

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Национальный исследовательский ядерный университет «МИФИ»  
**Технологический институт –**  
филиал федерального государственного автономного образовательного учреждения высшего  
образования «Национальный исследовательский ядерный университет «МИФИ»  
**(ТИ НИЯУ МИФИ)**

**Кафедра информационных технологий и прикладной математики**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ  
РАБОТА БАКАЛАВРА**

направление подготовки 09.03.01 «Информатика и вычислительная техника»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**Тема:** «Анализ наличия и движения МТР на складах комбината»

**Руководитель**

Начальник отдела в  
подразделении  
(должность)

(подпись)

А.А.Дорофеев

(И.О. Фамилия)

**Студент**

ИВТ-40Д  
(группа)

(подпись)

П.А.Лабазова

(И.О. Фамилия)

**ВКР допущена к защите в ГЭК**

**Заведующий кафедрой**

Н.В. Чупракова

«19» июня 2024 г.

ВКР защищена «21» июня 2024 г.

Протокол ГЭК № \_\_\_\_\_ на оценку « \_\_\_\_\_ »

Секретарь ГЭК \_\_\_\_\_ /Ю.А. Порохина /

г. Лесной – 2024 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
ГЛАВА 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ .....	8
1.1 Анализ предметной области .....	8
1.2 Описание бизнес-процесса .....	13
1.3 Средства разработки программного обеспечения .....	20
ГЛАВА 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ .....	25
2.1 Проектирование базы данных .....	25
2.2 Разработка серверной части приложения .....	32
2.3 Проектирование клиентской части .....	39
ГЛАВА 3 ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ .....	43
3.1 Руководство пользователя. Тестирование приложения .....	43
3.2 Планируемый эффект от внедрения .....	49
ЗАКЛЮЧЕНИЕ .....	52
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	53
Приложение А .....	56
Приложение Б .....	57
Приложение В .....	59
Приложение Г .....	61
Приложение Д .....	63
Приложение Е .....	67
Приложение Ж .....	76

## **ВВЕДЕНИЕ**

В современном мире информационные технологии представляют собой одну из ключевых отраслей, определяющих развитие экономики и общества в целом. Особенно важное значение приобретает использование программного обеспечения в IT-сфере, где зависимость от импортного софта нередко становится проблемой из-за различных факторов. Один из основных инструментов регулирования данной сферы - санкции, которые имеют непосредственное влияние на выбор и использование импортного программного обеспечения. Вопросы импортозамещения и использования отечественного программного обеспечения в IT-сфере приобретают все более значимое значение.

Еще до введения санкций наше государство осознавало важность независимого от импорта программного обеспечения и предпринимало шаги по стимулированию российских разработчиков к созданию и внедрению таких программных продуктов.

Проект «Цифровая экономика» 2018 года является ярким примером стремления к развитию отечественной IT-индустрии и снижению зависимости от иностранных технологий.

Данный проект представляет масштабную программу, которая была запущена правительством Российской Федерации с целью содействия развитию цифровой сферы в стране. Проект был разработан с учетом стратегической значимости цифровизации для дальнейшего экономического роста и устойчивого развития России.

Основной задачей Проекта "Цифровая экономика" было создание благоприятного инвестиционного климата для развития цифровых технологий, стимулирование инноваций, улучшение качества услуг и повышение эффективности государственного управления. Программа включала в себя целый ряд мероприятий, направленных на развитие Интернет-инфраструктуры, цифровых услуг, кибербезопасности, интеллектуальной

собственности и других аспектов цифровой сферы и создание благоприятных условий для цифровой трансформации отраслей, внедрения современных технологий и повышения уровня доступа к цифровым услугам для всего населения.

Одним из ключевых направлений развития цифровой экономики является продвижение инновационных веб-приложений, которые позволяют улучшить доступность и удобство использования информационных технологий для широкого круга пользователей. Переход к веб-приложениям открывает новые возможности для бизнеса, обеспечивая гибкость, масштабируемость и возможность работы через интернет.

Переход к веб-приложениям, в рамках проекта «Цифровая экономика» 2018 года, стимулирует развитие отечественной IT-индустрии и создание конкурентоспособных продуктов.

Перемены в офисном софте, системах управления базами данных и серверах становятся неотъемлемой частью цифровой трансформации многих организаций. Обновление программного обеспечения, переход на более современные версии программ, миграция данных на новые платформы - все это способы повышения эффективности работы и обеспечения безопасности информационных технологий.

Государственная корпорация «Росатом», многопрофильный холдинг, объединяющий активы в энергетике, машиностроении и строительстве, реализует масштабную программу единой цифровой стратегии по импортозамещению, консолидирующая множество крупных проектов

Векторы ИТ-импортозамещения в государственной корпорации «Росатом» включают в себя ряд стратегий и действий, направленных на снижение зависимости от зарубежных поставщиков и увеличение доли отечественных технологий и продукции в информационных технологиях. «Росатом» активно работает над развитием собственных технологий, цифровизацией процессов и укреплением кибербезопасности.

Одним из ключевых векторов ИТ-импортозамещения в «Росатоме» является развитие отечественных цифровых решений и программного обеспечения для обеспечения безопасности объектов атомной энергетики. Это позволяет снизить риски, связанные с использованием иностранных технологий, а также повысить качество и надежность систем безопасности.

Также, корпорация уделяет внимание развитию отечественных платформ и сервисов для облачных технологий и хранения данных. Это сокращает зависимость от зарубежных облачных провайдеров и обеспечивает контроль над хранением и обработкой информации.

Таким образом, «Росатом» активно работает над укреплением отечественного сегмента информационных технологий, снижая зависимость от импорта и способствуя развитию отечественной ИТ-индустрии.

ФГУП «Комбинат «Электрохимприбор» - один из ведущих подведомственных предприятий Государственной корпорации «Росатом», отвечающих за перевод информационных систем «Росатома» на импортонезависимую платформу.

Отдел «Управление информационных технологий и связи» (УИТиС) в ФГУП «Комбинат «Электрохимприбор» занимается разработкой и поддержкой программного обеспечения, необходимого для автоматизации бизнес-процессов предприятия. В свете реализации проектов госкорпорации Росатом по минимизации санкционных рисков и повышению информационной безопасности, система управления ФГУП «Комбинат «Электрохимприбор» должна быть переведена на импортонезависимые компоненты. Это необходимо для гарантированной полной защищенности жизненно-важной аппаратуры в любых ситуациях, в том числе от незадекларированных возможностей, которые могут присутствовать в иностранных вычислительных комплексах и процессорах и таким образом нарушать конфиденциальность информации.

С учетом вышеуказанных факторов было принято решение разработать программный модуль «Анализ наличия и движения МТР на складах

комбината» с помощью технологии веб-программирования для автоматизации учета и анализа материально-технических ресурсов на складах предприятия.

Объектом исследования выпускной квалификационной работы является деятельность группы материально-технического снабжения (МТС) отдела ООП УИТиС (Управление информационных технологий и связи) ФГУП «Комбинат «Электрохимприбор».

Целью работы является разработка веб-приложения «Анализ наличия и движения МТР на складах комбината», позволяющего пользователям группы осуществлять следующие функции:

- Поиск наличия материалов на складах;
- Просмотр движения материалов на складах;
- Просмотр материалов без движения на заданное количество дней;
- Просмотр сертификата на партию;
- Просмотр финансового анализа по нормируемым балансовым счетам (сравнение фактического запаса МТР на складах с нормативным лимитом).

Техническое задание на выпускную квалификационную работу определяет стек технологий для реализации программного продукта. Приложение должно разрабатываться на языке программирования Python с использованием фреймворков Django, Django Rest Framework для реализации API приложения и django-cors-headers, для возможности интеграции данного приложения с другими веб-сервисами. Веб-страницы должны быть разработаны с помощью языка разметки HTML, языка таблиц стилей CSS, языка программирования JavaScript, библиотек Bootstrap5.

Для реализации поставленной цели необходимо решить следующие задачи:

1. Изучение анализа предметной области. Определение терминов;
2. Изучение существующей на предприятии информационной системы для дальнейшей интеграции;

3. Составление моделей бизнес-процесса;
4. Проектирование базы данных;
5. Разработка серверной части приложения;
6. Разработка клиентской части приложения;
7. Реализация веб-сервиса;
8. Тестирование программного продукта;
9. Составление инструкции пользователя;
10. Проведение анализа эффекта от внедрения разработки.

# ГЛАВА 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ

## 1.1 Анализ предметной области

Задача анализа наличия и движения материально-технических ресурсов на складах комбината является важной и ответственной функцией в обеспечении бесперебойной работы предприятия, ведь упорядочение и систематизация ресурсов позволяют определить их наличие и использование в определенные периоды времени, а также решить следующие задачи:

- Учет и контроль использования ресурсов;
- Оптимизация закупочных процедур: анализ текущего состояния запасов, прогнозирование потребностей, определение оптимальных объемов закупок для обеспечения бесперебойного производства;
- Организация складского учета: контроль за поступлением, отгрузкой, перемещением и остатками материалов на складах, обеспечение точности и актуальности данных;
- Анализ эффективности использования ресурсов: оценка затрат на покупку, хранение и транспортировку МТР, выявление и устранение потерь и излишек.

В текущем дипломе используются следующие ключевые понятия и термины:

- Материально-технический ресурс (МТР) – ресурс, используемый предприятием в производстве продукции;
- Склад — специально выделенная территория, предназначенная для сохранения и хранения товаров и материалов. Основная функция склада заключается в собирании и хранении запасов, обеспечивая непрерывное и систематичное поступление товаров для удовлетворения потребностей клиентов;
- МОЛ (материально ответственное лицо) — это сотрудник, который несет ответственность за надзор за сохранностью материальных и финансовых ресурсов на предприятии во время их хранения или



- транспортировки. В случае утраты или повреждения таких ресурсов, МОЛ обязан возместить причиненный ущерб;
- ТМЦ — товарно-материальные ценности. Предметы и средства труда, которые нужны для функционирования фирмы и ее деятельности. В бухгалтерском учете это сырье и материалы, товары для перепродажи, готовая продукция и незавершенное производство;
  - ЦС — цеховой склад. Это специальный вид склада, в котором хранят и управляют материалами, которые требуются для производства. Такой склад привязывается к рабочему центру, с помощью которого материалы, которые требуются под операции, опускаются из запаса на нем (на связанную операцию);
  - ЛЗК — лимитно-заборная карта — первичный документ, предназначенный для отпуска материалов, систематически потребляемых для изготовления продукции, а также для контроля за соблюдением лимитов;
  - Счет-фактура — документ, используемый для описания и детализации продаваемых товаров или оказываемых услуг. Это обязательный бухгалтерский документ, который подтверждает, что заказ покупателем получен в полном объеме и что к поставщику не имеется никаких претензий;
  - Цех — основное производственное подразделение промышленного предприятия;
  - МТС — материально-техническое снабжение;
  - Сдаточная накладная — первичный документ, отражающий движение полуфабрикатов, деталей и узлов в производстве. На их основании ведется учет выполнения товарного плана цехами и участками. Учет заготовок, полуфабрикатов и деталей в промежуточных межцеховых и в цеховых складах ведется путем специальных приходно-расходных карт,

- в которых отмечается поступление и выдача полуфабрикатов, деталей и узлов;
- Сертификат МТР — это документ, используемый для учета движения материально-технических ресурсов (МТР) на предприятиях и организациях. Он содержит информацию о товарах, их количестве, стоимости, дате и месте поступления или отпуска, а также о лице, ответственном за выполнение операции с МТР. Сертификаты МТР позволяют контролировать оборот МТР, управлять запасами и обеспечивать эффективное планирование поставок и использования ресурсов;
  - Партия — однородный материал, который поступает по одному документу. Для каждой партии ответственное лицо оформляет партионную карту в двух экземплярах: один для склада, другой для бухгалтерии. Партия используется для группировки и контроля за однородными материалами или товарами, облегчая процессы учета, хранения и осуществления операций с ними. Каждая партия может иметь уникальные характеристики, такие как номер партии, дата поступления, количество, качество и другая информация, необходимая для идентификации и контроля за материалами. Партии помогают эффективно управлять запасами и обеспечивают прозрачность в процессах хранения и использования материалов;
  - Приход - принятие на производственное предприятие или в процесс производства сырья, материалов или комплектующих. Когда материалы или комплектующие поступают на склад или непосредственно на производственную линию, это считается операцией "приход". Приход в производство играет важную роль в управлении запасами и планировании производственных процессов.
  - Внутреннее перемещение — процесс передачи материалов между складами различных подразделений или цехов организации, а также на

объекты строительства. Это важная операция, которая обеспечивает необходимое снабжение материалами и оборудованием для всех производственных процессов и строительных работ;

- Отпуск материалов в производстве (списание, расход)— это процесс выдачи материалов со склада предприятия (цеха) для использования в производстве товаров (непосредственно для изготовления продукции, выполнении работ, предоставлении услуг), а также для управленческих целей организации.

Наличие МТР – наличие определенных материалов на предприятии на определенном складе. Наличие и доступность МТР для использования в производственном процессе:

- Наличие МТС;
- Наличие в подразделении или складе: в данном варианте учитываются все приходы и расходы;
- Наличие в бухгалтерском учете.

На комбинате учет МТР ведется по партиям. МТР-партия идентифицируется уникальным составным ключом, состоящим из номера подстроки, номера приходного ордера, кода МТР (семизначного значения).

В случае, когда материалы требуют наличия сертификатов или других соответствующих документов, сертификаты на материалы должны быть выданы отделом снабжения и связаны с соответствующими партиями.

Движение МТР – информация о перемещении и использовании материалов на предприятии, т.е. отражение всех операций, связанных с поступлением, хранением, перемещением, выдачей, использованием и списанием МТР на предприятии.

Реестр МТР без движения – информация о МТР, которые находятся на предприятии, и не подвергаются перемещению или использованию в текущий период времени. Проще говоря, партии материалов, по которым не происходило движение за определенный период времени. Такие партии

являются проблемными местами, как и МТР с подходящим или истекшим сроком хранения.

В корпоративной сети комбината обрабатывается (просматривается, рассчитывается, анализируется) информация, данные которой хранятся на SQL-сервере корпоративной сети, который находится в отделе управления информационных технологий и связи. Выгрузка и загрузка информации на сервер корпоративной сети производится с рабочей станции, подключенной к ней.

В качестве места хранения данных была дана система витрин данных «КУМИР» в PostgreSQL, содержащая в себе данные АСУП «Управление материально-техническим снабжением». Данная витрина обновляется ежедневно по расписанию [1].

Витрины базы данных (Data Warehouse) – это специальные структуры, представляющие собой срез данных или подмножество данных, отображаемых для конкретных задач или анализа. Они предназначены для ускорения выполнения аналитических запросов и формирования отчетов, минуя нагрузку на операционные системы [2].

Основные характеристики витрины базы данных:

- Централизованное хранение данных: данные из различных источников объединяются в единую структуру для удобного анализа.;
- Историчность данных: в витрине сохраняется информация о состоянии данных на различные моменты времени, что позволяет проводить анализ изменений и трендов;
- Оптимизация для аналитических запросов: структура витрины оптимизирована для выполнения сложных аналитических запросов, что обеспечивает быструю обработку данных;
- Денормализация данных: для ускорения доступа к данным витрины часто содержат денормализованные данные, что уменьшает количество необходимых соединений таблиц.

Преимущества витрины базы данных:

- Улучшенная производительность: специализированная структура хранения данных и оптимизированные запросы обеспечивают быстрый доступ к информации;
- Интеграция данных: возможность объединения различных источников данных для создания единой точки доступа к информации;
- Исторические данные: возможность хранить и анализировать данные за длительный период времени.

Недостатки витрины базы данных:

- Сложность внедрения: построение и настройка витрин требует времени и усилий, особенно при интеграции данных различных источников;
- Затраты на обслуживание: поддержка витрины баз данных требует специализированных знаний и ресурсов;
- Ограничения в реальном времени: витрины баз данных обычно работают на основе данных, актуализированных с определенной задержкой.

Витрины баз данных играют важную роль в аналитике организаций, обеспечивая доступ к ценной информации для принятия стратегических решений и оптимизации бизнес-процессов.

## **1.2 Описание бизнес-процесса**

Для успешного решения бизнес-задачи требуется разработать (спроектировать) информационную систему, которая будет ориентирована на определенное подразделение предприятия, отдельную функциональную подсистему или группу задач на основе использования программного продукта и процессного подхода, поэтому необходимо [3]:

- Создать модели бизнес-процессов или функционирования ресурсов с учетом архитектуры предприятия и автоматизации в ФГУП «Комбинат «Электрохимприбор»»;

- Обосновать выбор программного продукта и необходимых модификаций, анализ соответствия решения по задаче и моделям бизнес-процессов;
- Описать информационное обеспечение бизнес-задачи, включая классификацию и кодирование данных, формы документов и документооборот, концептуальную и логическую модели базы данных;
- Оценить влияние предложенного решения на работу организации и ключевые показатели бизнес-процессов.

На предприятии при моделировании используются следующие типы диаграмм:

- Диаграмма цепочки добавленной стоимости (VAD – Value Added Chain Diagram) используется для описания взаимодействия процессов верхнего уровня;
- Диаграмма плавательных дорожек (SLD – Swim Lane Diagram) используется для описания процессов среднего и нижнего уровня;
- Диаграмма расширенной цепочки процессов, управляемой событиями, (Eepc – Extended Event Driven Process Chain) используется для описания процессов верхнего, среднего и нижнего уровня;
- Диаграмма организационной структуры (OC – Organizational Chart) используется для описания организационных моделей;
- Диаграмма дерева функций (FT – Function Tree) используется для описания структуры процессов и функций;
- Диаграмма типов информационных систем (ASTD – Application System Type Diagram) используется для описания структуры информационных систем.

Для осуществления автоматизации данного процесса необходимо достичь ясного понимания структуры его разработки как заказчиком, так и исполнителем. Модель бизнес-процесса способна представить полную схему процесса и определить конкретные компоненты, которые подлежат автоматизации с использованием программного модуля [4].

Диаграмма внешнего уровня бизнес-процесса – это диаграмма, которая демонстрирует весь процесс в целом, но не углубляется в детали его выполнения. Она показывает более общую картину процесса и выделяет ключевые этапы, которые обычно охватываются в процессе.

Диаграмма внешнего уровня обычно включает в себя основные этапы процесса, которые отображают ключевые шаги, необходимые для выполнения процесса. Примерами могут быть этапы, связанные со сбором данных, анализом информации, разработкой планов и стратегий, выпуском продукта или услуги.

Диаграмма внешнего уровня также может отображать, как различные участники взаимодействуют между собой в рамках процесса, и показывает, какие ресурсы требуются для успешного выполнения процесса. Во многих случаях диаграмма внешнего уровня бизнес-процесса может быть использована для быстрой оценки процесса и его ключевых характеристик.

Таким образом, диаграмма внешнего уровня бизнес-процесса показывает общую картину процесса и его этапы, участников, ресурсы и взаимодействия между ними. Это полезный инструмент для понимания процесса и его ключевых характеристик, но она не раскрывает подробности и мелкие детали, которые могут быть выявлены на более детальных уровнях диаграммы [5].

На рисунке 1.1 представлена диаграмма внешнего бизнес-процесса, выстроенного в нотации swim-lane.

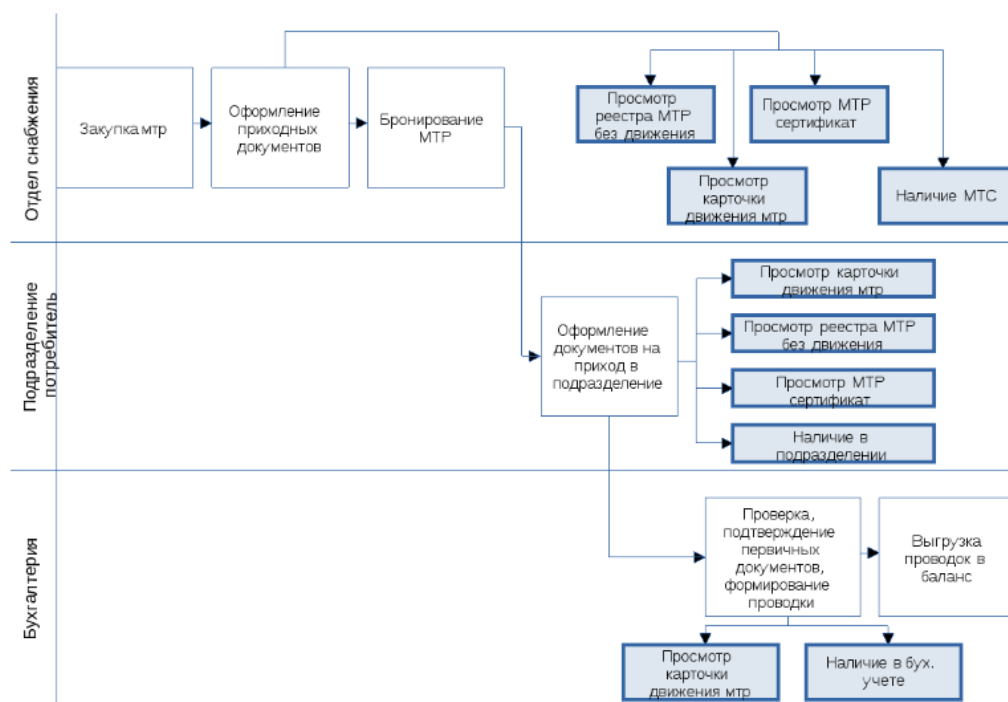


Рисунок 1.1 – Внешний уровень бизнес-процесса

Данный процесс начинается с получения отделом снабжения 054 заявок на закупку материалов от подразделений-потребителей. Отдел снабжения осуществляет планирование закупок, учитывая потребности различных подразделений, и формирует заказы поставщикам. После утверждения заказа отдел снабжения передает информацию о заказе на поставку материалов поставщику, который в свою очередь доставляет материалы на склад предприятия.

При поступлении материалов на предприятие проводится процедура приемки. В ходе нее осуществляется проверка соответствия фактического количества, качества и ассортимента материалов информации, указанной в сопроводительных документах от поставщика. По сути, при приемке материалы проходят инспекцию для убеждения в согласии их характеристик с указанными в документации требованиями и условиями контракта.

Когда грузы, доставленные уполномоченным лицом, попадают на склад предприятия, их передают под расписку заведующему складом. Материально-ответственное лицо (МОЛ) проверяет соответствие материалов информации,



заявленной в сопроводительных документах. В случае отсутствия несоответствий МОЛ составляет приходный ордер, который заведующий складом и экспедитор подписывают.

Информация о приемке передается в бухгалтерию для учета поступления. Весь процесс движения материалов строго фиксируется и учитывается в учетных системах бухгалтерии.

После приемки и учета материалы хранятся на центральном складе или в специальных складских помещениях. Внутри складов материалы организованы по секциям, затем сортированы по типу и размеру для удобства их приемки, выдачи и проверки на наличие. Это размещение на стеллажах и полках позволяет обеспечить эффективную логистику и оперативное управление запасами.

Для соблюдения утвержденных лимитов отпуска материалов используется ЛЗК. Этот документ также используется для контроля за соответствием установленных лимитов отпуска материалов и служит основанием для списания материальных ценностей со склада.

Затем, подразделение-потребитель подает запрос на выдачу необходимых материалов. Отдел снабжения осуществляет отгрузку материалов со склада, учитывая запросы подразделений-потребителей. На этом этапе происходит оформление документов на приход в подразделение.

Бухгалтерия осуществляет учет всех операций по движению материалов, включая поступление, отгрузку и списание материалов. Контроль за остатками материалов на складе и у подразделений-потребителей также осуществляется бухгалтерией. Все операции по движению материалов завершаются фиксацией и составлением отчетности для анализа и контроля процесса.

Диаграмма внутреннего уровня бизнес-процесса – это более детальная диаграмма, чем диаграмма внешнего уровня, которая позволяет представить процесс в более подробной форме. В отличие от диаграммы внешнего уровня, диаграмма внутреннего уровня описывает каждый этап процесса и его потоки, включая каждый шаг и взаимодействие между участниками.

Диаграмма внутреннего уровня включает в себя подробные описания шагов и действий, выполняемых на каждом этапе процесса, и может включать рисунки, текстили или диаграммы, чтобы проиллюстрировать детали и элементы процесса. Эта диаграмма может быть использована для детального анализа процесса, выявления недостатков и определения возможных решений.

На диаграмме внутреннего уровня бизнес-процесса могут быть отображены такие элементы, как потоки документов, списки задач, инструкции по выполнению действий, обозначение участников процесса и их роли в прохождении процесса, ограничения, сроки и прочее [6].

Таким образом, диаграмма внутреннего уровня бизнес-процесса позволяет представить детальный и подробный обзор процесса, его элементов и взаимодействия между ними. Она может использоваться для анализа и оптимизации процесса в рамках компании, что может улучшить производительность и повысить удовлетворенность клиентов.

Бизнес-процесс по анализу наличия и движения МТР планируется выстроить в нотации swim-lane и нотации, основанной на нотации IDEF0. В приложении А, представлена диаграмма внутреннего уровня бизнес-процесса в нотации swim-lane.

Есть несколько способов для поступления материалов на предприятие.

Во-первых, от поставщиков. При поступлении материалов на предприятие производится их приемка. МОЛ проверяет соответствие материалов их данным, указанным в сопроводительных документах, и при отсутствии каких-либо расхождений формируется приходный ордер. Это документ, который оформляется при поступлении товаров, медикаментов, материалов или других ценностей на склад организации. В приходном ордере указываются данные о поставщике, дате получения груза, количестве и качестве поступивших товаров, а также другая информация, необходимая для учета и контроля приема материалов. Подписание приходного ордера является подтверждением факта поступления товара на склад и осуществляется ответственным лицом, принимающим груз.

Во-вторых, инвентаризация. С целью выявления фактического наличия МТР, числящихся в бухгалтерском учете, проводится регулярная инвентаризация остатков материала на складе, т. е. проверка наличия имущества организации и состояние ее финансовых обязательств на определенную дату путем сличения фактических данных с данными бухгалтерского учета. Составляются документы «Инвентаризация».

В-третьих, собственное производство. Поступление материалов, изготовленных или переработанных цехами и участками собственного производства и сданных на склад материальных ценностей, должно оформляться путем выписки «Приход из производства». Также в данном случае возможен перевод МТР в ТМЦ с обязательной постановкой на бухгалтерский учет.

В-четвертых, документами «Возврат из производства» оформляются также операции по сдаче цехами (участками) на склад или в кладовую цеха (участка) из производства остатков неизрасходованных материалов (если они ранее были получены по требованию), а также сдача отходов от производства продукции, отходов от брака.

Движение материальных ценностей в другие склады, в кладовые цехов, участков, следует рассматривать не как отпуск на производство, а как внутреннее перемещение, которое сопровождается документом «Внутреннее перемещение».

Основным видом расхода материалов является отпуск на производство продукции, что сопровождается документом «Отпуск в производство».

Процесс фиксации и учета списания материалов или оборудования, которые уже не пригодны для использования или выбыли из активного использования на предприятии, сопровождается документом «Списание МТР».

«Накладная на отпуск на сторону» - документ, который используется при отпуске или передаче товаров или материалов с предприятия одной стороны на другое предприятие или организацию.

Разница между реальным физическим количеством товаров или материалов на складе и данными, указанными в учетной системе предприятия в результате проведения инвентаризации, указывается в документе «Недостача по инвентаризации».

В моделировании бизнес-процесса была использована диаграмма swimlane по следующим причинам:

- Swim-lane диаграмма позволяет визуально отобразить бизнес-процесс с большим количеством участников;
- Swim-lane позволяет визуализировать сложные бизнес-процессы и улучшает понимание функций процесса и взаимодействие между участниками;
- Данная диаграмма используется на предприятии, поэтому может однозначно интерпретироваться каждым работником IT-отдела;
- Позволяет оптимизировать процесс средствами анализа действий участников процесса для удаления ненужных действий.

### **1.3 Средства разработки программного обеспечения**

В соответствии с техническим заданием обязательное требование для разработки веб-приложения – использование стека веб-технологий (HTML, CSS, JavaScript и Python) и прочее импортонезависимое программное обеспечение:

Серверная часть:

- Python – высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью.
- Django – это фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC.
- MVC («Model-View-Controller», «Модель-Вид-Контроллер») – схема разделения данных приложения и управляющей логики на три

отдельных компонента, модификация каждого компонента может осуществляться независимо. В Django, MVC отображается в паттерне Model-View-Template (MVT), где модель (Model) отвечает за управление данными, представление (View) отображает содержимое действий пользователей, а шаблоны (Template) используются для визуализации вывода на сайте.

Таким образом, в Django Model представляет базу данных, а View и Template представляют пользовательский интерфейс. Контроллер, третья часть оригинального MVC, не упоминается в Django, так как он логикой обработки запросов middleware.

MVT-Pattern в Django помогает разработчикам создавать приложения, которые могут быть разделены на отдельные компоненты и, таким образом, делают приложения более модульными, легкими для поддержки и улучшения.

- Django Rest Framework (DRF) – это библиотека, которая работает со стандартными моделями Django для создания гибкого и мощного API.
- API (Application Programming Interface) – описание способов взаимодействия одной компьютерной программы с другими.
- PostgreSQL – объектно-реляционная система управления базами данных (СУБД).
- HTTP (Hypertext Transfer Protocol) – протокол передачи гипертекста, используемый в интернете для передачи данных между клиентами и серверами. Он определяет формат и правила передачи данных между персональными компьютерами, которые взаимодействуют в сети. HTTP работает на основе запросов и ответов, где клиент отправляет запрос на сервер, а сервер возвращает ответ с запрошенной информацией. Для отправки запроса используется метод, такой как GET, POST, PUT и 19 другие. Каждый метод определяет тип действия, которое требуется выполнить на сервере.

Клиентская часть:

- HTML (HyperText Markup Language) – стандартизированный язык гипертекстовой разметки документов, который используется для создания вебстраниц и определения их структуры и содержимого. Он состоит из набора тегов и атрибутов, которые позволяют описать содержание и форматирование веб-страницы.
- CSS (Cascading Style Sheets) – формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки.
- JavaScript – язык программирования, который используется для создания интерактивных веб-страниц и приложений. JavaScript позволяет добавлять функциональность на веб-страницы, например переключение между вкладками, создание анимации и проверка пользовательских данных перед отправкой на сервер. JavaScript работает только в браузере, поэтому используется только на клиентской части приложения.
- Bootstrap – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML и CSS шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript расширения

Для этого стека была выбрана кроссплатформенная интегрированная среда разработки PyCharm. Он обеспечивает удобное редактирование кода на различных языках, особенно Python. С его помощью можно быстро создать проект, он подсвечивает синтаксис кода и помогает автоматически исправлять ошибки. PyCharm предлагает широкий набор функций и инструментов. Среда разработки представлена на рисунке 1.2.

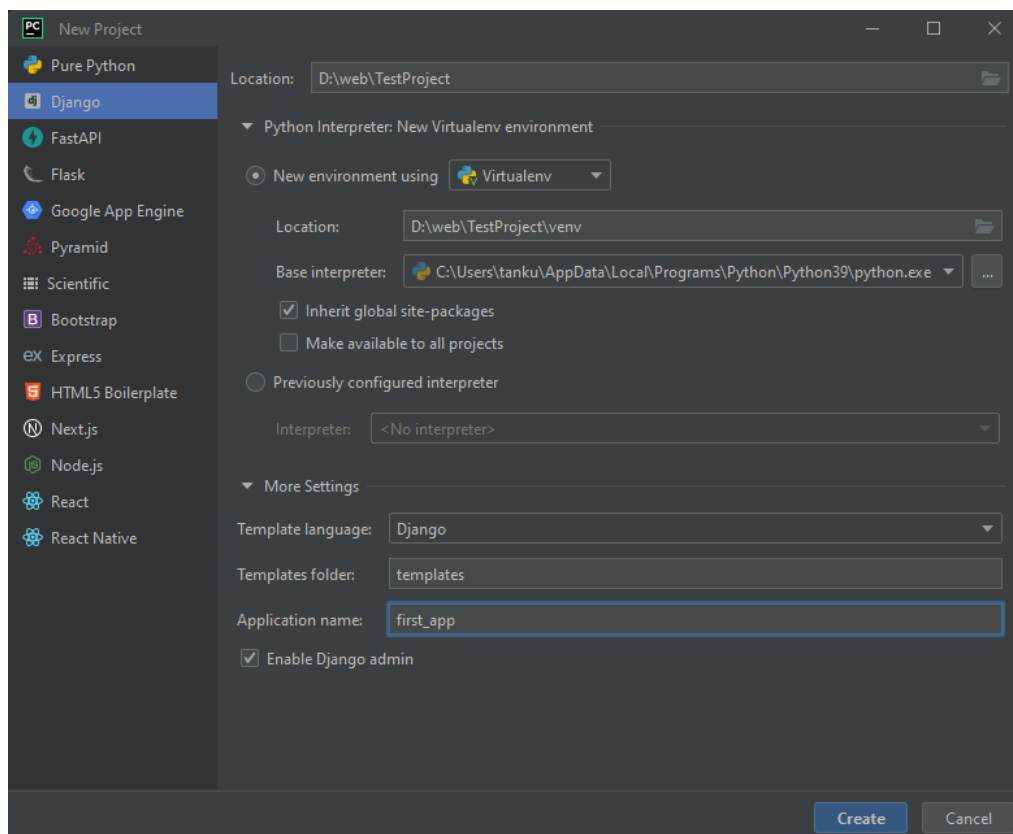


Рисунок 1.2 – Редактор кода PyCharm

Для разработки базы данных, необходимой для работы приложения, была использована стандартная для предприятия и указанная в техническом задании СУБД PostgreSQL. Управление СУБД осуществляется через графическую оболочку SQL Manager for PostgreSQL (рисунок 1.3).

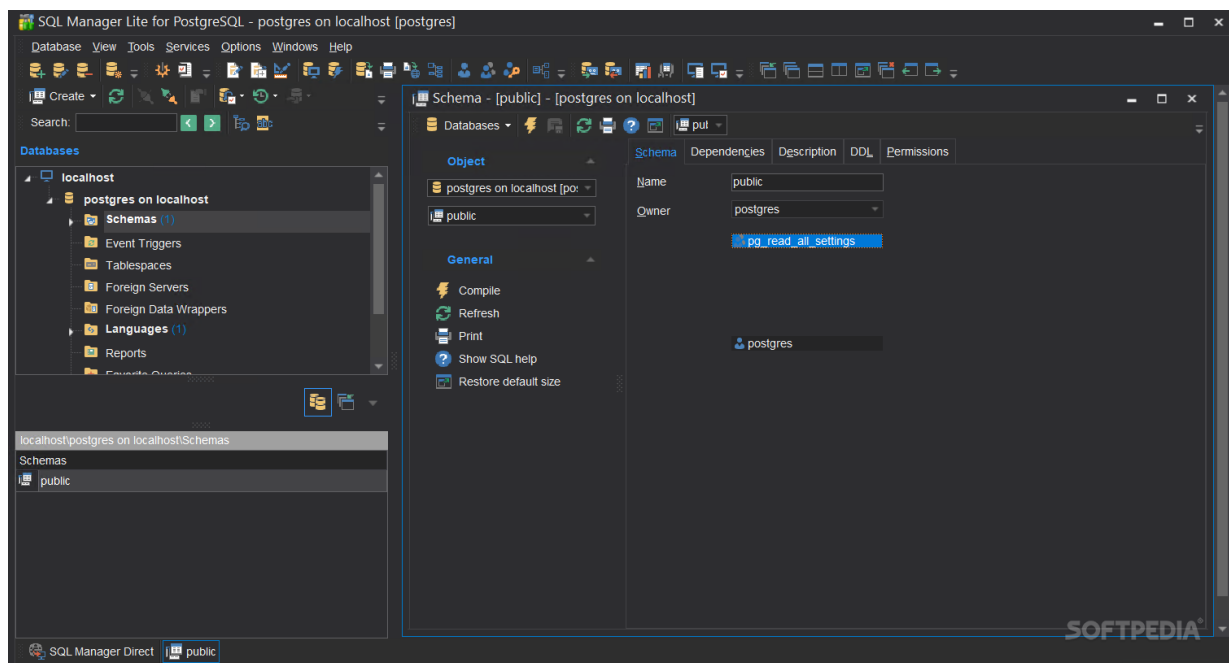


Рисунок 1.3 – SQL Manager for PostgreSQL

Вывод: в соответствии с техническим заданием на разработку веб-сервиса «Анализ наличия и движения МТР на складах комбината» в данной главе:

1. Проведен анализ предметной области. Определены термины, используемые в информационной системе;
2. Построены модели бизнес-процесса;
3. Описаны инструментальные средства, необходимые для реализации веб-сервиса.



## ГЛАВА 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ

### 2.1 Проектирование базы данных

Модель «сущность-связь» (Entity-Relationship model) – модель данных, позволяющая описывать концептуальные схемы предметной области. Используется при концептуальном проектировании баз данных.

Информационная модель «сущность-связь» больше, чем, метод обработки данных, который используется для описания отношений между объектами, называемыми «сущностями», в информационной системе [7].

ER-модель представляет информацию в виде графической диаграммы, которая состоит из трех основных компонентов: сущности, атрибутов и связей.

- Сущность (Entity) – объект или концепция, описывающая что-то вместе в категории объектов;
- Атрибут (Attribute) – характеристика сущности;
- Связь (Relationship) – отношение, которое связывает две или больше сущностей.

ER-модель позволяет проектировать сложные информационные системы, а также определять логику связей между сущностями. Это помогает разработчикам создавать более эффективные и точные базы данных любого уровня сложности. С помощью ER-модели можно проектировать логическую модель базы данных [8].

Для реализации функционала, определенного техническим заданием к приложению, была дана база данных cscd.

Часть базы данных имеет следующие сущности с атрибутами.

На рисунке 2.1 представлена таблица «sc\_cmr.mtr\_certificate», которая отвечает за хранение данных по МТР сертификатам.

mtr_certificate						
Fields	Properties	Foreign Keys	Checks	Indices	Triggers	Rules
Fields	Properties	Foreign Keys	Checks	Indices	Triggers	Rules
Field Name	Field Type	Key	Not Null	Default	Description	
mc_mtr_code	integer		<input checked="" type="checkbox"/>		код_мтр	
mc_number_substring	integer		<input checked="" type="checkbox"/>		номер_подстроки	
mc_country_id_manufacturer	integer		<input type="checkbox"/>	Null	страна_ID_изготовитель	
mc_name_manufacturer_id	integer		<input type="checkbox"/>	Null	изготовитель_предприятие_ID	
mc_country_id	integer		<input type="checkbox"/>	Null	страна_ID	
mc_count_day_guaran	smallint		<input type="checkbox"/>	Null	количество_дней_гарантии	
mc_warranty_period_year	smallint		<input type="checkbox"/>	Null	срок_гарантии_год	
mc_warranty_period_month	smallint		<input type="checkbox"/>	Null	срок_гарантии_месяц	
mc_count_pressing	smallint		<input type="checkbox"/>	Null	количество_запрессовок	
mc_certificate_date	date		<input type="checkbox"/>	Null	сертификат_дата	
mc_date_end_storage	date		<input type="checkbox"/>	Null	дата_окончания_хранения	
mc_date_manufacturing	date		<input type="checkbox"/>	Null	дата_изготовления	
mc_date_recheck	date		<input type="checkbox"/>	Null	дата_перепроверки	
mc_create_ts	timestamp(0)		<input type="checkbox"/>	Null	дата_ввода	
mc_presented_date_cert	timestamp(6)		<input type="checkbox"/>	Null	дата_предъявления_сертификата	
mc_accepted_date_cert	timestamp(6)		<input type="checkbox"/>	Null	дата_приемки_сертификата	
mc_create_man_ts	timestamp(0)		<input type="checkbox"/>	Null	дата_ввода_изготовитель	
mc_number_receipt_order	varchar(20)		<input checked="" type="checkbox"/>		номер_приходного_ордера	
mc_certificate_number	varchar(30)		<input type="checkbox"/>	Null	сертификат_номер	
mc_delivery_brand	varchar(50)		<input type="checkbox"/>	Null	поставка_по_марке	
mc_delivery_assort	varchar(50)		<input type="checkbox"/>	Null	поставка_по_сортаменту	
mc_batch_number	varchar(20)		<input type="checkbox"/>	Null	номер_партии	
mc_melting_number	varchar(20)		<input type="checkbox"/>	Null	номер_плавки	
mc_gost_chemical_compos	varchar(50)		<input type="checkbox"/>	Null	гост_химсостав	
mc_gost_assort	varchar(50)		<input type="checkbox"/>	Null	гост_сортамент	
mc_gost_tech_conditions	varchar(50)		<input type="checkbox"/>	Null	гост_техусловия	
mc_create_user	varchar(50)		<input type="checkbox"/>	Null	кто_ввел	
mc_presented_user_cert	varchar(50)		<input type="checkbox"/>	Null	кто_предъявил_сертификат	
mc_accepted_user_cert	varchar(50)		<input type="checkbox"/>	Null	кто_принял_сертификат	
mc_number_folder_cert	varchar(20)		<input type="checkbox"/>	Null	номер_папки_сертификата	
mc_conclusion_cert	varchar		<input type="checkbox"/>	Null	заключение_по_сертификату	
mc_article	varchar(50)		<input type="checkbox"/>	Null	артикул	
mc_variety	varchar(50)		<input type="checkbox"/>	Null	сорт	
mc_color	varchar(50)		<input type="checkbox"/>	Null	цвет	
mc_stamp	varchar(50)		<input type="checkbox"/>	Null	марка	
mc_recipe	varchar(50)		<input type="checkbox"/>	Null	рецептура	
mc_name_manufacturer	varchar		<input type="checkbox"/>	Null	предприятие_изготовитель	
mc_number_pressing	varchar(50)		<input type="checkbox"/>	Null	номер_запрессовки	
mc_factory_number	varchar(50)		<input type="checkbox"/>	Null	заводской_номер	
mc_allowance	varchar		<input type="checkbox"/>	Null	допуск	
mc_type_cert	varchar		<input type="checkbox"/>	Null	тип_сертификата	
mc_manufacturer	varchar(50)		<input type="checkbox"/>	Null	изготовитель	
mc_create_man_user	varchar		<input type="checkbox"/>	Null	кто_ввел_изготовитель	
mc_code_okvd	varchar		<input type="checkbox"/>	Null	код_ОКВЭД	

Рисунок 2.1 – Таблица «mtr\_certificate»

Данная сущность имеет следующие атрибуты:

- Код МТР;
- Номер подстроки;
- Идентификатор страны-изготовителя (начального);
- Идентификатор предприятия-изготовителя;
- Идентификатор страны поставщика;
- Количество дней гарантии;

- Срок гарантии (лет);
- Срок гарантии (месяцев);
- Количество запрессовок;
- Дата сертификата (Дата среза информации (формулирования сертификата));
- Дата окончания хранения материала;
- Дата изготовления МТР;
- Дата перепроверки (перепроверка сроков хранения мтр);
- Дата ввода (техническое поле);
- Дата предъявления сертификата (ответственный человек предъявляет сопроводительный сертификат при оприходовании на склад);
- Дата приемки сертификата;
- Изготовитель дата ввода (техническое поле);
- Номер приходного ордера;
- Номер сертификата;
- Поставка по марке;
- Поставка по сортаменту;
- Номер партии;
- Номер плавки;
- Номер ГОСТа химсостава;
- Номер ГОСТа сортамента;
- Номер ГОСТа техусловий;
- Номер папки сертификата;
- Заключение по сертификату;
- Артикул;
- Сорт, т.е. качество товара;
- Цвет;
- Марка, т.е. название, знак, символ идентификации продукции;
- Рецептатура;

- Название предприятия-изготовителя;
- Номер запрессовки;
- Заводской номер;
- Допуск, т.е. допустимые отклонения от указанных марок и сортов по ГОСТУ;
- Тип сертификата (например, копия или оригинал);
- Код ОКВЭД

Первичный ключ является составным из следующих полей: `mc_mtr_code` (код МТР), `mc_number_receipt_order` (номер приходного ордера), `mc_number_substring` (номер подстроки).

Таблица «`sc_cmr.mtr_moving`» отвечает за движение МТР и хранит следующие атрибуты:

- Балансовый счет получателя;
- Балансовый счет отправителя;
- Код МТР;
- Стоимость, т.е. общая цена МТР;
- Подразделение-получатель;
- Подразделение-отправитель;
- МОЛ-получатель;
- МОЛ-отправитель;
- Номер приходного ордера;
- Номер подстроки;
- Цена за 1 единицу МТР;
- Количество МТР;
- Идентификатор строки документа;

Первичным ключом является поле `pdr_row_id` (Идентификатор строки документа).

Схема в PostgreSQL – это пространство имен, которое содержит именованные объекты базы данных, такие как таблицы, представления,

индексы, типы данных, функции и операторы [9]. База данных разделена на несколько схем, использовалась схема `sc_cmrf` для выполнения технического задания (рисунок 2.2).

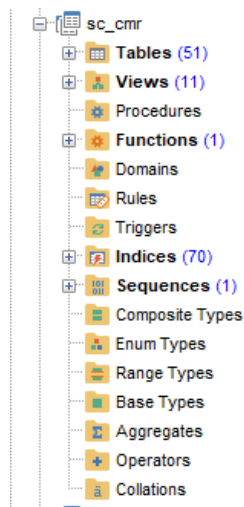


Рисунок 2.2 – Схема `sc_cmrf`

Для реализации функционала необходимого для прямого взаимодействия с данными на сервере следует использовать хранимые процедуры, функции или материализованные представления.

С точки зрения PostgreSQL хранимая процедура – это предопределенный код, назначение которого, заключается в обработке данных в базе данных. Она может реализовывать различные операции, например, изменение или удаление данных, а также выполнения нагруженных запросов к базе данных [10].

Функция базы данных – представляет собой блок кода на языке SQL или любом другом его диалекте. Функции могут использоваться для повторяющихся действий, упрощения работы с данными и улучшения производительности запросов. Они могут возвращать результаты, в отличие от хранимых процедур, принимать параметры и использовать множество встроенных функций и операторов [11].

Материализованные представления в PostgreSQL – специальный вид представлений, который содержит реальные данные, а не просто запрос к данным как обычные представления. Они являются кэшированными

результатами запросов, сохраненными в памяти, что позволяет ускорить выполнение сложных запросов снова и снова [12][13].

Преимущества материализованных представлений:

- Повышение производительности: материализованные представления сохраняют результаты запроса на диске или в памяти, что позволяет избежать повторного выполнения тяжелых запросов.
- Улучшение отзывчивости системы: за счет кэширования данных запросы выполняются быстрее, что повышает скорость отклика системы.
- Возможность предварительного вычисления сложных запросов: материализованные представления позволяют сохранить результаты сложных запросов заранее, что упрощает их повторное использование.

Недостатки материализованных представлений:

- Актуальность данных: поскольку данные в материализованных представлениях кэшируются, может возникнуть проблема с их актуальностью. Поэтому необходимо периодически обновлять материализованные представления.
- Использование дополнительного пространства: хранение результатов запросов в материализованных представлениях требует дополнительного пространства на диске.
- Сложность обновления: обновление данных в материализованных представлениях может потребовать дополнительных усилий и контроля, особенно если данные часто изменяются.

Были определены следующие материализованные представления на базе данных, представленные на рисунке 2.3.

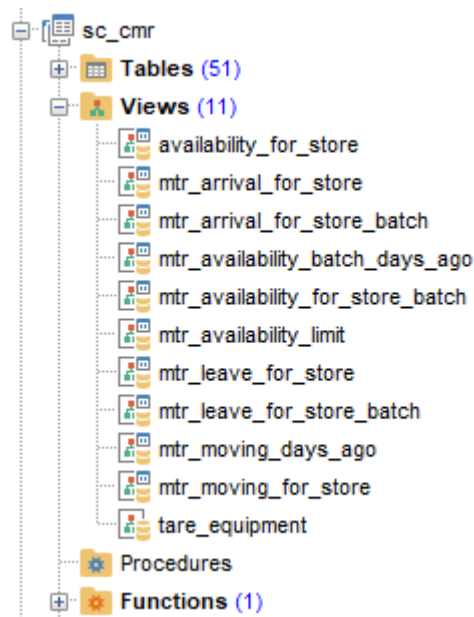


Рисунок 2.3 – Материализованные представления в базе данных

Для вывода наличия всех МТР на складе было создано следующее представление (листинг 1).

*Листинг 1*

```
CREATE MATERIALIZED VIEW sc_cmr.availability_for_store(
    mtr_div_no_receiver,
    mtr_keep_no_receiver,
    mtr_code,
    mtr_name,
    mtr_characteristic,
    mtr_quantity,
    ut_code,
    ut_designation,
    mtr_cost)
AS
SELECT afs.mtr_div_no_receiver,
       afs.mtr_keep_no_receiver,
       afs.mtr_code,
       mtr.mtr_name,
       mtr.mtr_characteristic,
       afs.mtr_quantity - COALESCE(lfs.mtr_quantity, 0::numeric) AS mtr_quantity,
       mtr.ut_code,
       un.ut_designation,
       afs.mtr_cost - COALESCE(lfs.mtr_cost, 0::numeric) AS mtr_cost
FROM sc_cmr.mtr_arrival_for_store afs
     LEFT JOIN sc_cmr.mtr_leave_for_store lfs ON afs.mtr_div_no_receiver =
lfs.mtr_div_no_sender AND afs.mtr_keep_no_receiver =
lfs.mtr_keep_no_sender AND afs.mtr_code = lfs.mtr_code
     LEFT JOIN sc_cmr.mtr mtr ON mtr.mtr_code = afs.mtr_code
     LEFT JOIN ref.units un ON un.ut_code = mtr.ut_code;
```

Наличие материалов на складе без движения с учетом заданной даты представлено на листинге 2.

```

CREATE MATERIALIZED VIEW sc_cmr.mtr_availability_batch_days_ago(
    mtr_code,
    mtr_name,
    mtr_characteristic,
    mtr_number_receipt_order,
    mtr_number_substring,
    mtr_div_no_receiver,
    mtr_keep_no_receiver,
    mtr_quantity,
    ut_code,
    ut_designation,
    mtr_cost,
    days_ago)
AS
SELECT afs.mtr_code,
       mtr.mtr_name,
       mtr.mtr_characteristic,
       afs.mtr_number_receipt_order,
       afs.mtr_number_substring,
       afs.mtr_div_no_receiver,
       afs.mtr_keep_no_receiver,
       afs.mtr_quantity,
       mtr.ut_code,
       un.ut_designation,
       afs.mtr_cost,
       mda.days_ago
FROM   sc_cmr.mtr_availability_for_store_batch afs
       JOIN sc_cmr.mtr_moving_days_ago mda ON afs.mtr_code = mda.mtr_code AND
afs.mtr_number_receipt_order::text = mda.mtr_number_receipt_order::text
       AND afs.mtr_number_substring = mda.mtr_number_substring
       LEFT JOIN sc_mtr.mtr ON mtr.mtr_code = afs.mtr_code
       LEFT JOIN ref.units un ON un.ut_code = mtr.ut_code
WHERE  afs.mtr_quantity > 0::numeric;

```

Сравнение фактического запаса МТР на складах с нормативным лимитом, то есть расчет финансового анализа представлен в Приложении Б.

Для просмотра движения материалов было создано следующее представление (Приложение Б).

## 2.2 Разработка серверной части приложения

Для реализации веб-сервиса «Анализ наличия и движения МТР», определен и настроен Django проект (рисунок 2.4).



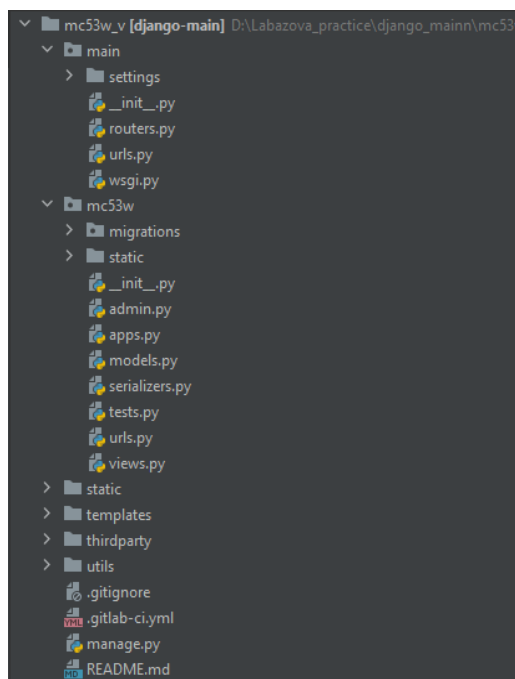


Рисунок 2.4 – Структура Django проекта

Django фреймворк позволяет создавать Django apps – модули, которые реализуют определенный функционал в проекте. Он может содержать свои модели, представления и шаблоны, статические файлы, утилиты и прочее [14]. Django app позволяет организовывать код на Django логически и модульно.

При проектировании серверной части веб-сервиса были определены настройки в файле settings.py. В параметр INSTALLED\_APPS, который отображает подключенные к проекту приложения, были добавлены сторонние библиотеки rest-framework и corsheaders [15], а также созданный Django модуль mc53w. Значение параметра INSTALLED\_APPS (листинг 3).

*Листинг 3*

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # Добавляем приложения
    'main',
    'mc53w',
    'rest_framework',
    'utils'
]
```

Файлы, которые используются для отображения веб-страницы и не подвержены изменению в процессе работы веб-сайта, называются статическими файлами. В основном это файлы изображений, стилей CSS и скрипты JavaScript. Django имеет встроенную систему для управления статическими файлами, которую можно настроить в файле настроек проекта.

В основном статические файлы хранят в директории `/static/`, далее структура определяется разработчиком проекта, на этапе проектирования, структура папки статических файлов (рисунок 2.5).

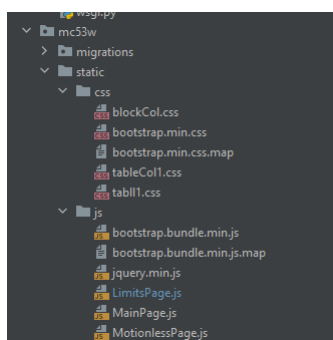


Рисунок 2.5 – Статические файлы

Для того, чтобы фреймворк Django мог обрабатывать статические файлы, была добавлена запись в файл `settings.py` Django проекта. Эта запись в параметр `STATICFILES_DIRS`, которая отражает, где Django серверу искать статистические файлы проекта. Код, добавленный в проект представлен на листинге 6.

*Листинг 4*

```
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, 'mc53w', 'static'),  
]
```

Связь с базой данных устанавливается путем задания необходимых параметров для переменной `DATABASES` с ключами, соответствующими используемой базе данных:

— `ENGINE` – указание используемого драйвера для работы с конкретной СУБД;

- NAME – наименование базы данных;
- USER – пользователь базы данных;
- PASSWORD – пароль пользователя базы данных;
- HOST – адрес сервера базы данных;
- PORT – порт сервера базы данных

Пример подключения к базе данных (листинг 5).

*Листинг 5*

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'HOST': 'eci-db1-dev.eci.local',
        'NAME': 'cscd',
        'SCHEME': 'auth',
        'PORT': '5432',
        'USER': 'django_auth_user',
        'PASSWORD': 'django_auth_user',
    }, #
    # 'local': {
    #     'ENGINE': 'django.db.backends.sqlite3',
    #     'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    # },
    'sc_cmr': {
        'ENGINE': 'django.db.backends.postgresql',
        'HOST': 'eci-db1-dev.eci.local',
        'NAME': 'cscd',
        'PORT': '5432',
        'USER': 'd79480',
        'PASSWORD': '3498',
        'SCHEME': 'sc_cmr'
    },
    'cmr': {
        'ENGINE': 'django.db.backends.postgresql',
        'HOST': 'eci-db1-dev.eci.local',
        'NAME': 'cscd',
        'PORT': '5432',
        'USER': 'd79480',
        'PASSWORD': '3498',
        'SCHEME': 'cmr'
    }
}
```

При подключении нескольких баз данных в Django проект, необходимо настроить роутинг. Django имеет свой собственный роутинг баз данных, который позволяет определить, какие базы данных будут использоваться для работы с моделями Django (листинг 6).

*Листинг 6*

```
class DbRouter:
    def db_for_read(self, model, **hints):
        return model.database if hasattr(model, 'database') else 'default'
    def db_for_write(self, model, **hints):
```

```

    return model.database if hasattr(model, 'database') else 'default'
def allow_relation(self, obj1, obj2, **hints):
    return True
def allow_migrate(self, db, app_label, model_name=None, **hints):
    return True

```

Чтобы работать с базами данных в Django так же были созданы классы моделей. Модель представляет собой класс Python, который отображает структуру базы данных. Модели используются для определения структуры данных, которые будут храниться в базе данных. Они содержат поля и поведение данных, определяют типы данных для каждого поля, а также связи между различными моделями.

Основные компоненты модели в Django:

- Поля (Fields): Поля определяют тип данных, который может храниться в базе данных. Например, CharField для строк, IntegerField для целых чисел и т.д.
- Связи (Relationships): Django позволяет определять различные типы связей между моделями, такие как ForeignKey, OneToOneField, ManyToManyField и др. Связи позволяют устанавливать связи между записями в разных таблицах базы данных.
- Методы и свойства класса: Модели могут содержать методы и свойства для выполнения различных операций с данными, таких как получение, обновление и удаление объектов.

Django предоставляет удобный способ автоматического создания моделей исходя из схемы базы данных. Сгенерировать классы моделей можно при помощи команды `python manage.py inspectdb`, которая соберет информацию о таблицах в БД и составит листинг моделей [16]. Без использования ключа `–database`, он будет обращаться к базе данных, которая определена как база данных по умолчанию, то есть имеет ключ `'default'`.

Пример класса модели приведен в Приложении В.

Функции вызова процедур из базы данных добавляются в файл `mc53w/views.py` (Приложение Г). На листинге 7 показан пример вызова

представления просмотра наличия МТР по партиям, лежащих на складах больше 90 дней (листинг 7).

*Листинг 7*

```
# наличие мтр по партиям, лежащих на складах больше 90 дней
class AvailabilityDays(generics.ListAPIView):
    pagination_class = Pagination
    queryset = AvailabilityBatchDaysAgo.objects.all().order_by('mtr_code').values()
    serializer_class = AvailabilityBatchDaysAgoSerializer
    filter_backends = [filters.OrderingFilter]
    ordering_fields = '_all_'
    def get_queryset(self):
        days = self.request.query_params.get('days_ago')
        div_no_receiver = self.request.query_params.get('mtr_div_no_receiver')
        keep_no_receiver = self.request.query_params.get('mtr_keep_no_receiver')
        if div_no_receiver and keep_no_receiver:
            queryset = AvailabilityBatchDaysAgo.objects.filter(days_ago__gte=days,
mtr_div_no_receiver=div_no_receiver, mtr_keep_no_receiver=keep_no_receiver)
        elif days:
            queryset = AvailabilityBatchDaysAgo.objects.filter(days_ago__gte=days)
        # else:
        #     queryset = AvailabilityBatchDaysAgo.objects.all()
        return queryset
```

Выделение ресурсов на стадии проектирования позволяет создавать масштабируемые и гибкие веб-приложения. Основные ресурсы, выделенные для модуля mc53w:

- /mtr-certificate/ - просмотр МТР сертификата;
- /mtr-moving-for-store/ - просмотр движения МТР;
- /availability/ - вывод всего наличия МТР на складе;
- /availabilitydays/ - вывод МТР без движения;
- /availability-limit/ - финансовый анализ по нормируемым балансовым счетам

Django REST Framework (DRF) является мощным инструментом для создания API на основе фреймворка Django. Он предоставляет различные инструменты, среди которых особое место занимают сериализаторы и представления.

Сериализаторы в DRF являются ключевым элементом для преобразования объектов Python в форматы данных, которые могут быть легко сериализованы и переданы через API [17]. С помощью сериализаторов можно выполнять процессы преобразования объектов в словари, JSON-объекты и

обратно. Сериализаторы также предоставляют возможность валидации данных, а также управления тем, как данные будут отображаться или приниматься. На рисунке 2.6 представлен пример обработки сериализатора AvailabilityBatchDaysAgoSerializer, в котором поля mtr\_keep\_no\_receiver (МОЛ), mtr\_div\_no\_receiver (подразделение), mtr\_code (код МТР) добиваются нулями до необходимой длины, а поле mtr\_cost (стоимость) форматируется до 2-х знаков после запятой.

```
class AvailabilityBatchDaysAgoSerializer(serializers.ModelSerializer):
    mtr_keep_no_receiver = models.SmallIntegerField(blank=True, null=True)
    mtr_div_no_receiver = models.SmallIntegerField(blank=True, null=True)
    mtr_cost = models.DecimalField(max_digits=19, decimal_places=4, blank=True, null=True)
    mtr_code = models.IntegerField(primary_key=True)

    def to_representation(self, instance):
        data = super().to_representation(instance)
        if 'mtr_cost' in data:
            rate_dec = float(data['mtr_cost'])
            data['mtr_cost'] = "{:.2f}".format(rate_dec)
        if 'mtr_code' in data:
            value_int = str(data['mtr_code'])
            while len(value_int) < 7:
                value_int = '0'+value_int
            data['mtr_code'] = value_int
        if 'mtr_div_no_receiver' in data:
            value_int = str(data['mtr_div_no_receiver'])
            if value_int is not None:
                while len(value_int) < 3:
                    value_int = '0'+value_int
            if value_int == 'None':
                value_int = ''
            data['mtr_div_no_receiver'] = value_int
        if 'mtr_keep_no_receiver' in data:
            val = str(data['mtr_keep_no_receiver'])
            while len(val) < 2:
                val = '0'+val
            if val == 'None':
                val = ''
            data['mtr_keep_no_receiver'] = val
        return data
```

Рисунок 2.6 – Сериализатор AvailabilityBatchDaysAgoSerializer

Представления в DRF отвечают за обработку запросов API и возвращение данных. Они определяют, как API взаимодействует с соответствующими сериализаторами для выполнения различных операций, таких как создание, чтение, обновление и удаление данных. Представления могут быть написаны как в функциональном, так и в классовом стиле.

## 2.3 Проектирование клиентской части

Разработка клиентской части приложения играет ключевую роль в создании современного и успешного продукта. Она необходима для обеспечения удобного и интуитивно понятного взаимодействия пользователя с приложением. Для реализации клиентской части приложения были выполнены следующие шаги:

- Создание дизайна пользовательского интерфейса (UI): это включает в себя описание стилей для веб-страниц, проектирование и создание веб-страниц, чтобы обеспечить приятный и функциональный внешний вид приложения.
- Настройка поведения пользовательского интерфейса (UX): это важная часть, где обеспечивается корректное взаимодействие пользователя с приложением, включая проверку ввода данных и создание удобного пользовательского опыта [18].
- Настройка взаимодействия между клиентской и серверной частями: это включает в себя работу над передачей данных между клиентом и сервером, а также настройку логики работы приложения для эффективного обмена информацией.

В Django функциональность позволяет использовать родительские и дочерние шаблоны для удобной организации кода интерфейса [19]. Родительские шаблоны представляют собой общую структуру HTML-документа или контейнера, где разработчик может определять различные блоки кода. Дочерние шаблоны, в свою очередь, предназначены для многократного использования в различных частях других HTML-страниц. Это позволяет повторно использовать код и упрощает работу над макетом и структурой веб-приложения. Родительский шаблон и дочерние шаблоны представлены в приложении Д.

Блок с МТР сертификатом является дочерним шаблоном (рисунок 2.7) Он содержит в своей реализации JavaScript [20] и CSS код, представленный в приложении Е.



Рисунок 2.7 – Внешний вид блока с данными для МТР сертификата

На рисунке 2.8 показана главная страница сайта, код которого приведен в приложении Д.

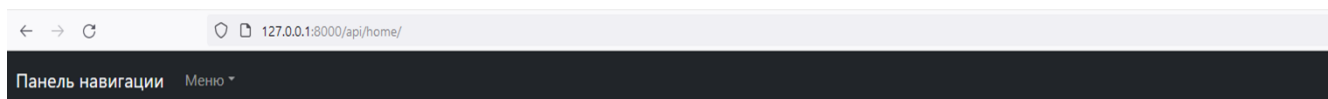


Рисунок 2.8 – Внешний вид главной страницы

Внешний вид страницы с наличием МТР на складах можно увидеть на рисунке 2.9. Код вывода данных в таблицу с наличием МТР приведен в приложении Е.

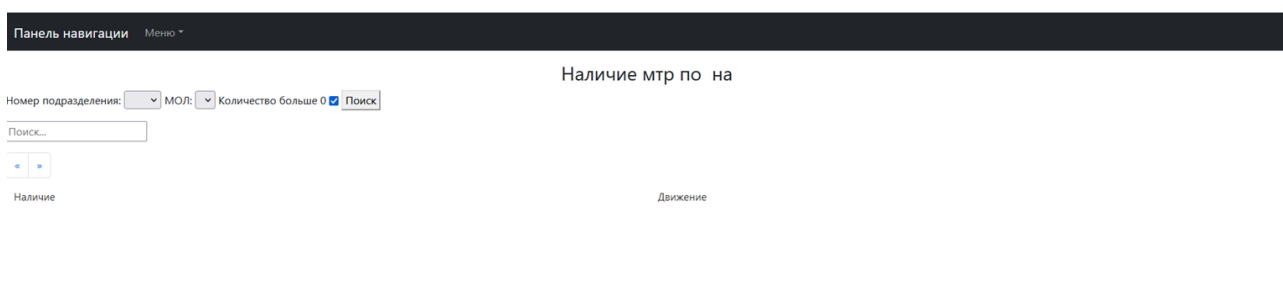


Рисунок 2.9 – Внешний вид страницы с наличием МТР



Взаимодействие клиентской и серверной части происходит путем отправки формы с методом GET на сервер, если были переданы параметры поиска, такие как номер подразделения или номер МОЛ, представление вернет отфильтрованный запрос, иначе вернутся все данные без фильтрации.

Вывод информации о МТР движении представляет собой таблицу с данными с фильтрацией по МТР коду, номеру подразделения, номеру МОЛ (рисунок 2.10). Код представлен в приложении Е.

Наличие							Движение											
Склад/мол	Код мтр	Ед.измерения	Характеристика мтр		Количество	Стоимость,руб	внутреннее перемещение	Тип	Номер	Дата создания	Отправитель	Получатель	Стоимость	Количество приход	Количество расход	Номер приходного ордена	Номер подстроки	
910/0	5040131	796(шт)	ККТ "ПРИМ 08-Ф" ТУ 4017-091-23045784-2016		2.0000000	52711.86												
780/0	1333923	776(шт)	СТРУЖКА КАБЕЛЬНАЯ GT-300С УПАК. 100 ШТ. НЕЙЛОН 3.6*300		4.0000000	1960.00												
780/0	5060092	796(шт)	Картридж лазерный HP 824A C8382A		2.0000000	56049.96												

Рисунок 2.10 – Внешний вид таблицы «Движение»

Для таблиц реализована пагинация (рисунок 2.11), при которой динамически формируется количество страниц. Если пагинация присутствует, то на каждой странице таблицы выводится по 400 записей (полей) (приложение Ж).

«	1	2	3	4	5	6	7	8	9	10	»
---	---	---	---	---	---	---	---	---	---	----	---

Рисунок 2.11 – Пагинация таблицы

Возможность выбрать порядок сортировки полей таблицы показана в листинге 8.

Листинг 8

```
//сортировка таблицы
function 41ortable(columnIndex){
  const rows = Array.from(body.rows);
  const isDescending = columnSortState[columnIndex] === 'desc';
  rows.sort((a,b)=>{
    const aData = parseFloat(a.cells[columnIndex].textContent) || 0;
    const bData = parseFloat(b.cells[columnIndex].textContent) || 0;
    if (aData < bData){
      return isDescending ? 1: -1;
    }
    if (aData > bData) {
      return isDescending ? -1: 1;
    }
    return 0;
  })
}
```

Вывод: в соответствии с техническим заданием на разработку программного продукта в данной главе:

- Спроектирована база данных;
- Разработана серверная часть приложения;
- Разработана клиентская часть приложения

## ГЛАВА 3 ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ

### 3.1 Руководство пользователя. Тестирование приложения

Проведение тестирования необходимо для контроля и обеспечения высокого качества программного продукта. Оно помогает выявить и исправить недочеты, ошибки и дефекты до внедрения продукта в эксплуатацию. Тестирование также позволяет убедиться в том, что программа соответствует техническому заданию, а также функционирует корректно в конкретном окружении.

Программный продукт «Анализ наличия и движения МТР на складах комбината» представляет из себя приложение, содержащее клиентскую часть в виде веб-сервиса, функционирующего в браузере на рабочем месте.

Установка клиентского модуля «Анализ наличия и движения МТР на складах комбината» не требуется, так как система функционирует посредством веб-сервиса.

Запуск модуля осуществляется путем перехода по ссылке, установленной администратором на рабочем месте.

Главная/основная страница: после успешного входа в систему (перехода по ссылке), в браузере должна отобразиться HTML-страница с панелью навигации по сайту (рисунок 3.1).



Рисунок 3.1 – Главная страница сайта

На панели навигации можно выбрать одну из нескольких страниц (см. рисунок 3.2):

- Главная страница;
- Наличие;
- Без движения;
- Финансовый анализ

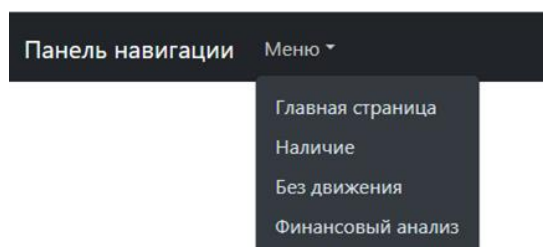


Рисунок 3.2 – Панель навигации

При выборе страницы «Наличие» внешний вид сайта показан на рисунках 3.3 и 3.4.

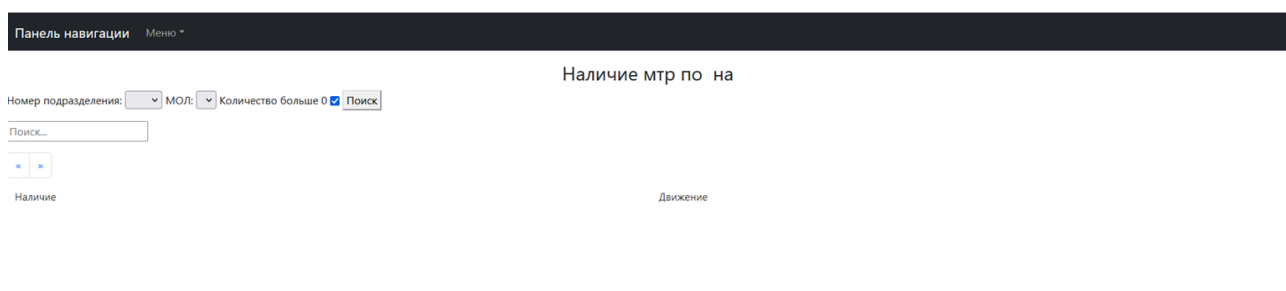


Рисунок 3.3 – Внешний вид сайта со страницей «Наличие»

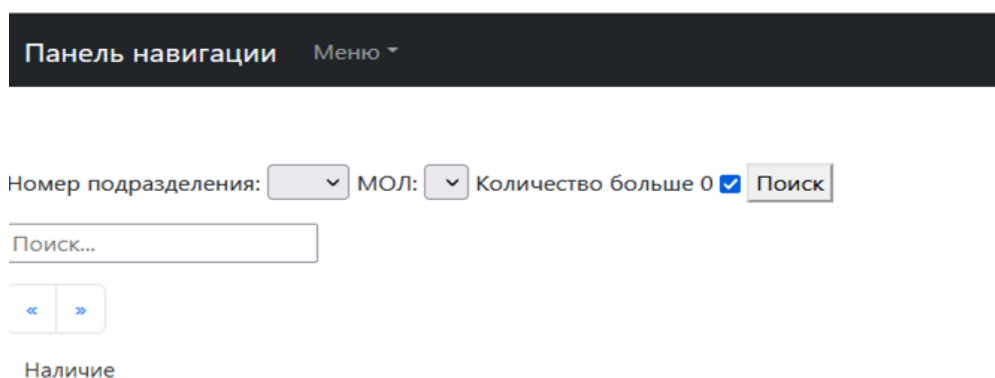


Рисунок 3.4 – Выбор фильтров на странице «Наличие»

Номер подразделения: 008 ▼ МОЛ: 00 ▼ Количество больше 0 ☒ Поиск

Поиск...

« 1 2 3 4 5 6 7 10 »

Наличие

Ссылка/код	Код	Б/г наименование	Удостоверительные метки
------------	-----	------------------	-------------------------

Номер подразделения:  МОЛ:  Количество больше 0 ☒

Наличие

После нажатия кнопки «Поиск» в таблицу с заголовком «Наличие» выводятся данные, как на Рисунке 3.7.

[illegible]

45

По нажатию на любую строку таблицы «Наличие» соседняя таблица «Движение» заполняется данными, отфильтрованными по МТР коду, номеру подразделения, МОЛу (см. рисунок 3.8)

Наличие						Движение										
Склад/мол	Код мтр	Ед.измерения	Характеристика мтр		Количество	Стоимость,руб	Тип	Номер	Дата создания	Отправитель	Получатель	Стоимость	Количество приход	Количество расход	Номер приходного ордена	Номер подстроки
910/0	5040131	796(ШТ)	ККТ "ПРИМ 08-Ф" ТУ 4017-091-23045784-2016		2.0000000	52711.86	внутреннее перемещение	24-780-000010	17.01.2024	744/00	780/00	1960.00	4.0000000		23-744-00-0473	0
780/0	1333923	7730(ПМК)	СТРЕЛКА КАБЕЛЬНАЯ GT-309КС-УПАК, 100 ШТ. НЕЙЛОН 3.6*300		4.0000000	1960.00										
780/0	5060092	796(ШТ)	Карtridge лазерный HP 624A, CE382A		2.0000000	56049.96										

Рисунок 3.8 – Заполнение таблицы «Движение»

По нажатию на любую строку таблицы «Движение» по составному ключу, состоящему из МТР коду, номеру приходного ордена, номеру подстроки происходит фильтрация данных, в случае нахождения подходящего значения пустые поля блока МТР сертификат заполняются данными (см. рисунок 3.9).

МТР сертификат		
Код мтр:	Количество запросов:	Дата изготовления мтр:
Номер приходного ордена:	Номер запросов:	Исполнитель:
Номер подстроки:	Поставка по марки:	Дата перепроверки сроков хранения:
Дата создания сертификата:	Поставка по сертификату:	Дата предъявления сертификата:
Номер сертификата:	Номер партии:	Кто предъявил сертификат:
Срок гарантии (месяцев):	Номер партии:	Дата приема сертификата:
Срок гарантии (лет):	ГОСТ качества:	Кто принял сертификат:
Количество дней гарантии:	ГОСТ сертификат:	Предприятие-изготовитель:
Дата окончания хранения:	ГОСТ текстовики:	Заводской номер:
Номер партии сертификата:	Допуск:	
Артикул:	Заслужение по сертификату:	
Сорт:		
Цвет:	Код ОКБЗД:	
Марка:	Тип сертификата:	
Рецептура:		

Рисунок 3.9 – Заполненный блок «МТР сертификат»

При выборе страницы «Без движения» внешний вид сайта показан на рисунке 3.10.

Панель навигации

Меню

Материалы без движения по более дней на

Количество дней: 90
Номер подразделения:
МОЛ:
Поиск

Поиск по коду мтр...

<
>

Рисунок 3.10 – Внешний вид сайта со страницей «Без движения»

Пользователь, как и на предыдущей странице может выбрать номер подразделения и соответствующие МОЛ из выпадающих списков. Также можно выбрать количество дней из выпадающего списка (90, 60, 30) и ввести частично или полностью МТР код в поле поиска (см. рисунок 3.11). Есть

возможность не выбирать ничего, тогда в таблицу будут выведены все данные без фильтрации.

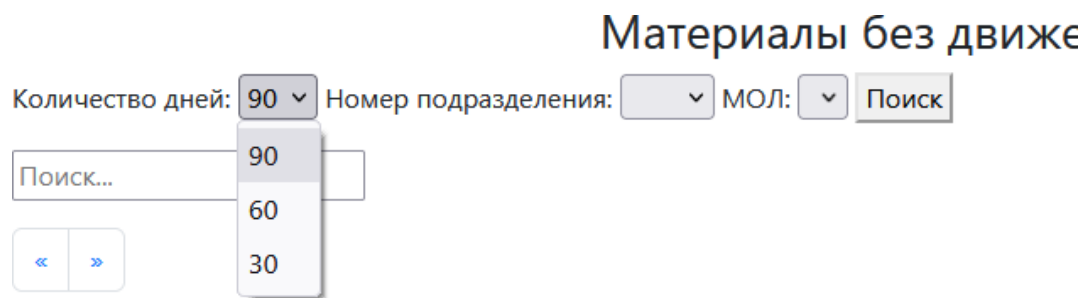


Рисунок 3.11 – Фильтры страницы «Без движения»

Также после вывода данных в таблицу «Наличие» на странице «Без движения» по нажатию на любую строку заполняется таблица «Движение» (рисунок 3.12).



Рисунок 3.12 – Внешний вид страницы «Без движения» с таблицами с данными

Для лучшего отображения большого количества данных в таблице реализована пагинация (см. рисунок 3.13).

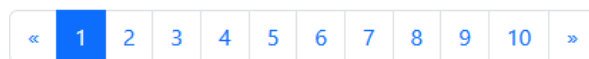


Рисунок 3.13 – Пагинация

Пользователь может сортировать данные для удобного пользования нажатием на заголовки столбцов таблицы (рисунок 3.14 и рисунок 3.15).

Наличие

Склад/мол	Код мтр	Ед.измерения	Характеристика	Количество	Стоимость,руб	Номер приходного ордена	Номер подстроки	Кол-во дней
тл	тл	тл		тл	тл			тл

Рисунок 3.14 – Сортировка таблицы «Наличие» на странице «Без движения»

Наличие

Склад/мол	Код мтр	Ед.измерения	Характеристика мтр	Количество	Стоимость,руб
тл	тл	тл		тл	тл

Рисунок 3.15 – Сортировка таблицы «Наличие» на странице «Наличие»

Также для удобства пользователя блок МТР сертификат может закрываться и открываться по нажатию на значок в правом углу (рисунок 3.16).



Рисунок 3.16 – Открытие и закрытие блока





связанными с изменениями цен, курсов валют, недоступностью определенных материалов и другими факторами, которые могут повлиять на операционную деятельность предприятия.

— Повышение прозрачности и контроля: внедрение программы анализа МТР поможет создать более прозрачные и контролируемые процессы управления материальными ресурсами, улучшив мониторинг и отчетность по задачам управления складской деятельностью.

Внедрение автоматизации бизнес-процессов имеет значительное экономическое значение для компании. Один из основных плюсов автоматизации заключается в снижении издержек на ручное выполнение задач. Автоматизация рутинных и трудоемких операций способна существенно сократить время, необходимое для их выполнения, и уменьшить вероятность ошибок.

Другим важным экономическим преимуществом автоматизации бизнес-процессов является повышение производительности и эффективности. Автоматизация позволяет ускорить процесс выполнения задач и улучшить качество работы, что в конечном итоге может привести к росту объемов производства и снижению издержек.

Автоматизация бизнес-процессов способна упростить и оптимизировать деятельность предприятия, улучшив управление процессами, уменьшив сложность и повысив гибкость. В результате это может привести к сокращению штата сотрудников, что приводит к экономии на заработной плате и других социальных выплатах. Таким образом, внедрение автоматизации бизнес-процессов способно приносить компании значительные экономические выгоды, включая экономию затрат, повышение эффективности, улучшение управления и повышение качества услуг.

Экономический эффект от внедрения автоматизации бизнес-процессов действительно может быть косвенным, так как средства автоматизации не являются непосредственным источником доходов, а скорее служат как вспомогательное средство для повышения эффективности предприятия и

снижения затрат. Основной выгодой от автоматизации является улучшение экономических и хозяйственных показателей функционирования компании, при этом экономический эффект чаще всего проявляется в форме экономии трудовых и финансовых ресурсов.

Следует особо отметить, что снижение трудозатрат на предприятии возможно за счет автоматизации работы с документами, снижения затрат на поиск информации.

## ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы были изучены предметная область, основные термины, инструментальные средства, связанные с информационной системой «Анализ наличия и движения МТР на складах комбината». Состоялось знакомство с ФГУП «Комбинат «Электрохимприбор» и частью его бизнес-процессов. В результате были созданы модели бизнес-процессов, информационная модель «сущность-связь» и сделан вывод о необходимости создания приложения, предназначенного для управления задачами, стоящими перед пользователями системы.

Была изучена база данных со всеми необходимыми сущностями и атрибутами, были реализованы материализованные представления, необходимые для выполнения бизнес-процесса. На основе структуры базы данных были реализованы Django модели и серверная часть информационной системы. С использованием моделей Django были реализованы все необходимые бизнес-функции. Были созданы шаблоны, необходимые для отображения пользовательского интерфейса приложения. С помощью современных технологий таких как, JavaScript, CSS и Bootstrap5 страницы были оформлены с понятным и приятным интерфейсом.

Информационная система «Анализ наличия и движения МТР на складах комбината» после доработки и улучшения функционала приложения будет в дальнейшем интегрирована на предприятии ФГУП «Комбинат «Электрохимприбор».

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

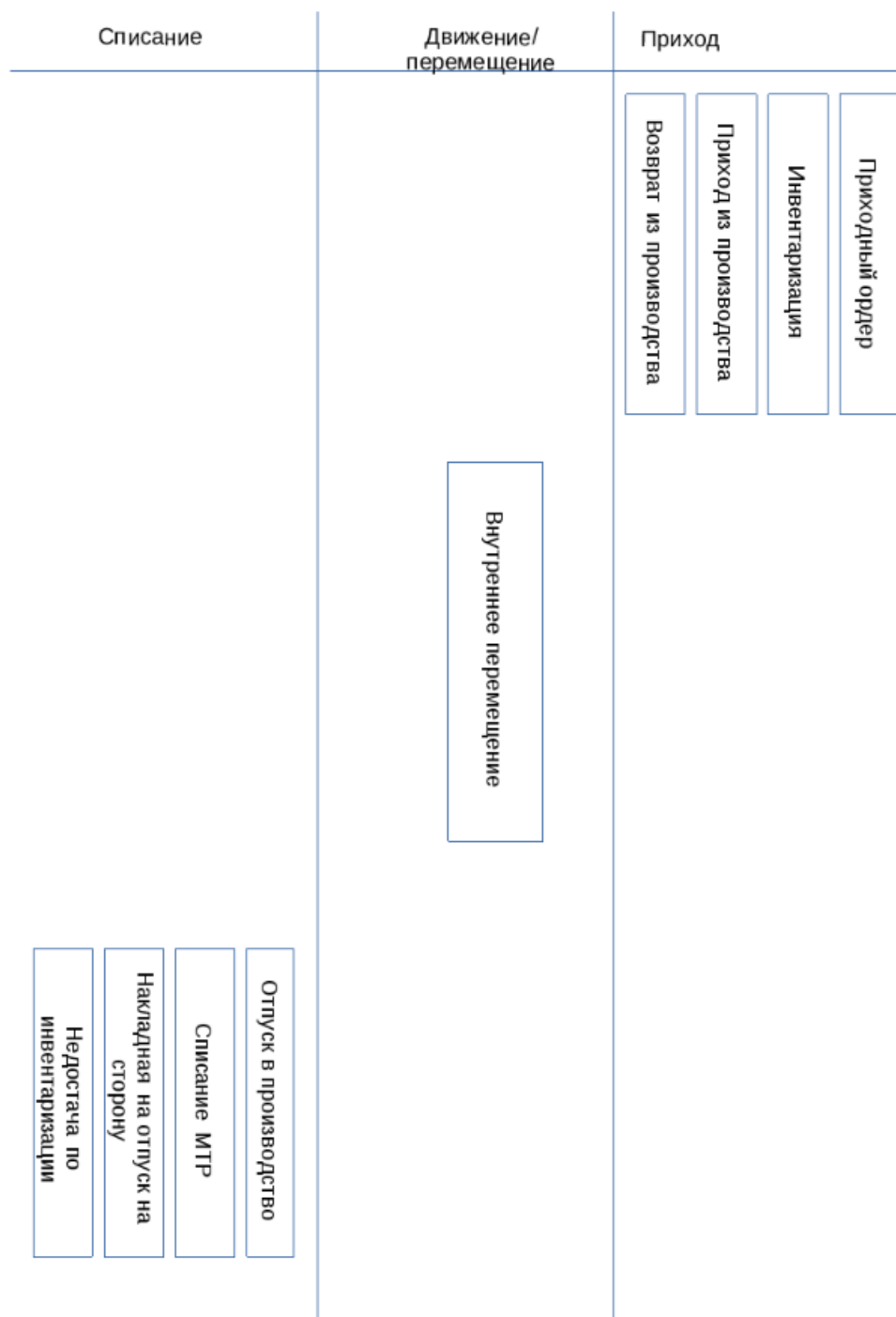
1. АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ, "Управление материально-техническим снабжением", Паспорт витрины данных АСУЖ.МС51Р.15. Редакция 2, 2024. – 46 с.
2. Петров А.Б. Создание и использование витрин (materialized views) в PostgreSQL. — М.: Издательство Техносфера, 2020. — 180 с.
3. Бизнес-процессы: языки моделирования, методы, инструменты / Ф. Шёнталер [и др.]. — Москва : Альпина Паблишер, 2019. — 264 с. — ISBN 978-5-9614-2022-7. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/124474.html> (дата обращения: 21.04.2024).
4. Виафоре, П. Надежный Python / П. Виафоре: БХВ, 2023. – 352 с. – ISBN 978-5-9775-1174-2.
5. Бизнес-процессы промышленного предприятия : учебное пособие / Н.Р. Кельчевская [и др.]. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2016. — 340 с. — ISBN 978-5-7996-1824-7. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/68423.html> (дата обращения: 30.04.2024).
6. Бевзюк Е.А. Регламентация и нормирование труда : учебное пособие для бакалавров / Бевзюк Е.А., Попов С.В. — Москва : Дашков и К, Ай Пи Эр Медиа, 2023. — 211 с. — ISBN 978-5-394-05121-0. — Текст : электронный // IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/124097.html> (дата обращения: 2.05.2024).
7. Павличева, Е. Н. Введение в информационные системы управления предприятием : учебное пособие / Е. Н. Павличева, В. А. Дикарев. — Москва : Московский городской педагогический университет, 2013. — 84 с. — ISBN 2227-8397. — Текст : электронный // Электронно-

- библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/26456.html> (дата обращения: 2.04.2024).
8. Бакланова, О. Е. Информационные системы : учебное пособие / О. Е. Бакланова. — Москва : Евразийский открытый институт, 2008. — 290 с. — ISBN 978-5-374-00052-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/10682.html> (дата обращения: 2.04.2024).
9. Логинов А. С. Разработка схем баз данных в PostgreSQL: практическое руководство. — М.: Диалектика, 2019. — 240 с.
10. Гладышев Д. А. Разработка баз данных в PostgreSQL: управление данными, хранимые процедуры. — СПб.: БХВ-Петербург, 2018. — 224 с.
11. Смирнов Д. В. Разработка пользовательских функций в PostgreSQL. — СПб.: Питер, 2018. — 240 с.
12. Иванов Д. Б. Разработка и оптимизация материализованных представлений в PostgreSQL. — СПб.: БХВ-Петербург, 2019. — 192 с.
13. Сидоров П. А. Материализованные представления в PostgreSQL: синтаксис, возможности, примеры применения. — М.: Издательство Техносфера, 2020. — 224 с.
14. Документация Django framework: официальный сайт. — Обновляется в течение суток. — URL: <https://docs.djangoproject.com> (дата обращения: 16.05.2024). — Текст: электронный.
15. Пакеты Django: официальный сайт. — Обновляется в течении недели. — URL: <https://djangopackages.org> (Дата обращения: 20.05.2024). — Текст: электронный.
16. Документация по библиотеке django-rest framework: официальный сайт. — Обновляется в течение суток. — URL: <https://django-rest-framework.org> (дата обращения: 23.05.2024). — Текст: электронный.
17. Общая документация по языку программирования Python: официальный сайт. — Обновляется в течение суток. — URL: <https://docs.python.org> (дата обращения: 2.05.2024). — Текст: электронный.

18. Mozilla Developer Network - Web technologies documentation: официальный сайт. – Обновляется в течение суток. – URL: <https://developer.mozilla.org/ru/> (Дата обращения: 30.05.2024). – Текст: электронный
19. Иньегски Т. Создание адаптивного интерфейса: современные подходы к веб-разработке / Т. Иньегски, А. Тейлов: – Москва, 2018. – 382 с.
20. Флэнаган Д. JavaScript. Полное руководство / Д. Флэнаган: Диалектика Вильямс, 2021. – 720 с. – ISBN 978-5-907203-79-2. // Про JS 11 ый ист.

## Приложение А (обязательное)

### Swim-lane диаграмма бизнес-процесса





## Приложение Б (обязательное)

### Код материализованных представлений

#### Б.1 Материализованное представление для расчета финансового анализа

```
CREATE MATERIALIZED VIEW sc_cmr.mtr_availability_limit(
    rate,
    mtr_div_no_receiver,
    mtr_cost)
AS
SELECT msr.rate,
    availab.mtr_div_no_receiver,
    availab.mtr_cost
FROM (
    SELECT afs.mtr_div_no_receiver,
        sum(afs.mtr_cost - COALESCE(lfs.mtr_cost, 0::numeric)) AS mtr_cost
    FROM (
        SELECT mm.mtr_div_no_receiver,
            mm.mtr_keep_no_receiver,
            mm.mtr_code,
            mm.mtr_bal_account_receiver,
            sum(mm.mtr_cost) AS mtr_cost
        FROM sc_cmr.primary_document pd
        JOIN sc_cmr.primary_document_row pdr ON pd.pd_doc_id = pdr.pd_doc_id
        JOIN sc_cmr.mtr_moving mm ON pdr.pdr_row_id = mm.pdr_row_id
        JOIN sc_cmr.mtr_doc_confirmat_store dcs ON pd.pd_doc_id =
dcs.pd_doc_id
        JOIN sc_cmr.mtr_normalized_accounts na ON mm.mtr_bal_account_receiver
= na.bal_account
        GROUP BY mm.mtr_div_no_receiver,
            mm.mtr_keep_no_receiver,
            mm.mtr_bal_account_receiver,
            mm.mtr_code
    ) afs
    LEFT JOIN
    (
        SELECT mm.mtr_div_no_sender,
            mm.mtr_keep_no_sender,
            mm.mtr_code,
            mm.mtr_bal_account_sender,
            sum(mm.mtr_cost) AS mtr_cost
        FROM sc_cmr.primary_document pd
        JOIN sc_cmr.primary_document_row pdr ON pd.pd_doc_id = pdr.pd_doc_id
        JOIN sc_cmr.mtr_moving mm ON pdr.pdr_row_id = mm.pdr_row_id
        JOIN sc_cmr.mtr_doc_confirmat_store dcs ON pd.pd_doc_id =
dcs.pd_doc_id
        JOIN sc_cmr.mtr_normalized_accounts na ON mm.mtr_bal_account_sender =
na.bal_account
        GROUP BY mm.mtr_div_no_sender,
            mm.mtr_keep_no_sender,
            mm.mtr_bal_account_sender,
            mm.mtr_code
    ) lfs ON afs.mtr_div_no_receiver = lfs.mtr_div_no_sender AND
afs.mtr_keep_no_receiver = lfs.mtr_keep_no_sender AND afs.mtr_code =
lfs.mtr_code AND afs.mtr_bal_account_receiver = lfs.mtr_bal_account_sender
    GROUP BY afs.mtr_div_no_receiver
    ) availab
    JOIN sc_cmr.mtr_stock_rate msr ON availab.mtr_div_no_receiver = msr.div_no;
```

## Б.2 Материализованное представление для просмотра движения материалов

```
CREATE MATERIALIZED VIEW sc_cmr.mtr_moving_for_store(  
    pd_type_doc,  
    pd_doc_id,  
    pd_doc_nom,  
    pd_doc_date,  
    pdr_row_id,  
    mtr_div_no_sender,  
    mtr_keep_no_sender,  
    mtr_keep_no_receiver,  
    mtr_div_no_receiver,  
    mtr_code,  
    mdm_additional_code,  
    mtr_characteristic,  
    ut_code,  
    mtr_cost,  
    mtr_quantity,  
    mtr_number_receipt_order,  
    mtr_number_substring,  
    pd_create_user,  
    pd_create_ts,  
    dvs_confirmat_ts_1,  
    dvs_confirmat_user_1,  
    dvs_number_registry,  
    pd_confirmat_ts,  
    pd_confirmat_user)  
AS  
    SELECT pd.pd_type_doc,  
           pd.pd_doc_id,  
           pd.pd_doc_nom,  
           pd.pd_doc_date,  
           pdr.pdr_row_id,  
           mm.mtr_div_no_sender,  
           mm.mtr_keep_no_sender,  
           mm.mtr_keep_no_receiver,  
           mm.mtr_div_no_receiver,  
           mm.mtr_code,  
           COALESCE(dm.mdm_additional_code, '0'::smallint) AS mdm_additional_code,  
           mtr.mtr_characteristic,  
           mtr.ut_code,  
           mm.mtr_cost,  
           mm.mtr_quantity,  
           mm.mtr_number_receipt_order,  
           mm.mtr_number_substring,  
           pd.pd_create_user,  
           pd.pd_create_ts,  
           dcs.dvs_confirmat_ts_1,  
           dcs.dvs_confirmat_user_1,  
           dcs.dvs_number_registry,  
           pd.pd_confirmat_ts,  
           pd.pd_confirmat_user  
    FROM sc_cmr.primary_document pd  
         JOIN sc_cmr.primary_document_row pdr ON pd.pd_doc_id = pdr.pd_doc_id  
         JOIN sc_cmr.mtr_moving mm ON pdr.pdr_row_id = mm.pdr_row_id  
         JOIN sc_cmr.mtr_doc_confirmat_store dcs ON pd.pd_doc_id = dcs.pd_doc_id  
         JOIN      sc_cmr.mtr_date_fixing_balances_last      fb      ON      pd.pd_doc_date      >=  
fb.mtr_date_fixing_balances_last  
         LEFT JOIN sc_mtr.mtr ON mtr.mtr_code = mm.mtr_code  
         LEFT JOIN sc_cmr.mtr_drag_metals dm ON dm.mtr_code = mm.mtr_code AND  
dm.mtr_number_receipt_order::text = mm.mtr_number_receipt_order::text AND  
dm.mtr_number_substring = mm.mtr_number_substring  
    WHERE pd.pd_cancel_ts IS NULL;
```

## Приложение В (обязательное)

### Код файла models.py

#### В.1 Модель МТР сертификата

```
class MtrCertificate(models.Model):
    database = 'sc_cmr'
    mc_mtr_code = models.IntegerField()
    mc_number_substring = models.IntegerField()
    mc_country_id_manufacturer = models.IntegerField(blank=True, null=True)
    mc_name_manufacturer_id = models.IntegerField(blank=True, null=True)
    mc_country_id = models.IntegerField(blank=True, null=True)
    mc_count_day_guaran = models.SmallIntegerField(blank=True, null=True)
    mc_warranty_period_year = models.SmallIntegerField(blank=True, null=True)
    mc_warranty_period_month = models.SmallIntegerField(blank=True, null=True)
    mc_count_pressing = models.SmallIntegerField(blank=True, null=True)
    mc_certificate_date = models.DateField(blank=True, null=True)
    mc_date_end_storage = models.DateField(blank=True, null=True)
    mc_date_manufacturing = models.DateField(blank=True, null=True)
    mc_date_recheck = models.DateField(blank=True, null=True)
    mc_create_ts = models.DateTimeField(blank=True, null=True)
    mc_presented_date_cert = models.DateTimeField(blank=True, null=True)
    mc_accepted_date_cert = models.DateTimeField(blank=True, null=True)
    mc_create_man_ts = models.DateTimeField(blank=True, null=True)
    mc_number_receipt_order = models.CharField(primary_key=True, max_length=20)
    mc_certificate_number = models.CharField(max_length=30, blank=True, null=True)
    mc_delivery_brand = models.CharField(max_length=50, blank=True, null=True)
    mc_delivery_assort = models.CharField(max_length=50, blank=True, null=True)
    mc_batch_number = models.CharField(max_length=20, blank=True, null=True)
    mc_melting_number = models.CharField(max_length=20, blank=True, null=True)
    mc_gost_chemical_compos = models.CharField(max_length=50, blank=True, null=True)
    mc_gost_assort = models.CharField(max_length=50, blank=True, null=True)
    mc_gost_tech_conditions = models.CharField(max_length=50, blank=True, null=True)
    mc_create_user = models.CharField(max_length=50, blank=True, null=True)
    mc_presented_user_cert = models.CharField(max_length=50, blank=True, null=True)
    mc_accepted_user_cert = models.CharField(max_length=50, blank=True, null=True)
    mc_number_folder_cert = models.CharField(max_length=20, blank=True, null=True)
    mc_conclusion_cert = models.CharField(max_length=500, blank=True, null=True)
    mc_article = models.CharField(max_length=50, blank=True, null=True)
    mc_variety = models.CharField(max_length=50, blank=True, null=True)
    mc_color = models.CharField(max_length=50, blank=True, null=True)
    mc_stamp = models.CharField(max_length=50, blank=True, null=True)
    mc_recipe = models.CharField(max_length=50, blank=True, null=True)
    mc_name_manufacturer = models.CharField(max_length=500, blank=True, null=True)
    mc_number_pressing = models.CharField(max_length=50, blank=True, null=True)
    mc_factory_number = models.CharField(max_length=50, blank=True, null=True)
    mc_allowance = models.CharField(max_length=500, blank=True, null=True)
    mc_type_cert = models.CharField(max_length=500, blank=True, null=True)
    mc_manufacturer = models.CharField(max_length=50, blank=True, null=True)
    mc_create_man_user = models.CharField(max_length=500, blank=True, null=True)
    mc_code_okvd = models.CharField(max_length=500, blank=True, null=True)
    class Meta:
        managed = False
        db_table = 'sc_cmr' + "." + 'mtr_certificate'
        unique_together = (('mc_number_receipt_order', 'mc_mtr_code',
            'mc_number_substring'),)
```

#### В.2 Модель движения МТР

```
class MtrMovingForStore(models.Model):
    database = 'sc_cmr'
```

```

pd_type_doc = models.CharField(max_length=50)
pd_doc_id = models.BigIntegerField(primary_key=True)
pd_doc_nom = models.CharField(max_length=20)
pd_doc_date = models.DateField()
pdr_row_id = models.BigIntegerField()
mtr_div_no_sender = models.SmallIntegerField()
mtr_keep_no_sender = models.SmallIntegerField()
mtr_keep_no_receiver = models.SmallIntegerField()
mtr_div_no_receiver = models.SmallIntegerField()
mtr_code = models.IntegerField()
mdm_additional_code = models.SmallIntegerField()
mtr_characteristic = models.CharField(max_length=500)
ut_code = models.SmallIntegerField()
mtr_cost = models.DecimalField(max_digits=19, decimal_places=4)
mtr_quantity = models.DecimalField(max_digits=18, decimal_places=7)
mtr_number_receipt_order = models.CharField(max_length=20)
mtr_number_substring = models.IntegerField()
pd_create_user = models.CharField(max_length=50)
pd_create_ts = models.DateTimeField()
dvs_confirmat_ts_1 = models.DateTimeField()
dvs_confirmat_user_1 = models.CharField(max_length=50)
dvs_number_registry = models.CharField(max_length=20)
pd_confirmat_ts = models.DateTimeField()
pd_confirmat_user = models.CharField(max_length=50)
class Meta:
    db_table = 'sc_cmr' + "." + 'mtr_moving_for_store'
    managed = False

```

## Приложение Г (обязательное)

### Код файла views.py

#### Г.1 Пагинация

```
class Pagination(PageNumberPagination):
    page_size = 400
    page_size_query_param = 'page_size'
    max_page_size = 1000
```

#### Г.2 МТР сертификат

```
class MtrCertificateList(generics.ListAPIView):
    queryset = MtrCertificate.objects.all().order_by('mc_mtr_code').values()
    serializer_class = MtrCertificateSerializer
    filter_backends = [filters.SearchFilter]
    pagination_class = Pagination

    def get_queryset(self):
        queryset = super().get_queryset()
        params = self.request.query_params
        filter_params = {}
        for param, value in params.items():
            filter_params[param] = value
        queryset = queryset.filter(**filter_params)
        return self.paginate_queryset(queryset)
```

#### Г.3 Движение на складах

```
class MtrMovingForStoreList(generics.ListAPIView):
    pagination_class = Pagination
    serializer_class = MtrMovingForStoreSerializer
    queryset = MtrMovingForStore.objects.all().order_by('pd_doc_date',
        'dvs_confirmat_ts_1', 'pd_doc_nom').values()

    def get_queryset(self):
        code = self.request.query_params.get('mtr_code')
        div_no = self.request.query_params.get('div_no')
        keep_no = self.request.query_params.get('keep_no')
        substring = self.request.query_params.get('mtr_number_substring')
        order = self.request.query_params.get('mtr_number_receipt_order')
        queryset = self.queryset.filter(mtr_code=code)

        if div_no and keep_no:
            queryset1 = Q(mtr_div_no_receiver=div_no) & Q(mtr_keep_no_receiver=keep_no)
            queryset2 = Q(mtr_div_no_sender=div_no) & Q(mtr_keep_no_sender=keep_no)
            queryset = queryset.filter(queryset1 | queryset2)

            if order and substring:
                queryset = queryset.filter(mtr_number_receipt_order=order,
                    mtr_number_substring=substring)
            else:
                queryset = queryset

        else:
            queryset = MtrMovingForStore.objects.none()
        return queryset
```

#### Г.4 Общее наличие МТР на складах

```

class Availability(generics.ListAPIView):
    pagination_class = Pagination
    queryset = AvailabilityForStore.objects.all().order_by('mtr_code').values()
    serializer_class = AvailabilityForStoreSerializer
    filter_backends = [filters.OrderingFilter]
    ordering_fields = '_all_'

    def get_queryset(self):
        div_no_receiver = self.request.query_params.get('mtr_div_no_receiver')
        keep_no_receiver = self.request.query_params.get('mtr_keep_no_receiver')
        check = self.request.query_params.get('mtr_quantity')

        if check:
            if div_no_receiver and keep_no_receiver:
                queryset =
AvailabilityForStore.objects.filter(mtr_div_no_receiver=div_no_receiver,
mtr_keep_no_receiver=keep_no_receiver, mtr_quantity__gt=check)
            else:
                queryset = AvailabilityForStore.objects.filter(mtr_quantity__gt=check)
        else:
            if div_no_receiver and keep_no_receiver:
                queryset =
AvailabilityForStore.objects.filter(mtr_div_no_receiver=div_no_receiver,
mtr_keep_no_receiver=keep_no_receiver)
            else:
                queryset =
AvailabilityForStore.objects.all().order_by('mtr_code').values()
        return queryset

```

## Г.5 Лимиты на складах

```

class AvailabilityLimit(generics.ListAPIView):
    queryset = AvailabilityLimit.objects.all().order_by('mtr_div_no_receiver').values()
    serializer_class = AvailabilityLimitSerializer
    pagination_class = Pagination

```

## Г.6 Соотношение подразделений и МОЛ

```

class DivNoKeepNoList(generics.ListAPIView):
    pagination_class = Pagination
    queryset = DivNoKeepNo.objects.all().order_by('div_no', 'keep_no').values()
    serializer_class = DivNoKeepNoSerializer
    filter_backends = [filters.OrderingFilter]
    ordering_fields = '_all_'

    def get_queryset(self):
        divno = self.request.query_params.get('div_no')
        # keepno = self.request.query_params.get('keep_no')
        if divno:
            queryset = DivNoKeepNo.objects.all().filter(div_no=divno).order_by('div_no',
'keep_no')
        else:
            queryset = DivNoKeepNo.objects.all().order_by('div_no', 'keep_no')
        return queryset

```

## Приложение Д (обязательное)

### Код Django шаблонов

#### Д.1 Содержание файла родительского шаблона

```
<!DOCTYPE html>
{% load static %}
<html>
<head>
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}" />
    <link rel="stylesheet" href="{% static 'css/bootstrap.min.css.map' %}" />
    <link rel="stylesheet" href="{% static 'css/tableColl.css' %}" />
    <script src="{% static 'js/bootstrap.bundle.min.js' %}"></script>
<link rel="icon" href="data:; base64,=">
<title>{% block title %} MTP {% endblock title %}</title>
{% block menu %}
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
            <a class="navbar-brand" href="#">Панель навигации</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNavDarkDropdown"
            aria-controls="navbarNavDarkDropdown" aria-expanded="false" aria-
label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse">
                <ul class="navbar-nav">
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#"
id="navbarDrkDropdownMenuLink" role="button" data-bs-toggle="dropdown"
                        aria-expanded="false">Меню</a>
                        <ul class="dropdown-menu dropdown-menu-dark" aria-
labelledby="navbarDrkDropdownMenuLink">
                            <li><a class="dropdown-item"
href="http://127.0.0.1:8000/api/home/">Главная страница</a></li>
                            <li><a class="dropdown-item"
href="http://127.0.0.1:8000/api/availab/">Наличие</a></li>
                            <li><a class="dropdown-item"
href="http://127.0.0.1:8000/api/motionless/">Без движения</a></li>
                            <li><a class="dropdown-item"
href="http://127.0.0.1:8000/api/limits/">Финансовый анализ</a></li>
                        </ul>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
<p></p>
{% endblock %}
</head>
{%block namePage%}{% endblock %}

{%block cont%}{% endblock%}

{%block search%}{% endblock%}

{%block navig%}{% endblock%}

{%block contmain%}
<div class="container-fluid">
    <div class="row">
```

```

        <div class="col-md-6">
            {% block col1 %}{% endblock %}
        </div>
        <div class="col-md-6">
            {% block col2 %}{% endblock %}
        </div>
    </div>
    <br>
    <div class="row">
        {% block certcol %} {% endblock %}
    </div>
</div>
{% endblock%}
</html>

```

## Д.2 Содержимое файла дочернего шаблона PageAvailab.html

```

{% extends "html/PageMain.html" %}

{% load static %}
<script type="text/javascript" src="{% static 'js/MainPage.js' %}"></script>
<head>
    {% block title %}Наличие{% endblock title %}
</head>

{%block namePage%}
<h3 align="center">Наличие мтр по <label id="war_mol"></label> на <label
id="currDate"></label></h3>
{% endblock %}

{%block navig%}
    {% include "navig.html" %}
    <script type="text/javascript" src="{% static 'js/MainPage.js' %}"></script>
{% endblock%}

{%block cont%}
<form>
    <label>Номер подразделения: </label>
    <select id="divNo"><option value=""></option></select>
    <label>МОЛ: </label>
    <select id="keepNo"><option value=""></option></select>
    <label>Количество больше 0 </label>
    <input id="quan" type="checkbox" checked>
    <input type="text" id="searchTab" oninput="searchTable()" placeholder="Поиск по МТР
коду...">
    <button type="button" onclick="resetPagination()">Поиск</button>
</form>
<p></p>
{% endblock cont%}

{% block col2 %}{% include "tablesMoving.html"%}{% endblock %}

{% block col1 %}{% include "tablesAvailab.html"%}{% endblock %}

{% block certcol%}{% include "CertMtr.html"%}{% endblock %}

```

## Д.3 Содержимое файла дочернего шаблона PageMotionless.html

```

{% extends "html/PageMain.html" %}

{% load static %}
<script type="text/javascript" src="{% static 'js/MotionlessPage.js' %}"></script>

```



```

<link rel="stylesheet" href="{% static 'css/tableCol1.css' %}"/>
<head>
    {% block title %}Без движения{% endblock title %}
</head>
{%block navig%}{% include "navig.html" %}{% endblock%}

{%block namePage%}
<h3 align="center">Материалы без движения по <label id="war_mol"></label> более <label
id="day"></label> дней на <label id="currDate"></label></h3>
{% endblock %}

{%block cont%}
<form>
    <label>Количество дней: </label>
    <select id="daysSelect">
        <option value="90">90</option>
        <option value="60">60</option>
        <option value="30">30</option>
    </select>
    <label>Номер подразделения: </label>
    <select id="divNo"><option value=""></option></select>
    <label>МОЛ: </label>
    <select id="keepNo">
        <option value=""></option>
    </select>
    <input type="text" id="searchTab" oninput="searchTable()" placeholder="Поиск по
МТР коду...">
    <button type="button" onclick="resetPagination()">Поиск</button>
</form>
<p></p>
{% endblock%}

{% block col2 %}{% include "tablesMoving.html"%}{% endblock %}

{% block col1 %}{% include "tablesAvailab.html"%}{% endblock %}

{% block certcol%}{% include "CertMtr.html"%}{% endblock %}

```

#### Д.4 Содержимое файла tablesMoving.html

```

{% load static %}
<link rel="stylesheet" href="{% static 'css/tableCol1.css' %}"/>
<label>Движение</label>
<div class="table-container">
    <table class="table table-bordered table-hover table-sm caption-top col-lg-6"
id="tableMoving">
        <thead class="table-light" >
            <tr id="headerRowMoving"></tr>
        </thead>
        <tbody id="bodyMoving"></tbody>
    </table>
</div>

```

#### Д.5 Содержимое файла tablesAvailab.html

```

{% load static %}

<link rel="stylesheet" href="{% static 'css/tableCol1.css' %}"/>

```

```

<label>Наличие</label>
<div class="table-container">
    <table class="table table-bordered table-hover table-sm caption-top col-lg-6
table-condensed" id="tableAvail">
        <thead class="table-light" >
            <tr id="headerRowAvail"></tr>
        </thead>
        <tbody id="bodyAvail"></tbody>
    </table>
</div>

```

## Д.6 Содержимое файла navig.html

```

{% load static %}
<nav>
    <ul class="pagination" id="pagination">
        <li class="page-item" id="prev-page">
            <a class="page-link" href="#" aria-label="Previous">
                <span aria-hidden="true">&laquo;</span>
            </a>
        </li>
        <li class="page-item" id="next-page">
            <a class="page-link" href="#" aria-label="Next">
                <span aria-hidden="true">&raquo;</span>
            </a>
        </li>
    </ul>
</nav>

```

## Приложение Е (обязательное)

### Код клиентских частей веб-сервиса

#### Е.1 Реализация функционала для отображения номеров подразделений и номеров МОЛ в combobox на языке JavaScript

```
const DivDropdown = document.getElementById('divNo');
const KeepDropdown = document.getElementById('keepNo');
removeDupl(DivDropdown);
removeDupl(KeepDropdown);
fetch('http://127.0.0.1:8000/api/divno/')
  .then(resp => resp.json())
  .then(data => {
    data.results.forEach(div =>{
      const option = document.createElement('option');
      option.text = div.div_no;
      option.value = div.div_no;
      DivDropdown.add(option)
    }); removeDupl(DivDropdown);
  });
DivDropdown.addEventListener('change', (event) => {
  const selectedDiv = event.target.value;

  fetch('http://127.0.0.1:8000/api/divno/?div_no='+selectedDiv)
    .then(resp => resp.json())
    .then(data => {
      KeepDropdown.innerHTML = '';
      data.results.forEach(keep =>{
        const option = document.createElement('option');
        option.text = keep.keep_no;
        option.value = keep.keep_no;
        KeepDropdown.add(option);
      });removeDupl(KeepDropdown);
    });KeepDropdown.innerHTML = '';
  });

function removeDupl(selectElement){
  let valuesFound = {};
  let index = 0;

  while (index<selectElement.options.length){
    let currentValue = selectElement.options[index].text;
    if (valuesFound[currentValue]) {
      selectElement.remove(index);
    } else {
      valuesFound[currentValue] = true;
      index++;
    }
  }
}
```

#### Е.2 Код функционала для вывода данных в таблицу «Наличие» на языке JavaScript

```
function AvailabilityDays(page, days, div, keep){
```

```

days = document.getElementById("daysSelect").value;
day.innerHTML = document.getElementById("daysSelect").value;

if (document.getElementById("divNo").value !== '') {
    war_mol.innerHTML = document.getElementById("divNo").value + '/' +
document.getElementById('keepNo').value;
}
else {
    war_mol.innerHTML='складу/мол'
}
currDate.innerHTML = new Date().toLocaleDateString();

div = document.getElementById('divNo').value;
keep = document.getElementById('keepNo').value;
console.log(div)
if (div===''){
    var url = 'http://'+url_list[2]+'api/availabilitydays/?page='+page +
'&days_ago='+days
} else
    {var url = 'http://'+url_list[2]+'api/availabilitydays/?page='+page +
'&days_ago='+days + '&mtr_div_no_receiver='+div + '&mtr_keep_no_receiver='+keep}
    console.log(url)
    fetch(url)
        .then(response=>response.json())
        .then(data=>{
            totalItems = data.count;
            console.log(totalItems);
            const table = document.getElementById('tableAvail');
            const headerRow = document.getElementById('headerRowAvail');
            const body = document.getElementById('bodyAvail');
            body.innerHTML = '';
            headerRow.innerHTML = '';
            // const columnHeaders = ['Код мтр', 'Характеристика','Склад/мол', 'Количество',
            'Ед.измерения', 'Стоимость,руб',
            // 'Номер приходного ордена', 'Номер подстроки', 'Кол-во дней', '']
            const columnHeaders = ['Склад/мол', 'Код мтр', 'Ед.измерения',
            'Характеристика', 'Количество', 'Стоимость,руб',
            'Номер приходного ордена', 'Номер подстроки', 'Кол-во дней']
            // const totalPages = Math.ceil(data.length / 400);
            columnHeaders.forEach((header, index) =>{
                const th = document.createElement('th');
                th.textContent = header;
                th.dataset.columnIndex = index;
                th.addEventListener('click', ()=>{
                    sortTable(index);
                })
                headerRow.appendChild(th);
            });

            data.results.forEach(rowData =>{
                const row = document.createElement('tr');
                Object.values(rowData).forEach((value, index) =>{
                    if (index !==0 && index !==1 && index !==3 && index !==4 && index !==5 &&
index !==6 && index !==7 && index !==8) {
                        const td = document.createElement('td');
                        td.textContent = value !==null ? value : '';
                        row.appendChild(td)
                    } else if (index ===0 ){
                        const td = document.createElement('td');

```

```

        const combinedTd = (Object.values(rowData)[0] !==null ?
Object.values(rowData)[0] : '') +
        '/' +(Object.values(rowData)[1] !==null ? Object.values(rowData)[1] :
'');

        td.textContent = combinedTd.trim();
        row.appendChild(td)
    } else if (index ===2){
        const td = document.createElement('td');
        if (Object.values(rowData)[2].length === 7){
//            formattedValue = '0'+Object.values(rowData)[0];
            td.textContent = Object.values(rowData)[2]+'0000';
        } else {
            td.textContent = Object.values(rowData)[2]
        }
        row.appendChild(td)
    }
    else if (index ===3 ){
        const td = document.createElement('td');
        const combinedTd = (Object.values(rowData)[3] !==null ?
Object.values(rowData)[3] : '') +
        '(' +(Object.values(rowData)[4] !==null ? Object.values(rowData)[4] :
'')+')';

        td.textContent = combinedTd.trim();
        row.appendChild(td)
    }
    else if (index ===5 ){
        const td = document.createElement('td');
        const combinedTd = (Object.values(rowData)[5] !==null ?
Object.values(rowData)[5] : '') +
        ' ' +(Object.values(rowData)[6] !==null ? Object.values(rowData)[6] :
'') ;

        td.textContent = combinedTd.trim();
        row.appendChild(td)
    }
    else if ((index ===8) || (index ===7)){
        const td = document.createElement('td');
        td.textContent = value !==null ? value : '';
        td.style.textAlign = 'right';
        row.appendChild(td);
    }
});

    row.addEventListener('click', function(){
        const allRows = document.querySelectorAll('tr');
        allRows.forEach(row=>{
            row.style.background = '';
        });
        const selectRow = this.closest('tr');
        selectRow.style.background = '#808080';
//        MovingMtr('\'+Object.values(rowData)[2]+'\'',
        '\'+Object.values(rowData)[9]+'\'',
        '\'+Object.values(rowData)[10]+'\'',\'+Object.values(rowData)[0]+'\'',
        '\'+Object.values(rowData)[1]+'\'' );
        MovingMtr(Object.values(rowData)[2], Object.values(rowData)[9],
Object.values(rowData)[10], Object.values(rowData)[0], Object.values(rowData)[1]);
    })

    body.appendChild(row);

```

```

    });
    columnHeaders.innerHTML='';

    currentPages = page;
    UpdatePagination();

  })
  .catch(error =>console.log(error));
  const body = document.getElementById('bodyMoving');
  body.innerHTML='';
}

```

### Е.3 Код функционала вывода данных в таблицу «Движение» на языке JavaScript

```

function MovingMtr(paramCode, paramDiv, paramKeep){
  const labelCert = document.querySelectorAll('.certLabel');
  labelCert.forEach(row=>{
    row.innerHTML='';
  });
  var url = 'http://'+url_list[2]+'api/mtr-moving-for-store/?mtr_code='+
  paramCode+'&div_no='+paramDiv+'&keep_no='+paramKeep;
  console.log(url);
  fetch(url)
    .then(response=>response.json())
    .then(data=>{
      const table = document.getElementById('tableMoving');
      const headerRow = document.getElementById('headerRowMoving');
      const body = document.getElementById('bodyMoving');
      body.innerHTML = '';
      headerRow.innerHTML = '';
      const columnHeaders = ['Тип', 'Номер', 'Дата создания', 'Отправитель',
      'Получатель', 'Стоимость', 'Количество приход', 'Количество расход',
      'Номер приходного ордена', 'Номер подстроки', '']
      columnHeaders.forEach((header, index) =>{
        const th = document.createElement('th');
        th.textContent = header;
        th.dataset.columnIndex = index;
        th.addEventListener('click', ()=>{
          sortTable(index);
        })
        headerRow.appendChild(th);
      });

      data.results.forEach(rowData =>{
        const row = document.createElement('tr');

        const pdTypeDocCell=document.createElement('td');
        pdTypeDocCell.textContent=rowData.pd_type_doc !==null ? rowData.pd_type_doc :
        '';

        row.appendChild(pdTypeDocCell);

        const pdDocNomCell=document.createElement('td');
        pdDocNomCell.textContent=rowData.pd_doc_nom !==null ? rowData.pd_doc_nom :
        '';

        row.appendChild(pdDocNomCell);

        const pdDocDateCell=document.createElement('td');

```

```

pdDocDateCell.textContent=rowData.pd_doc_date !==null ? rowData.pd_doc_date :
'';

row.appendChild(pdDocDateCell);

const mtrDivNoSenderCell=document.createElement('td');
if (rowData.mtr_div_no_sender !== '') {
    mtrDivNoSenderCell.textContent=(rowData.mtr_div_no_sender)+
    '/' +(rowData.mtr_keep_no_sender !==null ? rowData.mtr_keep_no_sender : '');
}
else if (rowData.mtr_div_no_sender == '') {
    mtrDivNoSenderCell.textContent='';
}
row.appendChild(mtrDivNoSenderCell);

const mtrKeepNoReceiverCell=document.createElement('td');
if (rowData.mtr_div_no_receiver !== '') {
    mtrKeepNoReceiverCell.textContent=(rowData.mtr_div_no_receiver)+
    '/' +(rowData.mtr_keep_no_receiver !==null ? rowData.mtr_keep_no_receiver :
'');
}
else if (rowData.mtr_div_no_receiver == '') {
    mtrKeepNoReceiverCell.textContent='';
}
row.appendChild(mtrKeepNoReceiverCell);

const mtrCodeCell=document.createElement('td');
mtrCodeCell.textContent=rowData.mtr_code;
mtrCodeCell.style="display:none";
row.appendChild(mtrCodeCell);

const mtrCostCell=document.createElement('td');
mtrCostCell.textContent=rowData.mtr_cost;
mtrCostCell.style.textAlign = 'right'
row.appendChild(mtrCostCell);

const mtrQuantitySenCell=document.createElement('td');
const mtrQuantityRecCell=document.createElement('td');

if (rowData.mtr_div_no_sender == paramDiv) {
    mtrQuantitySenCell.textContent='';
    mtrQuantityRecCell.textContent=rowData.mtr_quantity;
    mtrQuantityRecCell.style.textAlign = 'right'
}

if (rowData.mtr_div_no_receiver == paramDiv){
    mtrQuantitySenCell.textContent=rowData.mtr_quantity;
    mtrQuantitySenCell.style.textAlign = 'right'
    mtrQuantityRecCell.textContent = '';
}
row.appendChild(mtrQuantitySenCell);
row.appendChild(mtrQuantityRecCell);

const mtrNumberReceiptOrderCell=document.createElement('td');
mtrNumberReceiptOrderCell.textContent=rowData.mtr_number_receipt_order;
row.appendChild(mtrNumberReceiptOrderCell);

const mtrNumberSubstringCell=document.createElement('td');
mtrNumberSubstringCell.textContent=rowData.mtr_number_substring;
row.appendChild(mtrNumberSubstringCell);

```

```

        row.addEventListener('click', function(){
            const allRows = document.querySelectorAll('#bodyMoving tr');

            allRows.forEach(row=>{
                row.style.background = '';
            });
            const selectRow = this.closest('tr');
            selectRow.style.background = '#808080';
            MtrCert(mtrCodeCell.innerText, mtrNumberSubstringCell.innerText,
mtrNumberReceiptOrderCell.innerText);
        })

        body.appendChild(row)
    });
    columnHeaders.innerHTML='';
    currentPage = page;
    UpdatePagination();
})
.catch(error =>console.log(error));
}

```

#### Е.4 Код функционала для вывода данных в блок МТР сертификата

```

function MtrCert(paramMtrCode, paramNumberSubstring,paramNumberReceiptOrder){
    const url='http://'+url_list[2]+'api/mtr-
certificate/?&mc_number_receipt_order='+paramNumberReceiptOrder +
'&mc_mtr_code='+paramMtrCode+'&mc_number_substring='+paramNumberSubstring;
    console.log(url)
    fetch(url)
    .then(resp=>resp.json())
    .then(data=>{
        console.log(data);
        createMtrCertCont(data);
    })
    .catch(error=>console.log(error))
}

function createMtrCertCont(data){
    const labelCert = document.querySelectorAll('.certLabel');
    labelCert.forEach(row=>{
        row.innerHTML='';
    });
    var MtrCertCont = Object.values(data.results);
    MtrCertCont.forEach(rowData=>{
        document.getElementById('mtr').innerHTML=rowData.mc_mtr_code;
        document.getElementById('nro').textContent=rowData.mc_number_receipt_order;
        document.getElementById('ns').textContent=rowData.mc_number_substring;
        document.getElementById('cd').textContent=rowData.mc_certificate_date;
        document.getElementById('cn').textContent=rowData.mc_certificate_number;
        document.getElementById('wpm').textContent=rowData.mc_warranty_period_month;
        document.getElementById('wpy').textContent=rowData.mc_warranty_period_year;
        document.getElementById('cdg').textContent=rowData.mc_count_day_guaran;
        document.getElementById('des').textContent=rowData.mc_date_end_storage;

        document.getElementById('cp').textContent=rowData.mc_count_pressing;
        document.getElementById('np').textContent=rowData.mc_number_pressing;
        document.getElementById('db').textContent=rowData.mc_delivery_brand;
        document.getElementById('da').textContent=rowData.mc_delivery_assort;
        document.getElementById('bn').textContent=rowData.mc_batch_number;
        document.getElementById('mn').textContent=rowData.mc_melting_number;
    });
}

```



```

document.getElementById('gcc').textContent=rowData.mc_gost_chemical_compos;
document.getElementById('ga').textContent=rowData.mc_gost_assort;
document.getElementById('gtc').textContent=rowData.mc_gost_tech_conditions;

document.getElementById('dm').textContent=rowData.mc_date_manufacturing;
document.getElementById('m').textContent=rowData.mc_manufacturer;
document.getElementById('dr').textContent=rowData.mc_date_recheck;
document.getElementById('pdc').textContent=rowData.mc_presented_date_cert;
document.getElementById('puc').textContent=rowData.mc_presented_user_cert;
document.getElementById('adc').textContent=rowData.mc_accepted_date_cert;
document.getElementById('auc').textContent=rowData.mc_accepted_user_cert;
document.getElementById('nm').textContent=rowData.mc_name_manufacturer;
document.getElementById('fn').textContent=rowData.mc_factory_number;

document.getElementById('nfc').textContent=rowData.mc_number_folder_cert;
document.getElementById('a').textContent=rowData.mc_article;
document.getElementById('v').textContent=rowData.mc_variety;
document.getElementById('c').textContent=rowData.mc_color;
document.getElementById('s').textContent=rowData.mc_stamp;
document.getElementById('r').textContent=rowData.mc_recipe;

document.getElementById('al').textContent=rowData.mc_allowance;
document.getElementById('cc').textContent=rowData.mc_conclusion_cert;
document.getElementById('co').textContent=rowData.mc_code_okvd;
document.getElementById('tc').textContent=rowData.mc_type_cert;
});
}

```

## Е.5 Код функционала вывода данных в таблицу с лимитами на языке JavaScript

```

function AvailabilityLimits(){
    fetch('http://'+url_list[2]+'api/availability-limit/')
    .then(resp=>resp.json())
    .then(data=>{
        console.log(data);
        const table = document.getElementById('tableAvail');
        const headerRow = document.getElementById('headerRowAvail');
        const body = document.getElementById('bodyAvail');
        currDate.innerHTML = new Date().toLocaleDateString();

        const columnHeaders = ['', 'Подразделение', 'Лимит,руб', 'Фактическая
стоимость,руб'];

        columnHeaders.forEach((header, index)=>{
            const th = document.createElement('th');
            th.textContent = header;
            // th.dataset.columnIndex = index;
            headerRow.appendChild(th);
        });

        data.results.forEach(rowData =>{
            const row = document.createElement('tr');

            const planCell=document.createElement('td');

            const par1 = parseFloat((rowData.rate).replace(/\s/g, '').replace(',',''))
            const par2 = parseFloat((rowData.mtr_cost).replace(/\s/g, '').replace(',','')
            ''))

```

```

        diff = par1-par2;

        if (diff>0){
            planCell.innerHTML='&#8730';
        }
        if (diff<0){
            planCell.innerHTML='&#x2716';
        }
        console.log(planCell.innerText);
        row.appendChild(planCell);

        const divNoCell=document.createElement('td');
        divNoCell.textContent=rowData.mtr_div_no_receiver;
        row.appendChild(divNoCell);

        const rateCell=document.createElement('td');
        rateCell.textContent=rowData.rate;
//        rateCell.textContent = '{:,.2f}'.format(rowData.rate).replace(',',' ');
        rateCell.style.textAlign = 'right'
        row.appendChild(rateCell);

        const costCell=document.createElement('td');
        costCell.textContent=rowData.mtr_cost;
        costCell.style.textAlign = 'right'
        row.appendChild(costCell);

        body.appendChild(row)
    });
}

```

## Е.6 Стили для таблиц

```

.container-fluid{
    position: relative;
}
.cont{
    border: 1px solid #ccc;
    max-height: 100vh;
    height: 250px;
    overflow-y: auto;
    position: relative;
    transition: height 0.3s;
}
.toggle-btn{
    position: absolute;
    top: 0;
    right:0;
    cursor: pointer;
    font-size: 35px;
}
.col-md-6{
    font-size: 15px;
    color: #333333;
}
.col-md-2{
    font-size: 20px;
    color: #333333;
}

```

```
.mtrlab{  
  font-size: 30px;  
  color: #333333;  
}
```

## Приложение Ж (обязательное)

### Код реализации пагинации

```
function UpdatePagination(){
    const pagination =document.getElementById('pagination');
    pagination.innerHTML='';
    totalPages = Math.ceil(totalItems/400);
    maxPageShow = Math.min(totalPages, 10);
    const prevBtn = document.createElement('li');
    prevBtn.classList.add('page-item');
    prevBtn.innerHTML = `<<span aria-
hidden="true">&laquo</span></a>`;
    prevBtn.addEventListener\('click', \(\)=>{
        if \(currentPages>1\){
            AvailabilityDays\(currentPage-1\);
        }
    }\);
    pagination.appendChild\(prevBtn\);

    let startPage = Math.max\(1, currentPages - Math.floor\(maxPageShow/2\)\);
    let endPage = Math.min\(totalPages, startPage+maxPageShow-1\);

    for \(let i=startPage; i<=endPage; i++\){
        const pageBtn = document.createElement\('li'\);
        pageBtn.classList.add\('page-item'\);
        pageBtn.innerHTML = `{
            AvailabilityDays\\(i\\);
        }\\);
        if \\(i===currentPages\\){
            pageBtn.classList.add\\('active'\\);
        }
        pagination.appendChild\\(pageBtn\\);
    }

    const nextBtn = document.createElement\\('li'\\);
    nextBtn.classList.add\\('page-item'\\);
    nextBtn.innerHTML = `>>span aria-
hidden="true">&raquo</span></a>`;
    nextBtn.addEventListener\\\('click', \\\(\\\)=>{
        if \\\(currentPages<totalPages\\\){
            AvailabilityDays\\\(currentPages+1\\\);
        }
    }\\\);
    pagination.appendChild\\\(nextBtn\\\);
    document.querySelector\\\('.table-container'\\\).scrollTop=0;
}
```