- только на приём |
| Книга вывоза контейнеров. | select  k.id\_keu as num  , k.containerNumber  , k.containerSize  , k.deliveryDate  , k.keu16Num  , c.clientCode  , k.client  , k.weight  , k.autoRegNum  , k.driver  , k.note  from keu16 k  left join mop on mop.id = k.mop  left join Plots p on p.id\_plots = mop.id\_plots  left join DM dm on dm.id\_dm = p.id\_dm  left join MCH mch on mch.id\_mch = dp.id\_mch  left join client c on c.id\_client = k.id\_client  where k.reason\_type = 1 -- только на выдачу |

## **Пользовательский интерфейс**

### **Выбор справочной формы**

После авторизации, пользователь может выбрать необходимое ему действие. В рамках данной системы оперативно-аналитической подсистемы по учету операций с контейнерами создаются такие справочные формы, как «Книга выгрузки контейнеров ГУ-44к», «Наличие контейнеров с грузом», «Книга завоза контейнеров» и «Книга вывоза контейнеров». Для доступа к этим формам необходимо открыть раздел «Отчёты», затем перейти в подраздел «Оперативная деятельность». В данном подразделе разработанные справочные формы будут представлены в алфавитном порядке (рисунок 7).

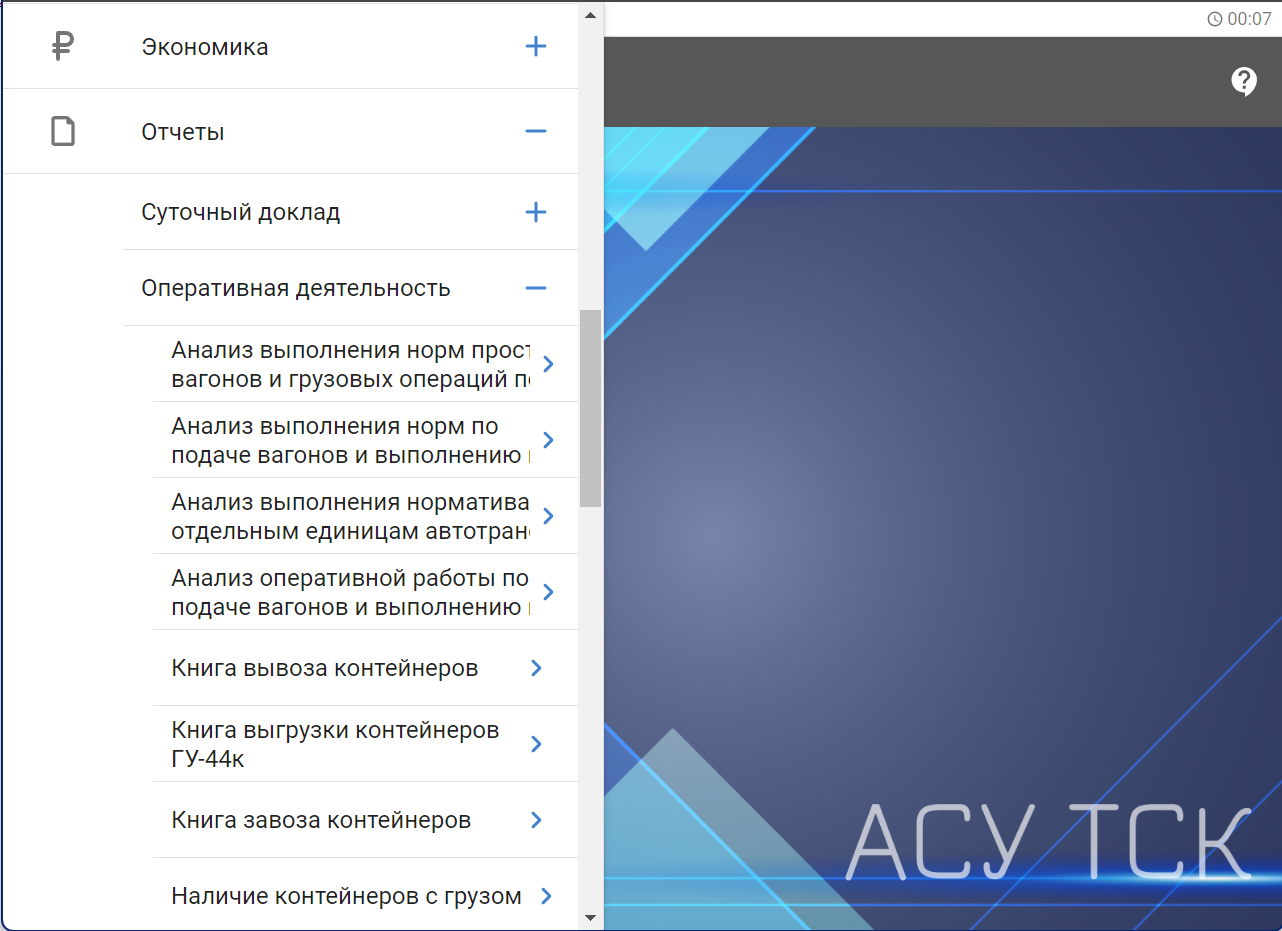


Рисунок 7 – Выбор справочной формы

### **Фильтры**

Большая часть фильтров в справочных формах повторяется. При использовании этих фильтров осуществляется ввод параметров, на основании которых формируются справочные формы. Для составления отчётов, используются следующие фильтры:

* ДМ – дирекция по управлению терминально-складским комплексом (Приложение Д, рисунок Д.1. и рисунок Д.2);
* МЧ – механизированная дистанция погрузочно-разгрузочных работ и коммерческих операций (Приложение Д, рисунок Д.3. и рисунок Д.4);
* МОП – место общего пользования (Приложение Д, рисунок Д.5.);
* Дата с (Приложение Д, рисунок Д.6.);
* Дата по (Приложение Д, рисунок Д.7.);
* Наименование груза: указывается наименование груза из справочника единой тарифно-статистической номенклатуры грузов (Приложение Д, рисунок Д.8.);
* Состояние контейнера: указывается, является ли контейнер груженым или порожним (Приложение Д, рисунок Д.9. и рисунок Д.10).

В том случае, если обязательные фильтры не будут заполнены, и пользователь решит сформировать отчёт, то данные фильтры окрасятся в красный цвет (рисунок 8) и выведется сообщение «Не заполнены обязательные фильтры для отчётной формы» (рисунок 9).

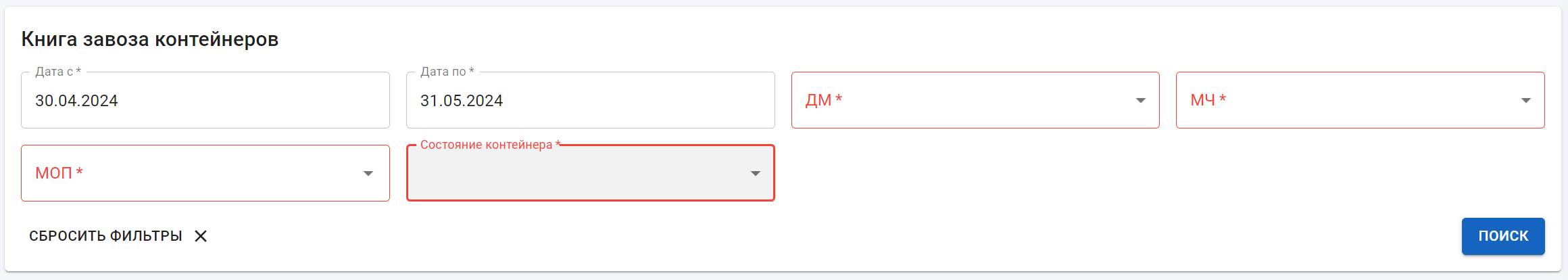


Рисунок 8 – Обязательные фильтры для заполнения.

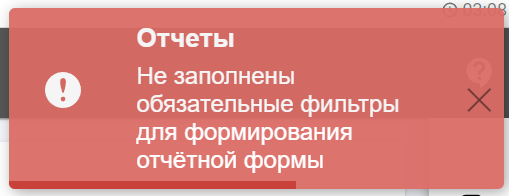


Рисунок 9 – Вывод сообщения. Не заполнены обязательные фильтры для формирования отчётной формы

### **Сообщение «Данные не найдены»**

При формировании справочной формы, может возникнуть ситуация, когда по заданным параметрам нет информации в базе данных. В этом случае выводится сообщение «Данные не найдены» (рисунок 10).

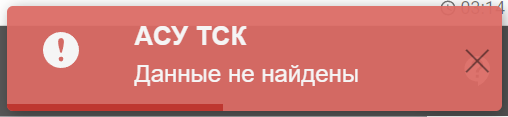


Рисунок 10 – Вывод сообщения. Данные не найдены

### **Страница справочной формы** **«Книга выгрузки контейнеров ГУ-44к»**

На рисунке 11 изображена страница справочной формы «Книга выгрузки контейнеров ГУ-44к».

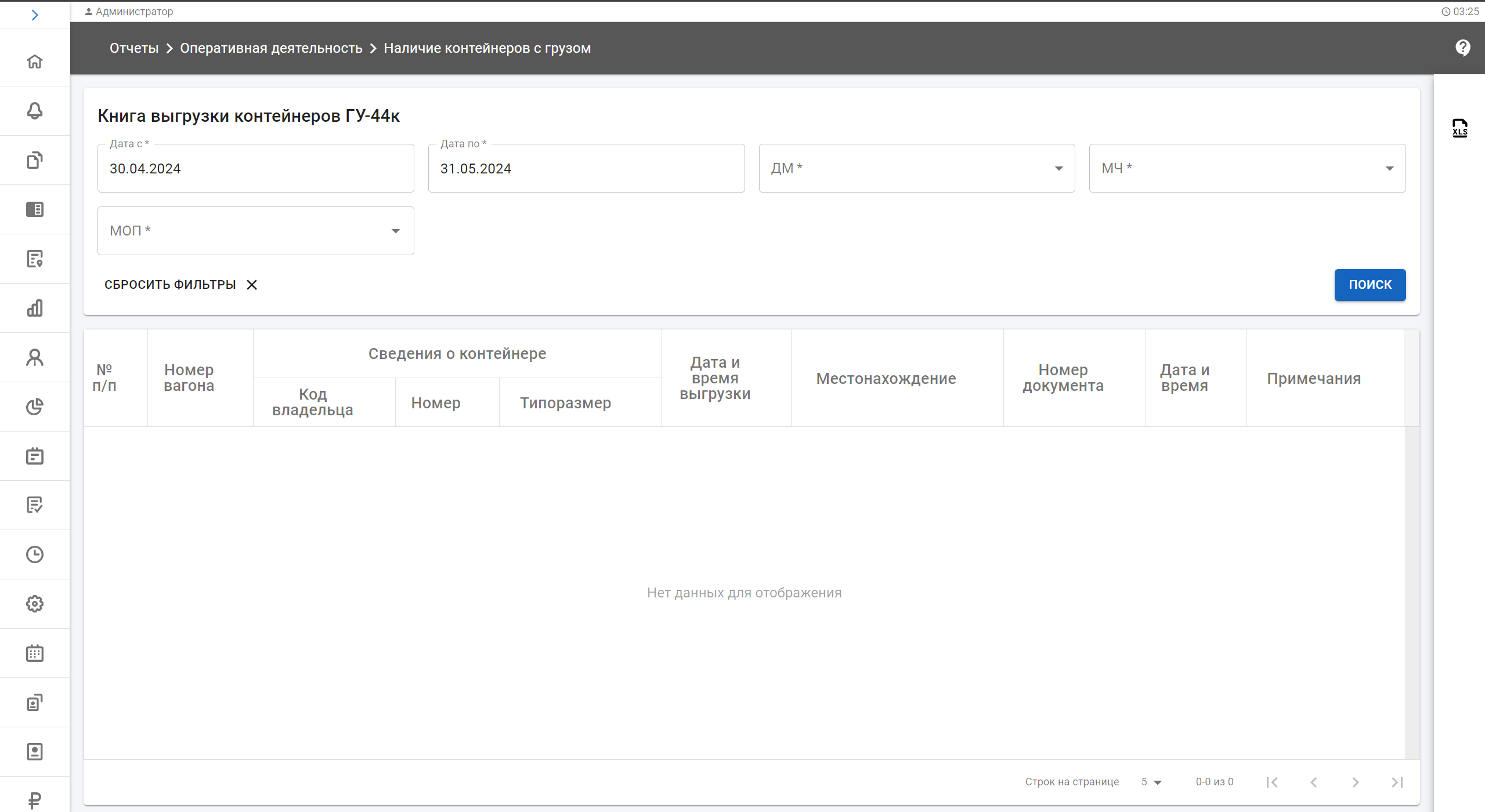
****

Рисунок 11 – Страница справочной формы. Книга выгрузки контейнеров ГУ-44к

### **Страница справочной формы «Наличие контейнеров с грузом»**

На рисунке 12 изображена страница справочной формы «Наличие контейнеров с грузом».

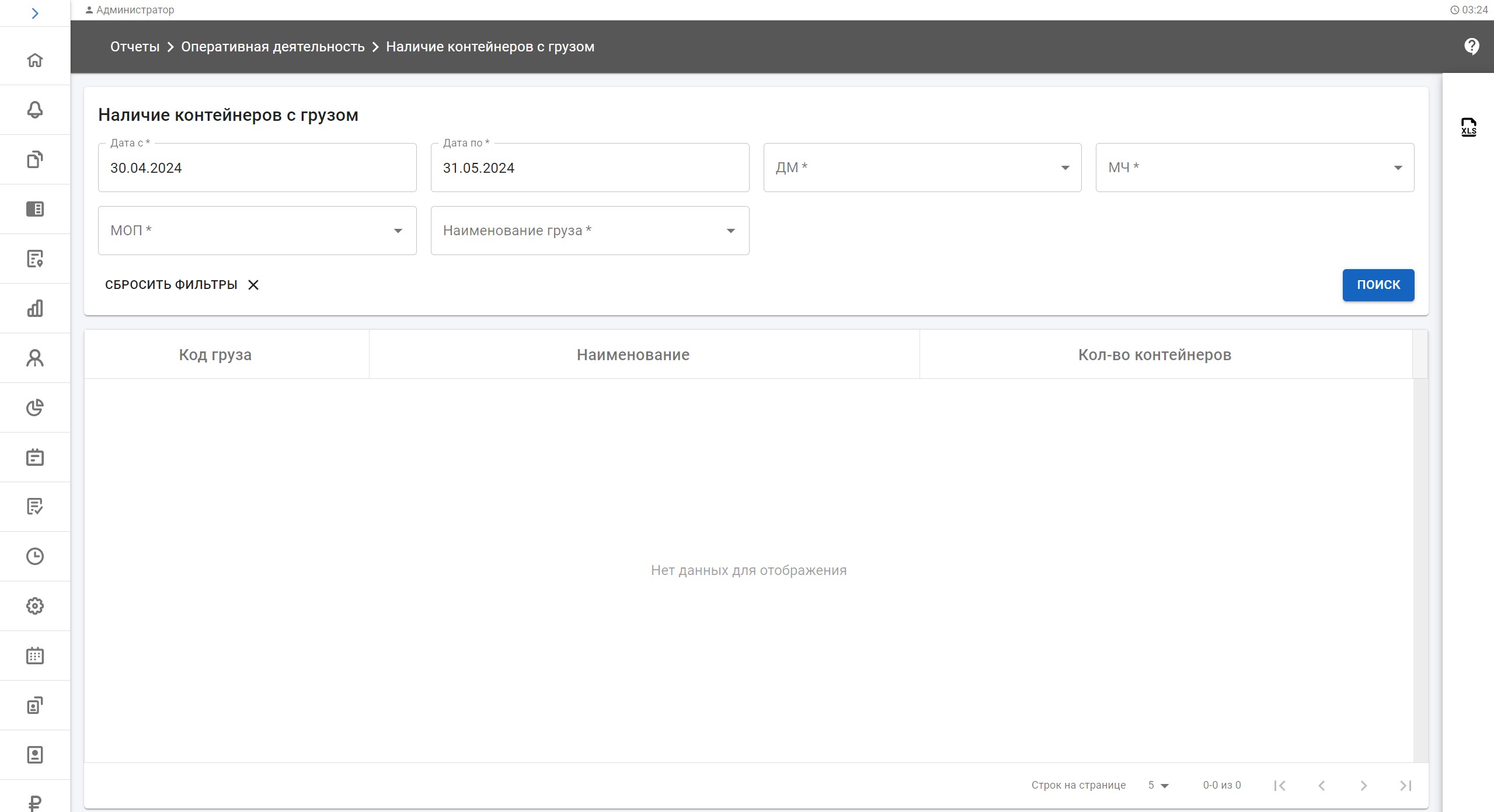
****

Рисунок 12 – Страница справочной формы. Наличие контейнеров с грузом

### **Страница справочной формы «Книга завоза контейнеров»**

На рисунке 13 изображена страница справочной формы «Книга завоза контейнеров».

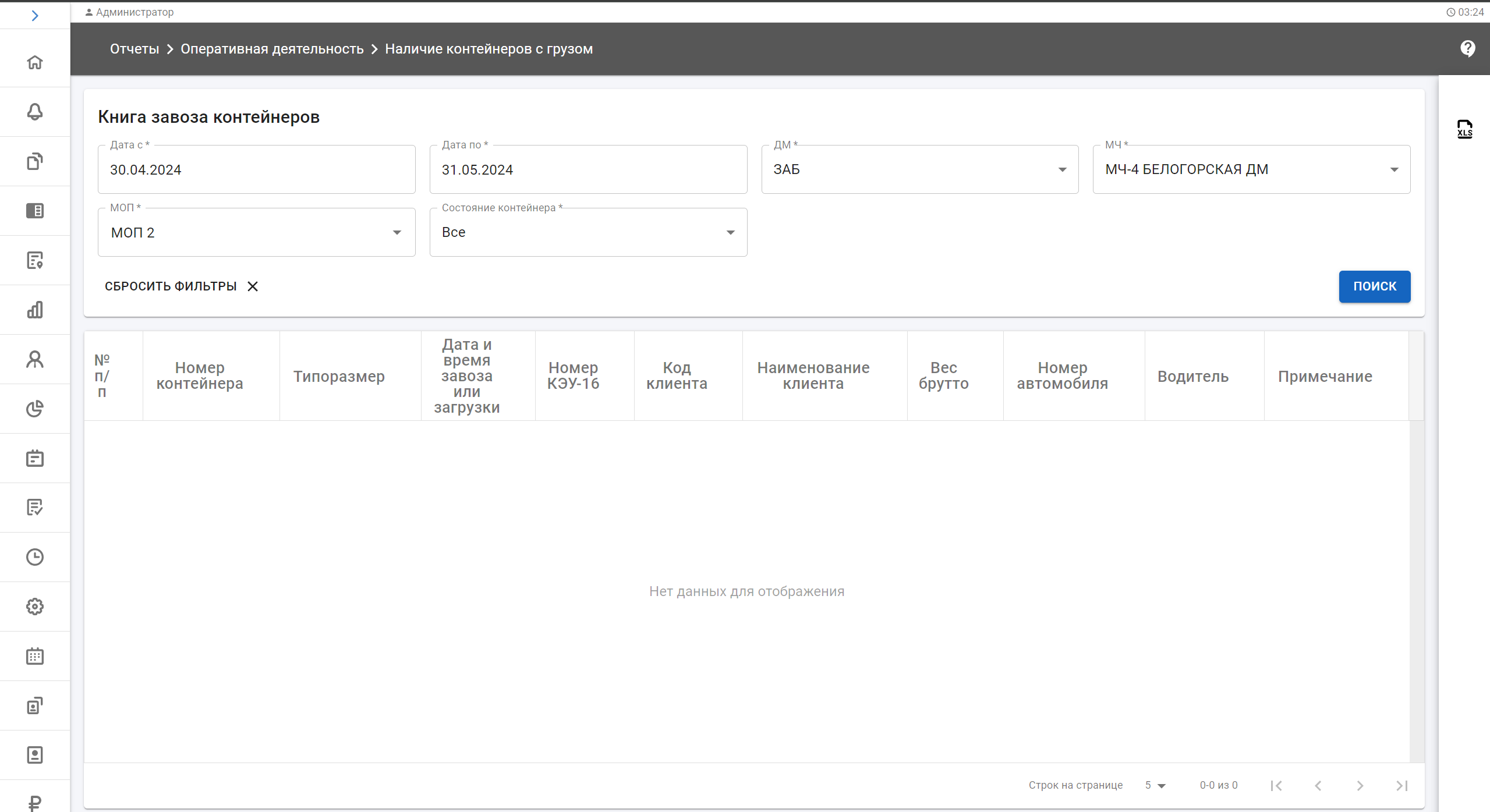
****

Рисунок 13 – Страница справочной формы. Книга завоза контейнеров

### **Страница справочной формы «Книга вывоза контейнеров»**

На рисунке 14 изображена страница справочной формы «Книга вывоза контейнеров».

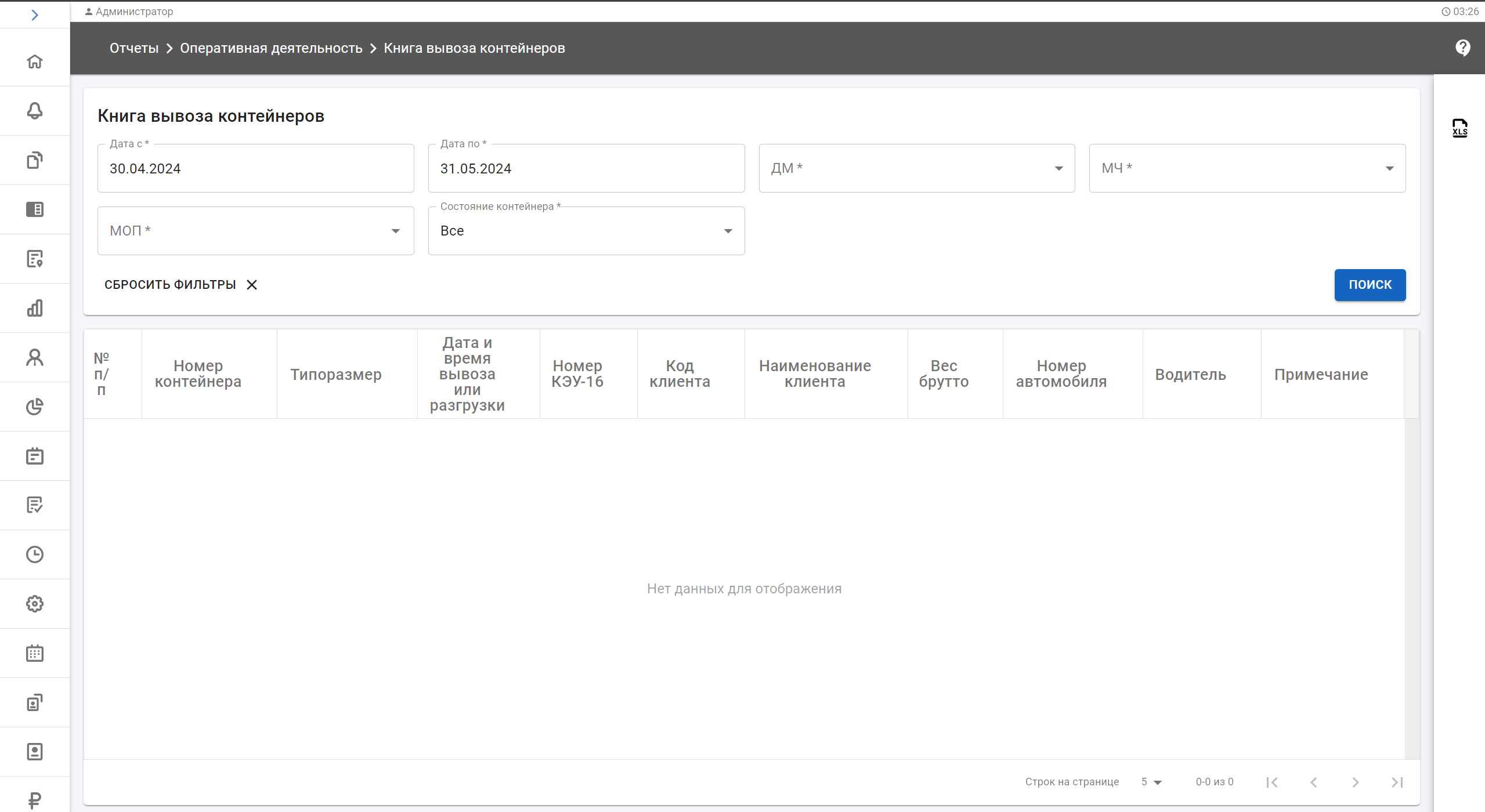


Рисунок 14 – Страница справочной формы. Книга вывоза контейнеров

# **Оценка разработанной подсистемы**

В данной главе будет произведено исследование, целью которой является экспериментальное определение среднего времени обработки операций по формированию справочных форм в разработанной подсистеме. В соответствии с поставленными требованиями к показателям назначения, формирование справок, выписок не должно превышать 60 секунд. Для достижения этой цели будет проведен анализ времени выполнения операций по формированию справочных форм для каждого отчета. Полученные данные будут использоваться для сравнительного анализа с установленным пороговым значением в 60 секунд.

При проведении эксперимента в рамках формирования отчёта использовался инструмент браузерной среды DevTools для измерения различных временных характеристик операций. На рисунке 15 представлена диаграмма операций. Были включены следующие параметры измерения:

* Время на передачу запроса от клиента к серверу (t1);
* Время выполнения запроса на сервере (t2);
* Время на передачу ответа от сервера к клиенту (t3).

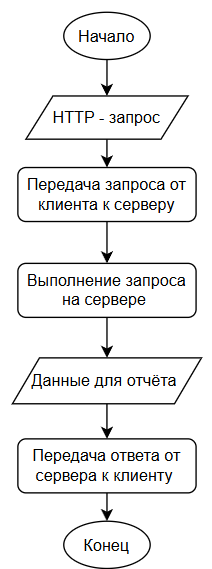


Рисунок 15 – Диаграмма операций

Для каждой справочной формы было проведено по 30 экспериментов. Все результаты экспериментов представлены в приложении E (таблица E.1). На основании полученных данных были вычислены средние значения и дисперсии *di* для параметров t1, t2 иt3 по формуле (1) и (2) соответственно. Поскольку длительности последовательных операций при формировании справочных форм являются независимыми величинами, среднее время формирования отчёта можно определить путем сложения средних значений этих параметров, аналогично и для дисперсии , и для среднеквадратического отклонения (таблица 3). Все измерения представлены в секундах.

где *n–* количество экспериментов;

*xij –* время j-го эксперимента по i-ому параметру измерения;

*k –* номер отчёта;

*–* среднеквадратичное отклонение по i-ому параметру измерения.

Таблица 3 – Сводка статистических данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Справочная форма | №1 – Книга выгрузки контейнеров ГУ-44к | №2 – Наличие контейнеров с грузом | №3 – Книга вывоза контейнеров | №4 – Книга завоза контейнеров |
| Величины |
|  | 0,622333333 | 0,659 | 0,492 | 0,562333333 |
|  | 11,73966667 | 12,021 | 9,293333333 | 10,53433333 |
|  | 4,694333333 | 4,015 | 2,660666667 | 2,772333333 |
|  | 0,007524556 | 0,014275667 | 0,007616 | 0,005024556 |
|  | 5,924729889 | 3,182342333 | 1,707402222 | 1,356277889 |
|  | 1,902897889 | 1,471811667 | 0,103299556 | 0,422157889 |
|  | 0,086744196 | 0,119480821 | 0,087269697 | 0,070884099 |
|  | 2,434076804 | 1,783912087 | 1,30667602 | 1,164593444 |
|  | 1,379455649 | 1,213182454 | 0,321402482 | 0,649736784 |
|  | 17,05633333 | 16,695 | 12,446 | 13,869 |
|  | 7,835152333 | 4,668429667 | 1,818317778 | 1,783460333 |
|  | 2,799134211 | 2,160654916 | 1,348450139 | 1,335462592 |

Для того чтобы определить, соответствует ли разработанная подсистема по учёту операций с контейнерами заданным требованиям по среднему времени выполнения операций, необходимо провести анализ вероятности выполнения всех отчётов в пределах заданного времени. Для этого преобразуем время формирования справочных форм , в стандартизированную нормальную переменную с использованием формулы (7). Это преобразование переводит нормальный случайный процесс с параметрами  и в стандартное нормальное распределение с параметрами 0 и 1.

По найденному значению рассчитаем искомую вероятность по разным значениям *U* для всех отчётов c помощью функции стандартного нормального распределения (8), использовав таблицу значений функции Лапласа. Результаты вычислений представлены в таблице 4.

Где *U –* заданное время, по которому будет оцениваться вероятность.

Таблица 4 – вероятность выполнения опреаций менее заданного времени

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Книга выгрузки контейнеров ГУ-44к | Наличие контейнеров с грузом | Книга вывоза контейнеров | Книга выгрузки контейнеров |
|  | 0,005853 | 0,00097 | 0,03495 | 0,00188 |
|  | 0,23134 | 0,21678 | 0,97016 | 0,80119 |
|  | 0,85352 | 0,93679 |  | 0,999997 |
|  | 0,99766 |  |  |  |

Таким образом, вероятность того, что подсистема будет формировать справочные формы менее чем за 60 секунд, составляет приблизительно 0,999999. Этот анализ позволяет с уверенностью подтвердить, что разработанная оперативно-аналитическая подсистема по учёту операций с контейнерами в составе АСУ ТСК соответствует заданным требованиям по времени выполнения.

# **ЗАКЛЮЧЕНИЕ**

В заключение можно отметить, что исследование, посвящённое созданию оперативно-аналитической подсистемы учёта операций с контейнерами в составе АСУТСК, продемонстрировало важность модернизации существующих систем управления ТСК. Проведённый анализ текущего состояния учёта операций в АРМПКП выявил ключевые проблемы и недостатки, что позволило предложить эффективное решение в виде новой подсистемы.

Рассмотрение и применение современных инструментов разработки, таких как «VS Code», «Postman» и «Star UML», а также акцент на использовании веб-технологий и интуитивно понятного интерфейса, способствовали созданию качественного и функционального программного обеспечения. Практическая часть работы, включающая разработку интерфейса и серверной части с использованием диаграмм классов и последовательности, обеспечила реализацию всех поставленных требований и предоставила ценный практический опыт в области разработки программных систем.

Особое внимание в исследовании было уделено производительности подсистемы, что позволило экспериментально определить среднее время обработки формирования отчётов и найти вероятность того, что подсистема будет оставлять справочные формы менее чем за 60 секунд. Результаты анализа временных характеристик подтверждают соответствие разработанной подсистемы заданным требованиям.

Таким образом, выполненная работа решает актуальные проблемы учёта операций с контейнерами в ТСК, полученные результаты и разработанные решения могут быть использованы в дальнейших исследованиях и практической деятельности для повышения эффективности и оптимизации работы терминально-складских комплексов.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. ГОСТ 7.32-2017 Отчёт о научно-исследовательской работе
2. Моделирование бизнес-процессов в нотации BPMN: Пособие для начинающих. Часть I / Владимир Репин. – [б. м.]: Издательские решения, 2019. – 84 с.
3. Смирнова А.В., Черноносова Н.В. Логистика складирования: Учебное пособие. – М.: РУТ (МИИТ), 2018. – 53 с.
4. ГОСТ 34.602-2020 Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы
5. StarUML documetation. – Текст: электронный // StarUML: [сайт]. – URL: https://docs.staruml.io/

# **ПРИЛОЖЕНИЕ А** Форма «Книга выгрузки контейнеров ГУ-44К»

Таблица А.1. − Форма «Книга выгрузки контейнеров ГУ-44К»

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Номер вагона | Сведения о контейнере | | | Дата и время выгрузки | Местонахождение | Номер документа | Дата и время | Примечания |
| Код владельца | Номер | Типоразмер |
|  |  |  |  |  |  |  |  |  |  |

# **ПРИЛОЖЕНИЕ Б** Форма «Наличие контейнеров с грузом»

Таблица Б.1. − Форма «Наличие контейнеров с грузом»

|  |  |  |  |
| --- | --- | --- | --- |
| МОП | Код груза | Наименование | Кол-во контейнеров |
|  |  |  |  |

# **ПРИЛОЖЕНИЕ В** Форма «Книга завоза контейнеров»

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Номер контейнера | Типоразмер | Дата и время вывоза или разгрузки | Номер КЭУ-16 | Код клиента | Наименование клиента | Вес брутто | Номер автомобиля | Водитель | Примечание |
|  |  |  |  |  |  |  |  |  |  |  |

Таблица В.1. − Форма «Книга завоза контейнеров»

# **ПРИЛОЖЕНИЕ Г**Форма «Книга вывоза контейнеров»

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Номер контейнера | Типоразмер | Дата и время вывоза или разгрузки | Номер КЭУ-16 | Код клиента | Наименование клиента | Вес брутто | Номер автомобиля | Водитель | Примечание |
|  |  |  |  |  |  |  |  |  |  |  |

Таблица Г.1. − Формы «Книга вывоза контейнеров»

# **ПРИЛОЖЕНИЕ Д**Интерфейс, отвечающий за фильтры.

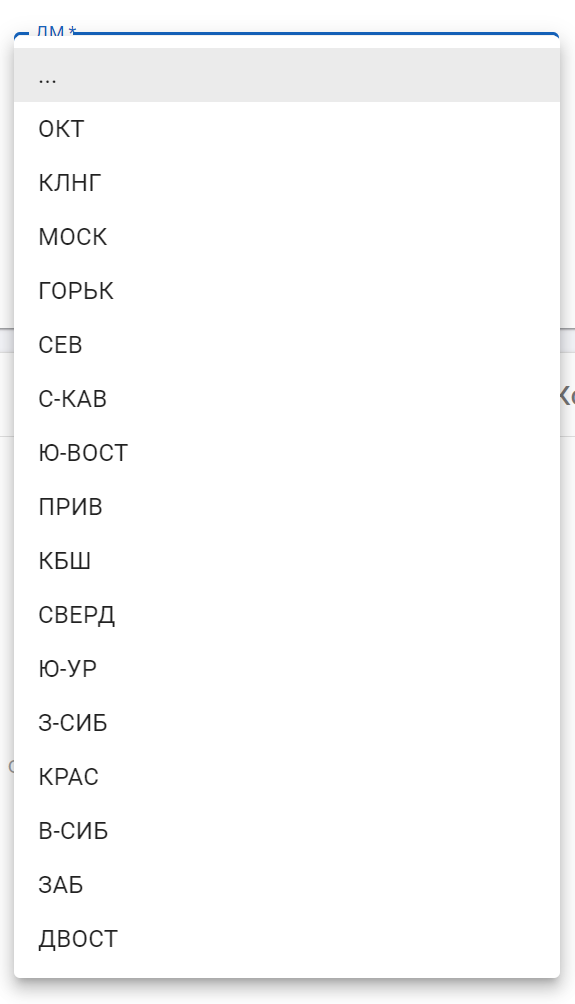


Рисунок Д. 1. – Справочник ДМ

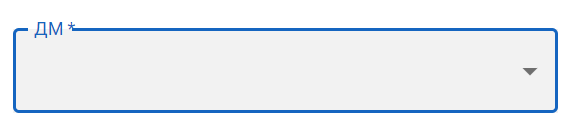


Рисунок Д. 2. – Фильтр ДМ

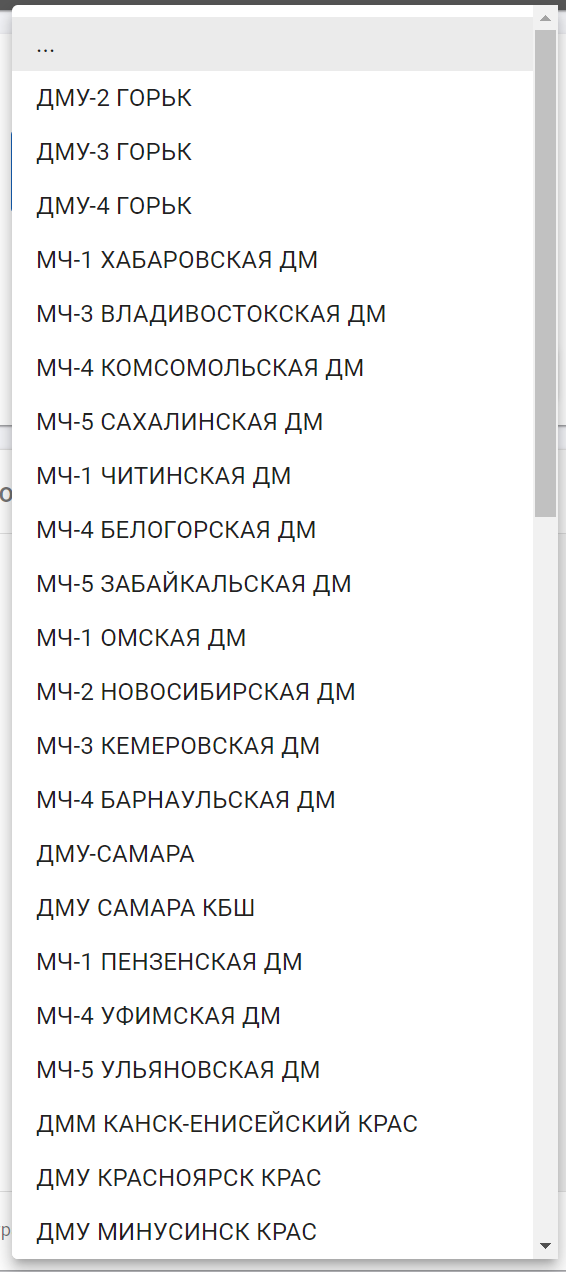


Рисунок Д. 3. – Справочник МЧ

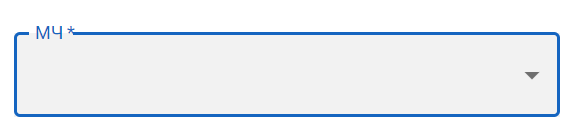


Рисунок Д. 4. – Фильтр МЧ

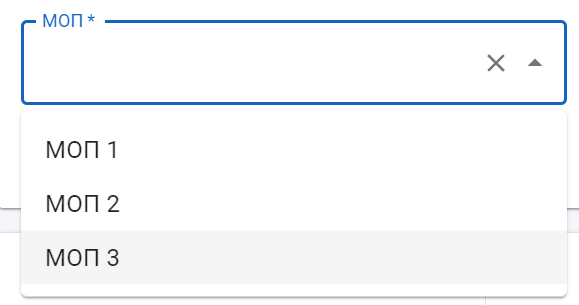


Рисунок Д. 5. – Фильтр МОП

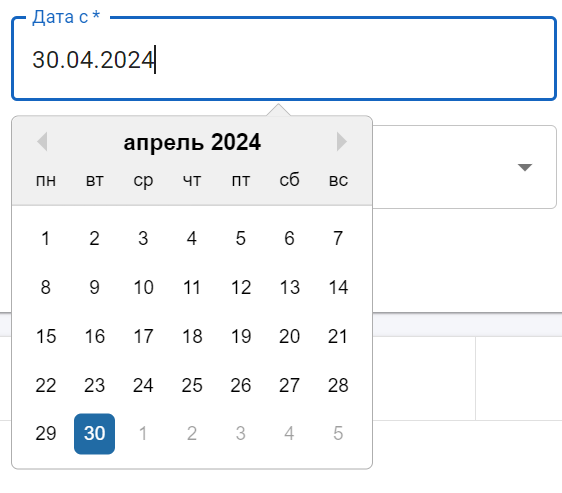


Рисунок Д. 6. – Фильтр. Дата с

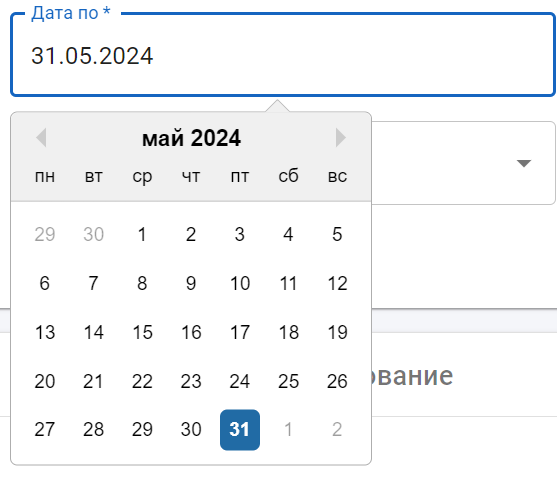


Рисунок Д. 7. – Фильтр. Дата по

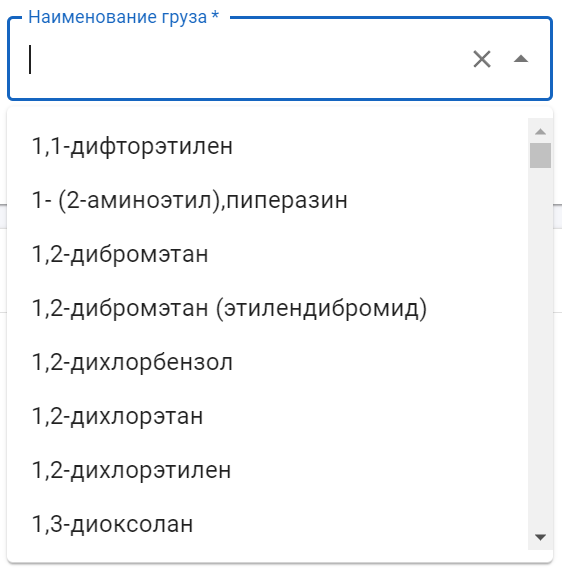


Рисунок Д. 8. – Фильтр. Наименование груза

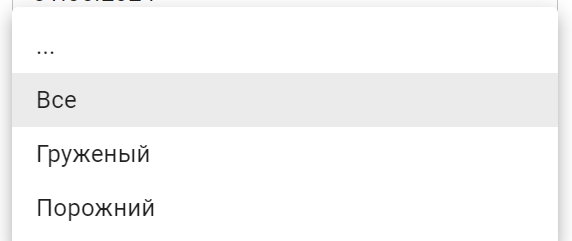


Рисунок Д. 9. – Варианты выбора состояний контейнера



Рисунок Д. 10. – Фильтр. Состояние контейнера

# **ПРИЛОЖЕНИЕ Е**Результаты проведенных экспериментов.

* Время на передачу запроса от клиента к серверу (t1);
* Время выполнения запроса на сервере (t2);
* Время на передачу ответа от сервера к клиенту и загрузку содержимого (t3).

Единица измерения – секунды.

Таблица А.5. - Результаты проведённых экспериментов

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Книга выгрузки контейнеров ГУ-44к | | | Наличие контейнеров с грузом | | | Книга вывоза контейнеров | | | Книга завоза контейнеров | | |
| № | t1 | t2 | t3 | t1 | t2 | t3 | t1 | t2 | t3 | t1 | t2 | t3 |
| 1 | 0,53 | 9,92 | 3,08 | 0,68 | 10,77 | 2,87 | 0,34 | 8,55 | 2,31 | 0,66 | 11,07 | 2,66 |
| 2 | 0,66 | 10,57 | 4,73 | 0,55 | 11,96 | 3,39 | 0,47 | 8,79 | 2,61 | 0,54 | 12,46 | 3,12 |
| 3 | 0,52 | 11,88 | 6,47 | 0,67 | 9,61 | 5,93 | 0,46 | 9,97 | 2,95 | 0,5 | 11,5 | 3,41 |
| 4 | 0,54 | 14,21 | 6,91 | 0,83 | 11,58 | 5,88 | 0,63 | 6,73 | 3,05 | 0,58 | 9,68 | 3,43 |
| 5 | 0,74 | 11,63 | 2,72 | 0,62 | 8,81 | 4,78 | 0,57 | 11,14 | 2,20 | 0,59 | 10,4 | 3,07 |
| 6 | 0,63 | 10,94 | 7,39 | 0,61 | 9,73 | 3,30 | 0,45 | 10,51 | 2,86 | 0,53 | 10,12 | 1,89 |
| 7 | 0,79 | 14,28 | 2,53 | 0,62 | 10,05 | 2,77 | 0,49 | 9,63 | 2,45 | 0,67 | 11,43 | 1,51 |
| 8 | 0,48 | 6,65 | 5,04 | 0,51 | 10,46 | 4,22 | 0,38 | 10,39 | 2,38 | 0,65 | 10,84 | 3,05 |
| 9 | 0,65 | 13,99 | 4,26 | 0,89 | 13,45 | 3,24 | 0,58 | 6,93 | 2,99 | 0,55 | 9,45 | 2,76 |
| 10 | 0,64 | 14,36 | 5,62 | 0,54 | 13,91 | 4,72 | 0,42 | 8,71 | 2,55 | 0,65 | 8,95 | 3,16 |
| 11 | 0,49 | 8,73 | 4,85 | 0,85 | 15,62 | 5,78 | 0,65 | 11,23 | 2,93 | 0,45 | 8,98 | 3,48 |
| 12 | 0,73 | 13,43 | 3,91 | 0,77 | 13,11 | 2,17 | 0,49 | 11,01 | 2,76 | 0,64 | 10,02 | 1,98 |
| 13 | 0,76 | 11,65 | 4,37 | 0,74 | 12,64 | 2,52 | 0,52 | 7,93 | 2,11 | 0,51 | 12,08 | 3,31 |
| 14 | 0,77 | 13,92 | 3,79 | 0,43 | 13,34 | 5,13 | 0,48 | 11,67 | 2,13 | 0,41 | 8,96 | 1,69 |
| 15 | 0,68 | 6,23 | 7,24 | 0,67 | 13,41 | 5,89 | 0,58 | 8,47 | 2,78 | 0,61 | 11,43 | 2,85 |
| 16 | 0,63 | 9,57 | 3,62 | 0,61 | 14,34 | 2,23 | 0,4 | 11,29 | 2,88 | 0,53 | 10,76 | 3,59 |
| 17 | 0,62 | 11,22 | 3,37 | 0,78 | 13,68 | 4,56 | 0,53 | 8,68 | 2,62 | 0,61 | 10,77 | 2,86 |
| 18 | 0,68 | 14,02 | 7,11 | 0,78 | 10,48 | 4,21 | 0,55 | 7,92 | 2,39 | 0,55 | 9,81 | 2,74 |
| 19 | 0,68 | 9,77 | 5,48 | 0,61 | 12,05 | 2,97 | 0,61 | 7,27 | 3,04 | 0,54 | 8,66 | 2,53 |
| 20 | 0,57 | 7,56 | 4,83 | 0,56 | 11,09 | 4,06 | 0,41 | 9,59 | 3,09 | 0,65 | 8,62 | 1,89 |
| 21 | 0,54 | 14,32 | 3,49 | 0,83 | 11,27 | 3,04 | 0,48 | 10,43 | 3,11 | 0,49 | 12,08 | 2,64 |
| 22 | 0,66 | 13,34 | 3,83 | 0,47 | 10,43 | 2,87 | 0,56 | 8,65 | 2,79 | 0,44 | 12,33 | 1,85 |
| 23 | 0,52 | 13,77 | 2,62 | 0,54 | 13,79 | 2,26 | 0,55 | 9,78 | 2,51 | 0,58 | 12,21 | 3,49 |
| 24 | 0,56 | 14,12 | 4,17 | 0,58 | 14,18 | 5,15 | 0,39 | 8,37 | 2,73 | 0,54 | 8,58 | 3,43 |
| 25 | 0,69 | 12,69 | 3,39 | 0,68 | 13,21 | 6,06 | 0,4 | 9,12 | 2,17 | 0,62 | 10,95 | 1,94 |
| 26 | 0,6 | 10,22 | 5,82 | 0,54 | 13,52 | 4,73 | 0,43 | 9,37 | 3,06 | 0,65 | 10,84 | 1,82 |
| 27 | 0,6 | 13,88 | 5,95 | 0,78 | 12,73 | 3,22 | 0,67 | 7,70 | 2,06 | 0,6 | 11,11 | 2,67 |
| 28 | 0,57 | 8,55 | 4,59 | 0,59 | 9,81 | 4,33 | 0,48 | 9,38 | 2,65 | 0,48 | 11,36 | 3,22 |
| 29 | 0,48 | 14,11 | 5,38 | 0,8 | 12,79 | 4,90 | 0,36 | 9,97 | 2,98 | 0,58 | 10,77 | 3,49 |
| 30 | 0,66 | 12,66 | 4,27 | 0,64 | 8,81 | 3,27 | 0,43 | 9,62 | 2,68 | 0,47 | 9,81 | 3,64 |

# **ПРИЛОЖЕНИЕ Ж**Листинг кода фронтенда

containersAvailabilityCargo.js

import { format, sub } from 'date-fns';

import { appNsi } from 'src/helpers/nsiHelper';

import { mchFilter, mopFilter } from 'src/helpers/filterHelper';

import { handleDmMchMop } from 'src/containers/Reports/changeFiltersHandlers';

/\*\*

\* Идентификатор отчёта.

\*

\* Используется для запросов бекенда (по url: http://<host>/api/reports/<id>).

\*

\* Вьюха отчёта подвязывается в роутер по url: http://<host>/#/reports/<id>

\*

\* Обязательно для заполнения.

\*/

export const id = 'containers-availability-cargo';

/\*\*

\* НаЗвание отчёта

\*

\* Обязательно для заполнения.

\*/

export const title = 'Наличие контейнеров с грузом';

/\*\*

\* Перечень колонок, ожидаемых ожидается к передаче с бекенда,

\* а также правила отображения этих колонок.

\*

\* Синтаксис: см. http://aboyko.dev.easapr/storybook-v2/?path=/docs/ui-table--column-data

\*

\* Обязательно для заполнения.

\*/

export const columnData = [

{

id: 'cargoCode',

label: 'Код груза'

},

{

id: 'cargoName',

label: 'Наименование'

},

{

id: 'containerCount',

label: 'Кол-во контейнеров'

}

];

export const filters = ({ dm, mch }, userData) => [

{

name: 'Дата с',

type: 'date',

field: 'date\_from',

maxDateField: 'date\_to',

required: true

},

{

name: 'Дата по',

type: 'date',

field: 'date\_to',

minDateField: 'date\_from',

required: true

},

{

name: 'ДМ',

type: 'nsiSelect',

field: 'dm',

api: appNsi.D\_DM,

codeField: 'id',

nameField: 'name',

disabled: !!userData.dm,

required: true

},

{

name: 'МЧ',

type: 'nsiSelect',

field: 'mch',

api: appNsi.D\_MCH,

codeField: 'id',

nameField: 'name',

filter: mchFilter(dm),

disabled: !!userData.mch,

returnArray: true,

required: true

},

{

name: 'МОП',

type: 'nsiSuggest',

field: 'mop',

api: appNsi.D\_MOP,

codeField: 'id',

nameField: 'name',

filter: mopFilter(dm, mch?.id),

disabled: !!userData.mop,

returnArray: true,

required: true

},

{

name: 'Наименование груза',

type: 'nsiSuggest',

field: 'cargo',

api: appNsi.D\_Cargo,

required: true

}

];

export const filtersInitialState = userData => ({

date\_from: format(sub(new Date(), { months: 1 }), 'yyyy-MM-dd'),

date\_to: format(new Date(), 'yyyy-MM-dd'),

dm: userData.dm,

mch: userData.mch ? { id: userData.mch, dm: userData.dm } : null,

mop: userData.mop

? { id: userData.mop, mh: userData.mch, dm: userData.dm }

: null

});

export const changeFilters = (field, value, filters, userData) => {

return handleDmMchMop({

field,

value,

filters,

dmIsDisabled: !!userData.dm,

mchIsDisabled: !!userData.mch

});

};

export const prepareLoadParams = filters => {

filters.mch = filters.mch?.id;

filters.mop = filters.mop?.id;

return filters;

};

containersDelivery.js

import { format, sub } from 'date-fns';

import { appNsi } from 'src/helpers/nsiHelper';

import { mchFilter, mopFilter } from 'src/helpers/filterHelper';

import { handleDmMchMop } from 'src/containers/Reports/changeFiltersHandlers';

/\*\*

\* Идентификатор отчёта.

\*

\* Используется для запросов бекенда (по url: http://<host>/api/reports/<id>).

\*

\* Вьюха отчёта подвязывается в роутер по url: http://<host>/#/reports/<id>

\*

\* Обязательно для заполнения.

\*/

export const id = 'containers-delivery';

/\*\*

\* НаЗвание отчёта

\*

\* Обязательно для заполнения.

\*/

export const title = 'Книга завоза контейнеров';

/\*\*

\* Перечень колонок, ожидаемых ожидается к передаче с бекенда,

\* а также правила отображения этих колонок.

\*

\* Синтаксис: см. http://aboyko.dev.easapr/storybook-v2/?path=/docs/ui-table--column-data

\*

\* Обязательно для заполнения.

\*/

export const columnData = [

{

id: 'num',

label: '№ п/п'

},

{

id: 'containerNumber',

label: 'Номер контейнера'

},

{

id: 'containerSize',

label: 'Типоразмер'

},

{

id: 'deliveryDate',

label: 'Дата и время завоза или загрузки'

},

{

id: 'keu16Num',

label: 'Номер КЭУ-16'

},

{

id: 'clientCode',

label: 'Код клиента'

},

{

id: 'client',

label: 'Наименование клиента'

},

{

id: 'weight',

label: 'Вес брутто'

},

{

id: 'autoRegNum',

label: 'Номер автомобиля'

},

{

id: 'driver',

label: 'Водитель'

},

{

id: 'note',

label: 'Примечание'

}

];

export const filters = ({ dm, mch }, userData) => [

{

name: 'Дата с',

type: 'date',

field: 'date\_from',

maxDateField: 'date\_to',

required: true

},

{

name: 'Дата по',

type: 'date',

field: 'date\_to',

minDateField: 'date\_from',

required: true

},

{

name: 'ДМ',

type: 'nsiSelect',

field: 'dm',

api: appNsi.D\_DM,

codeField: 'id',

nameField: 'name',

disabled: !!userData.dm,

required: true

},

{

name: 'МЧ',

type: 'nsiSelect',

field: 'mch',

api: appNsi.D\_MCH,

codeField: 'id',

nameField: 'name',

filter: mchFilter(dm),

disabled: !!userData.mch,

returnArray: true,

required: true

},

{

name: 'МОП',

type: 'nsiSuggest',

field: 'mop',

api: appNsi.D\_MOP,

codeField: 'id',

nameField: 'name',

filter: mopFilter(dm, mch?.id),

disabled: !!userData.mop,

returnArray: true,

required: true

},

{

name: 'Состояние контейнера',

type: 'nsiSelect',

field: 'containerState',

api: {

header: { id: 0, name: 1 },

data: [

[1, 'Все'],

[2, 'Груженый'],

[3, 'Порожний']

]

},

required: true

}

];

export const filtersInitialState = userData => ({

date\_from: format(sub(new Date(), { months: 1 }), 'yyyy-MM-dd'),

date\_to: format(new Date(), 'yyyy-MM-dd'),

dm: userData.dm,

mch: userData.mch ? { id: userData.mch, dm: userData.dm } : null,

mop: userData.mop

? { id: userData.mop, mh: userData.mch, dm: userData.dm }

: null,

containerState: 1

});

export const changeFilters = (field, value, filters, userData) => {

return handleDmMchMop({

field,

value,

filters,

dmIsDisabled: !!userData.dm,

mchIsDisabled: !!userData.mch

});

};

export const prepareLoadParams = filters => {

filters.mch = filters.mch?.id;

filters.mop = filters.mop?.id;

return filters;

};

containersRemoval.js

import { format, sub } from 'date-fns';

import { appNsi } from 'src/helpers/nsiHelper';

import { mchFilter, mopFilter } from 'src/helpers/filterHelper';

import { handleDmMchMop } from 'src/containers/Reports/changeFiltersHandlers';

/\*\*

\* Идентификатор отчёта.

\*

\* Используется для запросов бекенда (по url: http://<host>/api/reports/<id>).

\*

\* Вьюха отчёта подвязывается в роутер по url: http://<host>/#/reports/<id>

\*

\* Обязательно для заполнения.

\*/

export const id = 'containers-removal';

/\*\*

\* НаЗвание отчёта

\*

\* Обязательно для заполнения.

\*/

export const title = 'Книга вывоза контейнеров';

/\*\*

\* Перечень колонок, ожидаемых ожидается к передаче с бекенда,

\* а также правила отображения этих колонок.

\*

\* Синтаксис: см. http://aboyko.dev.easapr/storybook-v2/?path=/docs/ui-table--column-data

\*

\* Обязательно для заполнения.

\*/

export const columnData = [

{

id: 'num',

label: '№ п/п'

},

{

id: 'containerNumber',

label: 'Номер контейнера'

},

{

id: 'containerSize',

label: 'Типоразмер'

},

{

id: 'removalDate',

label: 'Дата и время вывоза или разгрузки'

},

{

id: 'keu16Num',

label: 'Номер КЭУ-16'

},

{

id: 'clientCode',

label: 'Код клиента'

},

{

id: 'client',

label: 'Наименование клиента'

},

{

id: 'weight',

label: 'Вес брутто'

},

{

id: 'autoRegNum',

label: 'Номер автомобиля'

},

{

id: 'driver',

label: 'Водитель'

},

{

id: 'note',

label: 'Примечание'

}

];

export const filters = ({ dm, mch }, userData) => [

{

name: 'Дата с',

type: 'date',

field: 'date\_from',

maxDateField: 'date\_to',

required: true

},

{

name: 'Дата по',

type: 'date',

field: 'date\_to',

minDateField: 'date\_from',

required: true

},

{

name: 'ДМ',

type: 'nsiSelect',

field: 'dm',

api: appNsi.D\_DM,

codeField: 'id',

nameField: 'name',

disabled: !!userData.dm,

required: true

},

{

name: 'МЧ',

type: 'nsiSelect',

field: 'mch',

api: appNsi.D\_MCH,

codeField: 'id',

nameField: 'name',

filter: mchFilter(dm),

disabled: !!userData.mch,

returnArray: true,

required: true

},

{

name: 'МОП',

type: 'nsiSuggest',

field: 'mop',

api: appNsi.D\_MOP,

codeField: 'id',

nameField: 'name',

filter: mopFilter(dm, mch?.id),

disabled: !!userData.mop,

returnArray: true,

required: true

},

{

name: 'Состояние контейнера',

type: 'nsiSelect',

field: 'containerState',

api: {

header: { id: 0, name: 1 },

data: [

[1, 'Все'],

[2, 'Груженый'],

[3, 'Порожний']

]

},

required: true

}

];

export const filtersInitialState = userData => ({

date\_from: format(sub(new Date(), { months: 1 }), 'yyyy-MM-dd'),

date\_to: format(new Date(), 'yyyy-MM-dd'),

dm: userData.dm,

mch: userData.mch ? { id: userData.mch, dm: userData.dm } : null,

mop: userData.mop

? { id: userData.mop, mh: userData.mch, dm: userData.dm }

: null,

bookType: 1,

containerState: 1

});

export const changeFilters = (field, value, filters, userData) => {

return handleDmMchMop({

field,

value,

filters,

dmIsDisabled: !!userData.dm,

mchIsDisabled: !!userData.mch

});

};

export const prepareLoadParams = filters => {

filters.mch = filters.mch?.id;

filters.mop = filters.mop?.id;

return filters;

};

gu44kContainerUnloading.js

import { format, sub } from 'date-fns';

import { appNsi } from 'src/helpers/nsiHelper';

import { mchFilter, mopFilter } from 'src/helpers/filterHelper';

import { handleDmMchMop } from 'src/containers/Reports/changeFiltersHandlers';

/\*\*

\* Идентификатор отчёта.

\*

\* Используется для запросов бекенда (по url: http://<host>/api/reports/<id>).

\*

\* Вьюха отчёта подвязывается в роутер по url: http://<host>/#/reports/<id>

\*

\* Обязательно для заполнения.

\*/

export const id = 'gu44k-container-unloading';

/\*\*

\* НаЗвание отчёта

\*

\* Обязательно для заполнения.

\*/

export const title = 'Книга выгрузки контейнеров ГУ-44к';

/\*\*

\* Перечень колонок, ожидаемых ожидается к передаче с бекенда,

\* а также правила отображения этих колонок.

\*

\* Синтаксис: см. http://aboyko.dev.easapr/storybook-v2/?path=/docs/ui-table--column-data

\*

\* Обязательно для заполнения.

\*/

export const columnData = [

{

id: 'num',

label: '№ п/п',

skipInHeader: true

},

{

id: 'carriageNum',

label: 'Номер вагона',

skipInHeader: true

},

{

id: 'containerOwnerCode',

label: 'Код владельца'

},

{

id: 'containerNumber',

label: 'Номер'

},

{

id: 'containerSize',

label: 'Типоразмер'

},

{

id: 'unloadingDate',

label: 'Дата и время выгрузки',

skipInHeader: true

},

{

id: 'location',

label: 'Местонахождение',

skipInHeader: true

},

{

id: 'docNum',

label: 'Номер документа',

skipInHeader: true

},

{

id: 'date',

label: 'Дата и время',

skipInHeader: true

},

{

id: 'notes',

label: 'Примечания',

skipInHeader: true

}

];

export const complicatedHeader = [

[

{

id: 'num',

label: '№ п/п',

rowspan: 2

},

{

id: 'carriageNum',

label: 'Номер вагона',

rowspan: 2

},

{

label: 'Сведения о контейнере',

colspan: 3

},

{

id: 'unloadingDate',

label: 'Дата и время выгрузки',

rowspan: 2

},

{

id: 'location',

label: 'Местонахождение',

rowspan: 2

},

{

id: 'docNum',

label: 'Номер документа',

rowspan: 2

},

{

id: 'date',

label: 'Дата и время',

rowspan: 2

},

{

id: 'notes',

label: 'Примечания',

rowspan: 2

}

]

];

export const filters = ({ dm, mch }, userData) => [

{

name: 'Дата с',

type: 'date',

field: 'date\_from',

maxDateField: 'date\_to',

required: true

},

{

name: 'Дата по',

type: 'date',

field: 'date\_to',

minDateField: 'date\_from',

required: true

},

{

name: 'ДМ',

type: 'nsiSelect',

field: 'dm',

api: appNsi.D\_DM,

codeField: 'id',

nameField: 'name',

disabled: !!userData.dm,

required: true

},

{

name: 'МЧ',

type: 'nsiSelect',

field: 'mch',

api: appNsi.D\_MCH,

codeField: 'id',

nameField: 'name',

filter: mchFilter(dm),

disabled: !!userData.mch,

returnArray: true,

required: true

},

{

name: 'МОП',

type: 'nsiSuggest',

field: 'mop',

api: appNsi.D\_MOP,

codeField: 'id',

nameField: 'name',

filter: mopFilter(dm, mch?.id),

disabled: !!userData.mop,

returnArray: true,

required: true

}

];

export const filtersInitialState = userData => ({

date\_from: format(sub(new Date(), { months: 1 }), 'yyyy-MM-dd'),

date\_to: format(new Date(), 'yyyy-MM-dd'),

dm: userData.dm,

mch: userData.mch ? { id: userData.mch, dm: userData.dm } : null,

mop: userData.mop

? { id: userData.mop, mh: userData.mch, dm: userData.dm }

: null

});

export const changeFilters = (field, value, filters, userData) => {

return handleDmMchMop({

field,

value,

filters,

dmIsDisabled: !!userData.dm,

mchIsDisabled: !!userData.mch

});

};

# **ПРИЛОЖЕНИЕ И**Листинг кода серверной части бэкенда

return [

'api/reports/gu44k-container-unloading' => [

        'controller' => 'Reports\\gu44kContainerUnloading',

        'action'     => 'list',

        'pathType'   => 1,

        'auth'       => false,

    ],

'api/reports/containers-availability-cargo' => [

        'controller' => 'Reports\\AvailabilityContainerWithCargo',

        'action'     => 'list',

        'pathType'   => 1,

        'auth'       => false,

    ],

    'api/reports/containers-delivery' => [

        'controller' => 'Reports\\ContainerDeliveryBook',

        'action'     => 'list',

        'pathType'   => 1,

        'auth'       => false,

    ],

    'api/reports/containers-removal'   => [

        'controller' => 'Reports\\ContainerRemovalBook',

        'action'     => 'list',

        'pathType'   => 1,

        'auth'       => false,

    ],

];

class gu44kContainerUnloadingModel extends ModelAbstraction

{

    const CONDITION\_SCHEMA = [

        'dm'                     => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'dm',

        ],

        'mch'                    => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'mh',

        ],

        'mop'                    => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'mop',

        ],

        'date\_from'              => [

            'option' => '>=',

            'field'  => 'time',

            'type'   => ConditionHelper::TYPE\_DATE,

            'format' => 'Y-m-d 18:01:00',

        ],

        'date\_to'                => [

            'option' => '<=',

            'field'  => 'time',

            'type'   => ConditionHelper::TYPE\_DATE,

            'format' => 'Y-m-d 18:00:00',

        ],

    ];

    public function list(?array $request): array

    {

        [$conditions, $params] = (new ConditionHelper(self::CONDITION\_SCHEMA, $request))->getConditions();

        return $this->\_execute("

select

g.id\_gu44k as num

, gl.carriageNum

, g.containerOwnerCode

, gl.containerNumber

, ct.containerSize

, g.unloadingDate

, mop.name as location

, g.docNum

, g.date

, gl.notes

from GU-44k g

left join mop on mop.id\_mop = g.mop\_id

left join Plots p on p.id\_plots = mop.id\_plots

left join DM dm on dm.id\_dm = p.id\_dm

left join MCH mch on mch.id\_mch = dp.id\_mch

left join GU-44k\_list gl on gl.id\_gu44k = g.id\_gu44k

left join Cargo\_type ct on ct.id\_cargo = gl.id\_class

where {$conditions}

        ", $params);

    }

}

class gu44kContainerUnloadingController extends Controller

{

    public function listAction(): array

    {

        try {

            $this->return([

                'success' => true,

                'error' => null,

                'data' => (new gu44kContainerUnloadingModel())->list($this->request['params'])

            ]);

        } catch (\Throwable $e) {

            $this->return([

                'success' => false,

                'error' => $e->getMessage()

            ]);

        }

    }

}

class AvailabilityContainerWithCargoModel extends ModelAbstraction

{

    const CONDITION\_SCHEMA = [

        'dm'                     => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'dm',

        ],

        'mch'                    => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'mch',

        ],

        'mop'                    => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'mop',

        ],

        'date\_from'              => [

            'option' => '>=',

            'field'  => 'date',

            'type'   => ConditionHelper::TYPE\_DATE,

            'format' => 'Y-m-d 00:00:00',

        ],

        'date\_to'                => [

            'option' => '<=',

            'field'  => 'date',

            'type'   => ConditionHelper::TYPE\_DATE,

            'format' => 'Y-m-d 23:59:59',

        ],

        'cargo'      => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'cargo',

        ],

    ];

    public function list(?array $request): array

    {

        [$conditions, $params] = (new ConditionHelper(self::CONDITION\_SCHEMA, $request))->getConditions();

        return $this->\_execute("

select

ck.cargoCode,

e.cargoName,

count (distinct cd.containerNumber) as containerCount

from Cargo\_day cd

left join mop on mop.id\_mop = scd.mop\_id

left join Plots p on p.id\_plots = mop.id\_plots

left join DM dm on dm.id\_dm = p.id\_dm

left join MCH mch on mch.id\_mch = dp.id\_mch

left join CK ck on ck.id\_ck = cd.id\_ck

left join ENTSNG e on e.id\_etsng = ck.id\_etsng

group by

ck. ck.cargoCode

, e.cargoName

where {$conditions}

“,$params);

    }

}

class AvailabilityContainerWithCargoController extends Controller

{

    public function listAction(): array

    {

        try {

            $this->return([

                'success' => true,

                'error' => null,

                'data' => (new AvailabilityContainerWithCargoModel())->list($this->request['params'])

            ]);

        } catch (\Throwable $e) {

            $this->return([

                'success' => false,

                'error' => $e->getMessage()

            ]);

        }

    }

}

class ContainerDeliveryBookModel extends ModelAbstraction

{

    const CONDITION\_SCHEMA = [

        'dm'                     => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'dm',

        ],

        'mch'                    => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'mch',

        ],

        'mop'                    => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'mop',

        ],

        'date\_from'              => [

            'option' => '>=',

            'field'  => 'date',

            'type'   => ConditionHelper::TYPE\_DATE,

            'format' => 'Y-m-d 18:01:00',

        ],

        'date\_to'                => [

            'option' => '<=',

            'field'  => 'date',

            'type'   => ConditionHelper::TYPE\_DATE,

            'format' => 'Y-m-d 18:00:00',

        ],

       'containerState'         => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'type',

        ],

    ];

    public function list(?array $request): array

    {

        [$conditions, $params] = (new ConditionHelper(self::CONDITION\_SCHEMA, $request))->getConditions();

        return $this->\_execute("

select

k.id\_keu as num

, k.containerNumber

, k.containerSize

, k.deliveryDate

, k.keu16Num

, c.clientCode

, k.client

, k.weight

, k.autoRegNum

, k.driver

, k.note

from keu16 k

left join mop on mop.id = k.mop

left join Plots p on p.id\_plots = mop.id\_plots

left join DM dm on dm.id\_dm = p.id\_dm

left join MCH mch on mch.id\_mch = dp.id\_mch

left join client c on c.id\_client = k.id\_client

where k.reason\_type = 2 -- только на приём

and where {$conditions}

       "

        , $params);

    }

}

class ContainerDeliveryBookController extends Controller

{

    public function listAction(): array

    {

        try {

            $this->return([

                'success' => true,

                'error' => null,

                'data' => (new ContainerDeliveryBookModel())->list($this->request['params'])

            ]);

        } catch (\Throwable $e) {

            $this->return([

                'success' => false,

                'error' => $e->getMessage()

            ]);

        }

    }

}

class ContainerRemovalBookModel extends ModelAbstraction

{

    const CONDITION\_SCHEMA = [

        'dm'                     => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'dm',

        ],

        'mch'                    => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'mch',

        ],

        'mop'                    => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'mop',

        ],

        'date\_from'              => [

            'option' => '>=',

            'field'  => 'date',

            'type'   => ConditionHelper::TYPE\_DATE,

            'format' => 'Y-m-d 18:01:00',

        ],

        'date\_to'                => [

            'option' => '<=',

            'field'  => 'date',

            'type'   => ConditionHelper::TYPE\_DATE,

            'format' => 'Y-m-d 18:00:00',

        ],

        'containerState'         => [

            'type'   => ConditionHelper::TYPE\_ARRAY,

            'option' => 'IN',

            'field'  => 'type',

        ],

    ];

    public function list(?array $request): array

    {

        [$conditions, $params] = (new ConditionHelper(self::CONDITION\_SCHEMA, $request))->getConditions();+

        return $this->\_execute("

select

k.id\_keu as num

, k.containerNumber

, k.containerSize

, k.deliveryDate

, k.keu16Num

, c.clientCode

, k.client

, k.weight

, k.autoRegNum

, k.driver

, k.note

from keu16 k

left join mop on mop.id = k.mop

left join Plots p on p.id\_plots = mop.id\_plots

left join DM dm on dm.id\_dm = p.id\_dm

left join MCH mch on mch.id\_mch = dp.id\_mch

left join client c on c.id\_client = k.id\_client

where k.reason\_type = 3 -- только на выдачу

and where {$conditions}"

        , $params);

    }

}

class ContainerRemovalBookController extends Controller

{

    public function listAction(): array

    {

        try {

            $this->return([

                'success' => true,

                'error' => null,

                'data' => (new ContainerRemovalBookModel())->list($this->request['params'])

            ]);

        } catch (\Throwable $e) {

            $this->return([

                'success' => false,

                'error' => $e->getMessage()

            ]);

        }

    }

}