МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ф Е Д Е Р А Л Ь Н О Е Г О С У Д А Р С Т В Е Н Н О Е А В Т О Н О М Н О Е О Б Р А З О В А Т Е Л Ь Н О Е У Ч Р Е Ж Д Е Н И Е В Ы С Ш Е Г О О Б Р А З О В А Н И Я

« Н а ц и о н а л ь н ы й и с с л е д о в а т е л ь с к и й я д е р н ы й у н и в е р с и т е т « М И Ф И »

**Технологический институт –**

филиал федерального государственного автономного образовательного учреждения высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

**(ТИ НИЯУ МИФИ)**

**Кафедра информационных технологий и прикладной математики**

В Ы П У С К Н А Я К В А Л И Ф И К А Ц И О Н Н А Я Р А Б О Т А Б А К А Л А В Р А

направление подготовки 09.03.01 «Информатика и вычислительная техника»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**Тема:**  Разработка программного модуля для сопровождения ведения журнала учета разрешений замены материала в

ИС МКАДД

**Руководитель**

Инженер

программист Д.С.Лебедев

*(должность) (подпись) (И.О. Фамилия)*

**Студент**

ИВТ-41Д М.Е.Химач

*(группа) (подпись) (И.О. Фамилия)*

**ВКР допущена к защите в ГЭК**

**Заведующий кафедрой**  Н.В. Чупракова

«» июня 2025 г.

*ВКР защищена «» июня 2025 г.*

*Протокол ГЭК №* *на оценку «* *»*

*Секретарь ГЭК* */Ю.А. Порохина /*

г. Лесной – 2025 г

СОДЕРЖАНИЕ

[ТЕРМИНЫ И СОКРАЩЕНИЯ 3](#_TOC_250014)

[ВВЕДЕНИЕ 4](#_TOC_250013)

[ГЛАВА 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ 7](#_TOC_250012)

* 1. [Анализ предметной области 7](#_TOC_250011)
  2. [Описание бизнес-процесса 10](#_TOC_250010)
  3. [Средства разработки программного обеспечения 12](#_TOC_250009)

[ГЛАВА 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ 17](#_TOC_250008)

* 1. [Построение информационной модели «сущность-связь» 17](#_TOC_250007)
  2. [Проектирование базы данных 20](#_TOC_250006)
  3. [Проектирование клиентской части 31](#_TOC_250005)

[ГЛАВА 3 ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ 37](#_TOC_250004)

* 1. Руководство пользователя. Тестирование приложения 37
  2. [Планируемый эффект от внедрения 46](#_TOC_250003)

[ЗАКЛЮЧЕНИЕ 49](#_TOC_250002)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 50](#_TOC_250001)

Приложение А 53

[Приложение Б 54](#_TOC_250000)

Приложение В 55

Приложение Г 62

Приложение Д 89

# ТЕРМИНЫ И СОКРАЩЕНИЯ

* **ВМ —** ведомость материала;
* **ДСЕ** — деталь, сборочная единица;
* **ИС** — информационная система;
* **ЛЗК** — лимитно-заборная карта — первичный документ, предназначенный для отпуска материалов, систематически потребляемых для изготовления продукции, а также для контроля за соблюдением лимитов;
* **Маршрут изготовления** или **маршрут** — это последовательность следования подразделений, участвующих в изготовлении ДСЕ;
* **МТР** — материально-технический ресурс , используемый предприятием в производстве продукции;
* **ПДГ** — производственно-диспетчерская группа (производственное подразделение);
* **ПИ —** предварительное извещение;
* **СП** — сопроводительный паспорт;
* **Склад** — специально выделенная территория, предназначенная для сохранения и хранения товаров и материалов. Основная функция склада заключается в собирании и хранении запасов, обеспечивая непрерывные и систематизированные поставки материалов и компонентов для удовлетворения производственных потребностей;
* **ТП** — технологический процесс;
* **Цех —** основное производственное подразделение промышленного предприятия;

# ВВЕДЕНИЕ

В современном обществе цифровизация и автоматизация бизнес- процессов занимают одно из главных мест в стратегии развития практически всех отраслей экономики, позволяя повысить эффективность, снизить затраты и минимизировать риски, связанные с человеческим фактором. Цифровые технологии приобретают особое значение в условиях увеличения объемов данных и усложнения производственных процессов. В таких высокотехнологичных и ответственных областях, как ядерный оборонный комплекс, где каждая операция чрезвычайно важна, внедрение информационных систем становится актуальным. Современные технологии не только упрощают работу специалистов, но и гарантируют соблюдение строгих нормативных требований и соответствие стандартам качества и безопасности. Важным аспектом является импортонезависимость на предприятии, поскольку независимые от внешних источников решения способствуют большому контролю и устойчивости в работе, что особенно актуально в условиях глобальных вызовов.

Одним из важнейших направлений цифровой трансформации является автоматизациях систем учета и мониторинга. Эти системы позволяют не только эффективно управлять производственными процессами, но и уменьшать количество ошибок при работе с данными. Ведение журнала учета замены материалов на производстве — одна из таких задач, требующих повышенного внимания. Ошибки или несоответствия в этом процессе могут привести к ряду последствий: простоям из-за отсутствия нужных материалов, увеличению затрат на дополнительные расходы срочных покупок или даже нарушениям безопасности.

При проектировании современных информационных систем необходимо учитывать следующие факторы [1]:

* Доступность данных: данные должны быть доступны уполномоченным сотрудникам в любой момент времени из любого места;
* Точность: вводимые данные должны быть защищены от случайной ошибки пользователя методами автоматизированного контроля;
* Безопасность: при работе с конфиденциальной информацией важно, чтобы данные были защищены от несанкционированного доступа, модификации или потери;
* Удобство использования: интерфейс системы должен быть удобным для пользователя, чтобы сотрудники могли эффективно выполнять свою работу и не нуждались в дополнительном обучении;
* Систематизация и структурирование данных: именно здесь автоматизация позволяет избежать дублирования, несогласованности и потери данных;

Информационные системы, устанавливаемые в такой отрасли, как ядерный оборонный комплекс, должны отвечать самым строгим требованиям. Они являются не только инструментом управления, но и средством обеспечения безопасности. Такие системы становятся основой для повышения эффективности производства, поскольку упрощают доступ к основной информации и ускоряют принятие решений.

Создание информационного модуля для учета разрешений на замену материалов в рамках информационной системы МКАДД отражает реальную потребность на уровне предприятия в инструменте, который позволит стать более импортонезависимыми.

Объектом исследования является процесс учета и управления разрешениями на замену материалов в рамках эксплуатации информационной системы МКАДД на предприятии «ФГУП Комбинат «Электрохимприбор».

Предметом исследования являются методы и подходы к автоматизации процессов учета разрешений на замену материалов с использованием веб- технологий.

Целью данной работы является разработка программного модуля, интегрированного с ИС МКАДД, который позволит автоматизировать учет разрешений на замену материалов, обеспечивая импортонезависимость,

точность данных и возможность анализа информации.

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. 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. Провести анализ существующих подходов к ведению журнала учета разрешений на замену материалов и выявить требования к системе;
2. Изучить существующую на предприятии систему для дальнейшей интеграции и составить модель бизнес-процесса;
3. Спроектировать функции для проекта в предоставленной базе данных;
4. Разработать серверную часть приложения;
5. Разработать клиентскую часть приложения;
6. Протестировать программный продукт;
7. Составить руководство пользователя;
8. Проанализировать эффект от внедрения разработки;

# ГЛАВА 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ

## Анализ предметной области

ФГУП «Комбинат «Электрохимприбор» - ключевое предприятие Росатома, специализирующиеся на производстве специзделий, гражданской продукций, оборудование для нефтегазовой отрасли, электроэнергетики и медицинских изотопов. Комбинат является единственным предприятием в мире, которое может осуществлять промышленное производство определенных изотопов как для внутреннего, так и для международных рынков. Он также располагает уникальными запатентованными технологиями позволяющими получать изотопы.

На предприятии основным документом, сопровождающим партию ДСЕ в течение всего жизненного цикла изготовления продукции на комбинате, является СП, который обеспечивает контроль последовательности выполнения операций по маршруту, установленному ТП. СП содержит информацию о продукции и её характеристиках:

* + - наименование;
    - обозначение;
    - типоразмер;
    - количестве ДСЕ в партии;
    - материале продукции;
    - последовательности выполнения операций (маршруте).

Номенклатура представляет собой систематизированный перечень деталей, сборочных единиц (ДСЕ) и материалов, используемых в производственном процессе. Она служит основой для учета, планирования и контроля учета замены материла, обеспечивая четкую идентификацию каждой единицы на всех этапах изготовления.

Маршрут представляет собой последовательность операций,

определяющую перемещение деталей сборочных единиц (ДСЕ) между производственными цехами. В данном контексте маршрут представляет собой

последовательность цехов (цехозаходов), через которые проходит ДСЕ в ходе производственного процесса. Каждый этап цехозахода предполагает выполнение операции отдельного производственного действия, направленного на обработку деталей сборочных единиц (ДСЕ).

Сменное задание представляет собой оперативный документ, регламентирующий конкретные задачи по сборке или обработки ДСЕ на определенную рабочую смену.

После получения СП в цех, мастер получает необходимый материал, после ознакамливается с СП и на основании него формирует сменное задание.

При отсутствии на складе материала, указанного в СП производится замена на другой, имеющий аналогичные свойства и характеристики. В этом случае, после согласования с технологом, ПДГ делает разрешающие записи в журнале замены материала в ИС и далее склад выдает необходимый материал[2].

Таким образом предметная область включает в себя объекты: сопроводительный паспорт, пользователь, маршрут изготовления, номенклатура, сменное задание, журнал замены материала, операция.

Объекты включают в себя следующие параметры.

* Паспорт: номер паспорта, год, цех, номер заказа, количество ДСЕ в партии, материалы, особые замечания, ДСЕ (объект Номенклатура), маршрут (объект Маршрут изготовления);
* Пользователь: табельный номер, фамилия, имя, отчество, логин;
* Маршрут изготовления: перечень цехозаходов (объект Цехозаход);
* Цехозаход: номер цеха, порядковый номер в маршруте, список операций (объект Операция);
* Операция: порядковый номер операции, название операции, код операции;
* Номенклатура: название ДСЕ, обозначение ДСЕ, размерные параметры ДСЕ, код;
* Сменное задание: номер задания, дата выдачи, дата приёма, плановая

дата окончания, список операций, ДСЕ (объект Номенклатура) , мастер (объект Пользователь), рабочий (объект Пользователь);

* Журнал замены материала: номер цеха, материал, дата заведения, дата изменения, документ разрешающий запуск материала, примечание ТП, примечание ПИ, номер разрешения, ответственный за паспорт (объект Пользователь), ответственный за материал (объект Пользователь), сопроводительный паспорт (объект Паспорт);

Задача ведения журнала учета разрешения замены материала в ИС МКАДД является важной и ответственной функцией в обеспечении качества выпускаемой продукции, система должна обеспечить легкость в использовании, высокую точность данных и возможность анализа информации, а также решить следующие задачи:

* + Организация учета замены материала: контроль замен материалов, обеспечение точности и актуальности данных;
  + Аналитика : Процентное соотношение паспортов с заменой материала к общему количеству СП;
  + Регистрация и редактирование журнала: введение записей о всех запросов на замену материала и возможность редактирования;

На момент разработки в корпоративной сети комбината обрабатывается информация, данные которой хранятся на SQL-сервере корпоративной сети, который находится в Управления информационных технологий и связи. Выгрузка и загрузка информации на сервер корпоративной сети производится с рабочей станции, подключенной к ней. В целях перехода на импортонезаваисимое ПО используется Postgres Pro – это отечественная объектно-реляционная система управления базами данных (СУБД), разработанная на основе PostgreSQL[16]. Она включает в себя дополнительные оптимизации, адаптированные под российские требования, и является одной из самых популярных СУБД для государственных и корпоративных организаций.

Основные характеристики:

* Надежность и согласованность: Поддержка транзакций (ACID) обеспечивает надежность и целостность данных.
* Расширяемость: Пользователи могут добавлять собственные функции, типы данных и операторы.
* Высокая производительность – оптимизированный механизм работы с многопоточной обработкой данных, увеличенная скорость работы с индексами и аналитическими запросами.
* Импортозамещение и безопасность – Postgres Pro полностью соответствует требованиям российских государственных организаций и может использоваться в государственном и корпоративном секторе.
* Масштабируемость: Поддержка репликации для горизонтального масштабирования. Репликация может быть полезным инструментом для повышения доступности, балансировки нагрузки и географического распределения данных. Тем не менее, она требует тщательного планирования и управления, чтобы избежать проблем с консистентностью данных и сложностями в администрировании.

## Описание бизнес-процесса

Бизнес-процесс — последовательность взаимосвязанных действий или операций, выполняемых для достижения конкретной бизнес-цели. Он описывает, как работа организована внутри компании, как информация и ресурсы передаются между отделами, а также как достигаются ключевые показатели эффективности[3].

Из информационной системы «Планирование производства» формируется список ДСЕ разрешенных к запуску, на основе которого ПДГ запрашивает наличие материала на складе и в случае наличия указанного материала создает СП на изготовление ДСЕ. СП сохраняется в информационной системе «Учета движения ДСЕ», после чего ПДГ печатает

СП. После печати паспорт становится недоступным для удаления и для изменения основной информации по паспорту (шапки).

На основании СП мастер создает заявку на получение материала. После этого кладовщик склада материалов выдаёт материал мастеру. После получения материала, мастер создает СЗ и вместе с материалом передает его рабочему. Рабочий получает материал, СЗ и выполняет операции по СЗ. После выполнения СЗ мастер проверяет выполнение задания.

Если изготовление ДСЕ завершено и выполнена последняя операция маршрута паспорта, то мастер проверяет работу и передаёт контролеру. Контролер проводит контрольные мероприятия, если контроль не пройден, то ДСЕ передаётся для работы с несоответствиями. Иначе контролер устанавливает признак готовности ДСЕ, признак готовности сохраняется в ИС. Далее ДСЕ передается на склад готовых ДСЕ по сдаточной накладной, после подтверждения кладовщиком склада ДСЕ сдаточной накладной происходит выгрузка отметки о выполнении позиции плана в ИС «Планирование производства».

Если изготовление ДСЕ не завершено и следующая операция не в этом цехе, то мастер передает материал кладовщику на склад для направления далее по маршруту , кладовщик передаёт ДСЕ по сдаточной накладной. При этом после подтверждения кладовщиком склада ДСЕ сдаточной накладной, происходит выгрузка отметки о выполнении позиции плана в ИС

«Планирование производства». Если следующая операция в этом же цехе, то мастер формирует новое задание и передает его рабочему.

В случае отсутствия на складе материалов материала с номенклатурным номером указанным в ВМ, ПДГ инициирует процесс «замены материала», формирует заявку на центральный склад, необходимый для изготовления ДСЕ, если на центральном складе есть материала то он отправляется на склад материалов и далее создается СП. Если материала нет на центральном складе формируется заявка на замену материала, запрашивается согласование на смену материала у технолога. Технолог утверждает замену по стандарту или

выпускает предварительное извещение, далее ПДГ вносит запись в электронный журнал учета разрешений и создает СП. В журнал вносятся данные: № СП, номер цеха, номер пункта разрешения, материал ТП, документ разрешающий запуск материала, примечание ПИ, ответственный за паспорт, ответственный за материал, эти данные сохраняются в журнале учета разрешения.

Мое веб-приложение реализует данный функционал, что поможет автоматизировать, ускорить и получить аналитическую информацию о проценте замен материала за заданный период. Также реализована возможность формирования печатного документа по запросу пользователя. Модель бизнес- процесса способна представить полную схему процесса и определить конкретные компоненты[4]. Бизнес-процесс представлен в приложении А.

## 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 импортонезависимое программное обеспечение:

Серверная часть:

* + Python – это высокоуровневый интерпретируемый язык программирования, известный своей удобочитаемостью и простотой. Python поддерживает множество парадигм программирования объектно- ориентированное и функциональное программирование [5].
  + Django – это высокоуровневый веб-фреймворк с открытым исходным кодом для создания веб-приложений. Он был разработан для обеспечения быстрой разработки и создания качественных и надежных веб- приложений. Django построен на архитектуре MVC.
  + MVC («Model-View-Controller», «Модель-Вид-Контроллер») – является архитектурным шаблоном программного обеспечения, обычно используемым для разработки пользовательских интерфейсов. Этот

шаблон делит приложение на три взаимосвязанных компонента . В Django, MVC отображается в паттерне Model-View-Template (MVT), где модель (Model) определяет структуру данных и бизнес логику, она взаимодействует с базой данных и представляет данные приложения с помощью ORM. Представление (View) выступает в качестве посредника между моделью и шаблоном, оно обрабатывает запросы пользователей, извлекает необходимые данные из модели и определяет какой шаблон отображать. Шаблоны (Template) отвечает за визуализацию данных на сайт конечному пользователю [11].

1. Django Rest Framework (DRF) – это мощная библиотека для Python созданная специально для разработки RESTful API Это инструмент который позволяет связать серверное приложение с другими сервисами предоставляя стандартизированный доступ к данным через удобные методы. Основные возможности DRF включают сериализацию данных которая позволяет легко преобразовывать данные из базы в JSON-формат и обратно работает как переводчик между сервером и клиентом понимая запросы пользователя и отправляя ответы в удобном формате. Автоматическое создание API всего несколько строк кода позволяют создать базовый API для работы с данными DRF автоматически генерирует маршруты формы запросов и документацию. Гибкая настройка возможность настроить сложные фильтры добавить авторизацию или реализовать кастомную логику под конкретные нужды проекта Встроенная документация интерактивная документация помогает разработчикам и служит удобным инструментом для тестирования API [18].
2. API (Application Programming Interface) – это набор строго определенных правил, протоколов и инструментов, которые позволяют различным программным приложениям взаимодействовать между собой. API выступает в роли посредника, который обеспечивает стандартизированный обмен данными между разными системами,

независимо от их внутренней реализации.

1. HTTP (Hypertext Transfer Protocol) – или протокол передачи гипертекста, - это протокол прикладного уровня, используемый для передачи гипертекста в Интернете . 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) – язык гипертекстовой разметки, является стандартным языком разметки, используемым для создания и оформления документов во Всемирной паутине. Он обеспечивает структуру веб-страниц [17].
  + CSS (Cascading Style Sheets) – это язык таблиц стилей, используемых для описания представления и макета документа, написанного в формате HTML.
  + JavaScript – это высокоуровневый, интерпретируемый язык программирования, который изначально создавался для добавления интерактивности на веб-страницы, но сегодня используется практически во всех сферах разработки. Благодаря своей гибкости, динамической типизации и асинхронной модели выполнения, JavaScript стал основой современного веба и серверной разработки.
  + Bootstrap – это популярный интерфейсный фреймворк с открытым исходным кодом, предназначенный для облегчения веб-разработки за счет представления компонентов HTML, CSS и JavaScript. Основные преимущества Bootstrap включают: Быстрое создание интерфейсов благодаря готовым шаблонам и компонентам, которые можно легко настроить. Адаптивность, позволяющую автоматически адаптировать элементы интерфейса к различным устройствам и разрешениям экрана. Кросс-браузерность, обеспечивающую корректное отображение веб-

страницы в разных браузерах и операционных системах.

Для этого стека был выбран редактор исходного кода Visual Studio Code. 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. Visual Studio Code предлагает широкий набор функций и инструментов. Среда разработки представлена на рисунке 1.1.

Рисунок 1.1 — Редактор кода Visual Studio Code

Для разработки веб-приложения, была использована стандартная для предприятия и указанная в техническом задании СУБД Postgres Pro [6]. Управление СУБД осуществляется через графический клиент pgAdmin (см. рисунок 1.2).

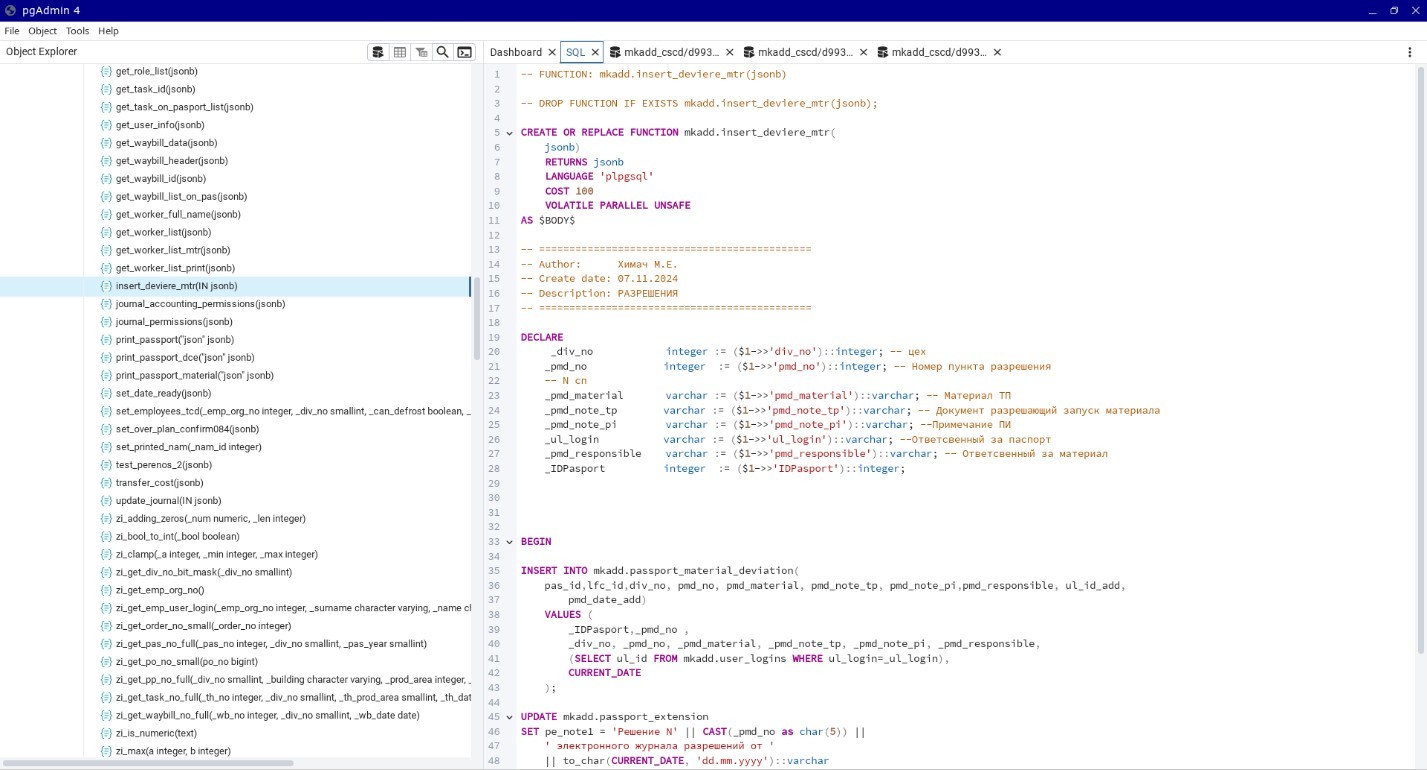


Рисунок 1.2 — Графический клиент pgAdmin

Вывод: в соответствии с техническим заданием на разработку веб-сервиса

«Сопровождения ведения журнала учета разрешений замены материала в ИС МКАДД» 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

## Построение информационной модели «сущность-связь»

Модель «сущность-связь» (Entity-Relationship model) – представляет собой мощный инструмент концептуального проектирования баз данных, который позволяет наглядно отображать структуру данных предметной области [13]. Она является ключевым этапом в процессе проектирования эффективной и надежной базы данных, поскольку позволяет визуализировать и понять взаимосвязи между различными элементами информации до начала фактической реализации. ER-моделирование помогает избежать дорогостоящих ошибок на более поздних этапах разработки.

Информационная модель «сущность-связь» больше чем просто метод обработки данных, благодаря которому можно описывать отношения между объектами, называемыми «сущностями», в информационной системе [10]. Это полноценный подход к моделированию, который позволяет бизнесу и разработчикам совместно определять и документировать ключевые концепции, правила и ограничения, относящиеся к данным. ER-модель является своего рода “словарем” данных, понятным как техническим, так и бизнес- специалистам.

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

* + Сущность (Entity) – это базовый строительный блок модели, представляющий собой объект или концепцию предметной области, информацию о котором необходимо хранить. Сущность может быть физическим объектом (например, “Клиент”, “Товар”), абстрактным понятием (например, “Заказ”, “Категория”) или событием (например, “Транзакция”). Важно четко определить, какие сущности важны для предметной области и какие данные о них необходимо хранить.
  + Атрибут (Attribute) – характеристика сущности. Атрибуты описывают свойства сущности и содержат фактические данные. Каждый атрибут

имеет имя и тип данных (например, “Имя клиента” (текст), “Возраст клиента” (число), “Дата заказа” (дата)). Атрибуты могут быть простыми (неделимыми) или составными (состоящими из нескольких частей), обязательными или необязательными, уникальными или неуникальными. При выборе атрибутов важно учитывать потребности бизнеса и требования к отчетности.

* + Связь (Relationship) – отношение, которое связывает две или больше сущностей. Связи определяют, как сущности взаимодействуют друг с другом. Они могут быть разных типов, например:
    - Один-к-одному (1:1): Один экземпляр сущности A связан с одним экземпляром сущности B. (Например, “Гражданин” и “Паспорт” – у каждого гражданина один паспорт, и у каждого паспорта один владелец).
    - Один-ко-многим (1:N): Один экземпляр сущности A связан с несколькими экземплярами сущности B. (Например, “Клиент” и “Заказ” – один клиент может сделать несколько заказов, но каждый заказ относится только к одному клиенту).
    - Многие-ко-многим (N:M): Несколько экземпляров сущности A связаны с несколькими экземплярами сущности B. (Например, “Студент” и “Курс” – один студент может посещать несколько курсов, и на каждом курсе могут учиться несколько студентов).

При создании ER-модели важно определить кардинальность (тип) связи, а также указать, является ли связь обязательной или необязательной для каждой сущности. Это позволяет точно отразить бизнес-правила и обеспечить целостность данных.

ER-моделирование остается фундаментальной методикой проектирования информационных систем, обеспечивающей четкое структурирование данных и выявление бизнес-правил предметной области. Оно служит мостом между

концептуальным пониманием системы и ее технической реализацией. Это позволяет разработчикам создавать более эффективные и точные базы данных любого уровня сложности. Оно помогает гарантировать, что база данных соответствует требованиям бизнеса и обеспечивает необходимую функциональность. С помощью ER-модели можно проектировать логическую модель базы данных, которая затем может быть реализована в конкретной СУБД (системе управления базами данных). ER-моделирование также упрощает коммуникацию между разработчиками, аналитиками и бизнес- пользователями, что приводит к более успешным проектам.

На рисунке 2.1 представлена ER-модель, использующаяся в проекте. Важно помнить, что ER-модель является итеративным процессом. Она может и должна развиваться по мере уточнения требований и углубления понимания предметной области.

Таблица 2.1 — Нотация к диаграмме ER

|  |  |
| --- | --- |
| Элемент | Значение |
|  | Сущность |
|  | Связь один к одному |
|  | Связь один ко многим |

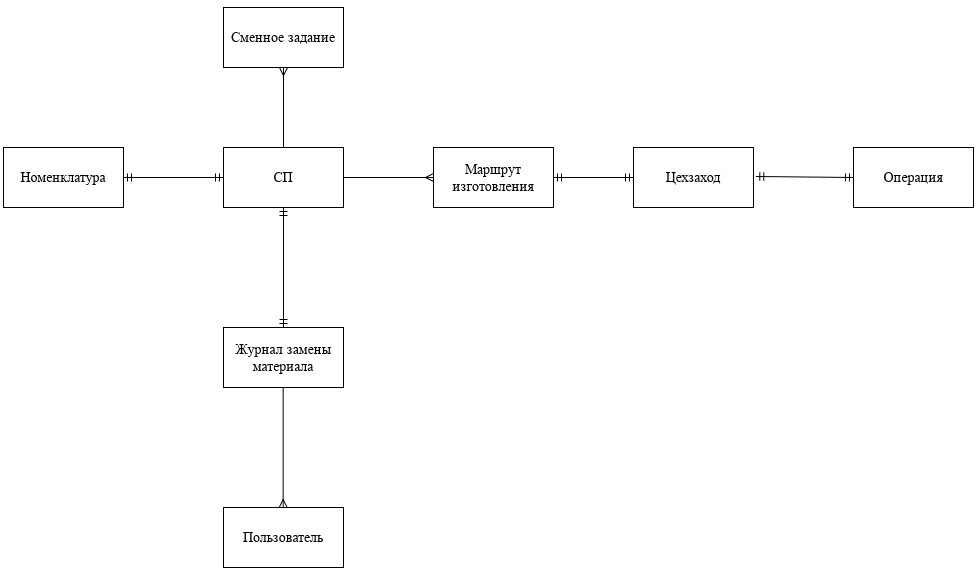


Рисунок 2.1 — ER-модель, использующаяся в проекте

## Проектирование базы данных

Физическая модель базы данных представляет собой техническую реализацию логической модели, адаптированную под конкретную систему управления базами данных. Этот уровень проектирования включает следующие аспекты:

Структура хранения данных:

* + - Определение форматов хранения для каждого атрибута(типы данных, размеры полей);
    - Схемы физического размещения данных (табличные пространства, файловые группы);
    - Стратегии секционирования и кластеризация данных;
    - Механизмы доступа и индексирования;
    - Проектирование системы индексов;
    - Оптимизация составных и покрывающихся индексов;
    - Реализация специализированных структур доступа для пространственных данных;
    - Реализация связей и ограничений;
    - Физическая реализация связей через внешние ключи;
    - Ограничение целостности;
    - Каскадные операции обновления и удаления;
    - Триггеры для поддержки сложных бизнес-правил;
    - Аспекты производительности;
    - Профилирование и оптимизация запросов;
    - Стратегии кэширования часто используемых данных;
    - Параллельная обработка сложных запросов;
    - Настройка параметров СУБД;

Физическая модель служит критически важным связующим звеном между:

* + - Концептуальным проектом;
    - Реализацией в конкретной СУБД;

Уровень проектирования требует глубокого понимания как принципов работы конкретной СУБД, так и особенностей предметной области, что делает его критически важным для создания эффективных масштабируемых и надежных систем хранения [15]. Таким образом, физическая модель необходима для создания и настройки базы данных, чтобы обеспечить правильную организацию хранения и доступа к данным, а также максимальную

производительность базы данных в целом [9].

Для работы приложения используются следующие таблицы: Таблица 2.2 – Таблицы базы данных

|  |  |
| --- | --- |
| Таблица | Роль таблицы |
| user\_logins | Таблица пользователей |
| passport\_material\_deviation | Таблица журнала замены материала |
| passport | Таблица паспортов |
| passport\_nomenclature | Таблица номенклатуры на паспорте |
| division\_nomenclature | Таблица номенклатуры |
| passport\_extension | Таблица расширений паспорта |
| passport\_lfc | Таблица паспорта ЛЗК |

Физическая модель БД для отображения данных представлена в приложении Б.

Таблица «Паспорт» отвечает за хранение данных о сопроводительных паспортах. Данная таблица имеет следующие атрибуты:

* + - Идентификатор паспорта;
    - Номер паспорта;
    - Год;
    - Цех;
    - Количество ДСЕ;
    - Заказ;
    - Ответственный;
    - Количество заготовок;
    - Приоритет;
    - Образцы;
    - Ответственный за удаление;
    - Ответственный за печать;
    - Дата заполнения;
    - Дата печати;
    - Дата удаления паспорта;
    - Когда изменен остаток на паспорте;
    - Признак заготовки;
    - Дата закачки готовых паспортов из данных цеха;
    - Фамилия контролера на паспорте;

Первичным ключом является поле pas\_id (Идентификатор паспорта). Таблица «Паспорт» имеет связи сразу с несколькими таблицами. Связь «один- к-одному» имеет с «Паспорт номенклатуры», «Паспорт расширения», «Журнал разрешений».

Таблица «Паспорт номенклатуры» отвечает за хранение данных о паспорте номенклатуры. Данная таблица имеет следующие атрибуты:

* + - Идентификатор паспорта;
    - Идентификатор номенклатуры;
    - Код принадлежности;
    - Ответственный;
    - Идентификатор ограничения;
    - Типоразмер;
    - Наименование;

Первичным ключом является поле pas\_id (Идентификатор паспорта).

Таблица «Паспорт расширения» отвечает за хранение данных о дополнительных атрибутах паспорта. Данная таблица имеет следующие атрибуты:

* + - Идентификатор паспорта;
    - Номер изменения;
    - Код МТР;
    - Номер позиции;
    - Признаки наличия затрат;
    - Примечание;
    - Размер заготовки;
    - Номер карточки;

Первичным ключом является поле pas\_id (Идентификатор паспорта).

Таблица «Журнал разрешений» отвечает за хранение данных о журнале разрешений замен материла. Данная таблица имеет следующие атрибуты:

* + - Идентификатор паспорта;
    - Идентификатор лзк;
    - Код МТР;
    - Номер разрешения;
    - Цех;
    - Ответственный за ведение;
    - Ответственный за изменение;
    - Дата добавления;
    - Дата изменения;
    - Ответственный за материал;
    - Материал по ТП;
    - Примечание ТП;
    - Примечание ПИ;

Первичным ключом является поле pas\_id (Идентификатор паспорта).

Таблица «Пользователи» отвечает за хранение данных о пользователях.

Данная таблица имеет следующие атрибуты:

* + - Идентификатор логина;
    - Табельный;
    - Цех;
    - Дата добавления;
    - Фамилия;
    - Имя;
    - Отчество;
    - Логин;

Первичным ключом является поле ul\_id (Идентификатор логина). С данной таблицей имеет связь «один-ко-многим» таблица «Журнал разрешений».

Таблица «Номенклатуры» отвечает за хранение данных о номенклатуре.

Данная таблица имеет следующие атрибуты:

* + - Идентификатор номенклатуры;
    - Код принадлежности;
    - Табельный ответственного;
    - Код МТР;
    - Дата заполнения;
    - Дата обновления;
    - Обозначение;
    - Наименование;
    - Типоразмер;
    - Номер ВМ;

Первичным ключом является поле dn\_id (Идентификатор номенклатуры). Данная таблица имеет связь «один-ко-многим» с таблицей «Паспорт номенклатуры».

Таблица «Паспорт ЛЗК» отвечает за хранение данных о ЛЗК на паспорте.

Данная таблица имеет следующие атрибуты:

* + - Идентификатор паспорта;
    - Идентификатор лзк;
    - Табельный добавления;
    - Дата добавления;
    - Материал;

Первичным ключом является поле pas\_id(Идентификатор паспорта).

Схема в PostgreSQL представляет собой важный механизм организации и управления объектами базы данных, выполняющий функцию пространства имен для логической группировки таблиц, представлений, функций и других сущностей. База данных разделена на несколько схем, для выполнения технического задания (см. рисунок 2.2) использовалась схема mkadd, которая служила основным контейнером для всех производственных данных и бизнес логики приложения.

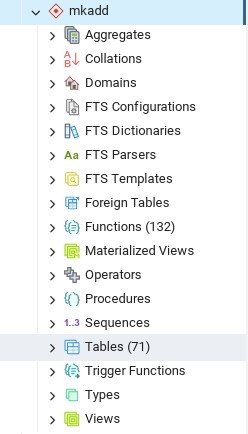


Рисунок 2.2 — Схема mkadd

Для обеспечения функциональности, необходимой для непосредственного взаимодействия с данными на сервере, целесообразно применять хранимые процедуры и функции.

В PostgreSQL хранимая процедура – это заранее написанный код, используемый для обработки данных в базе данных [7]. Она способна выполнять различные действия, включая изменение или удаление данных, а

также запускать сложные запросы к базе данных.

Функция базы данных – представляет собой блок кода на языке SQL или его аналогах. Функции предназначены для выполнения повторяющихся действий, упрощая работы с данными и улучшая скорость выполнения запросов [12]. В отличии от хранимых процедур, функции могут возвращать значения, принимать аргументы и использовать разнообразные встроенные операторы [19].

Функция journal\_accounting\_permissions необходима для получения всех данных по журналу (см. рисунок2.3). На вход функция journal\_accounting\_permissions принимает:

1. Номер разрешения;
2. Дата начала;
3. Дата окончания;
4. Номер цеха.

На выход формируется json с данными по журналу разрешений.



Рисунок 2.3 — Функция journal\_accounting\_permissions

Функция insert\_deviere\_mtr необходима для добавления данных в журнал (см. рисунок 2.4).На вход функция insert\_deviere\_mtr принимает:

1. Номер цеха;
2. Материал ТП;
3. Документ разрешающий запуск материала;
4. Примечание ПИ;
5. Ответственный за паспорт;
6. Ответственный за материал;

На выходе функция создает новую запись в журнале замены материала.



Рисунок 2.4 — Функция insert\_deviere\_mtr

Функция journal\_permissions необходима для получения одной конкретной записи в журнале (см. рисунок 2.5). На входе функция принимает идентификатор паспорта.

На выходе формируется jsonb с данными о конкретном журнале.

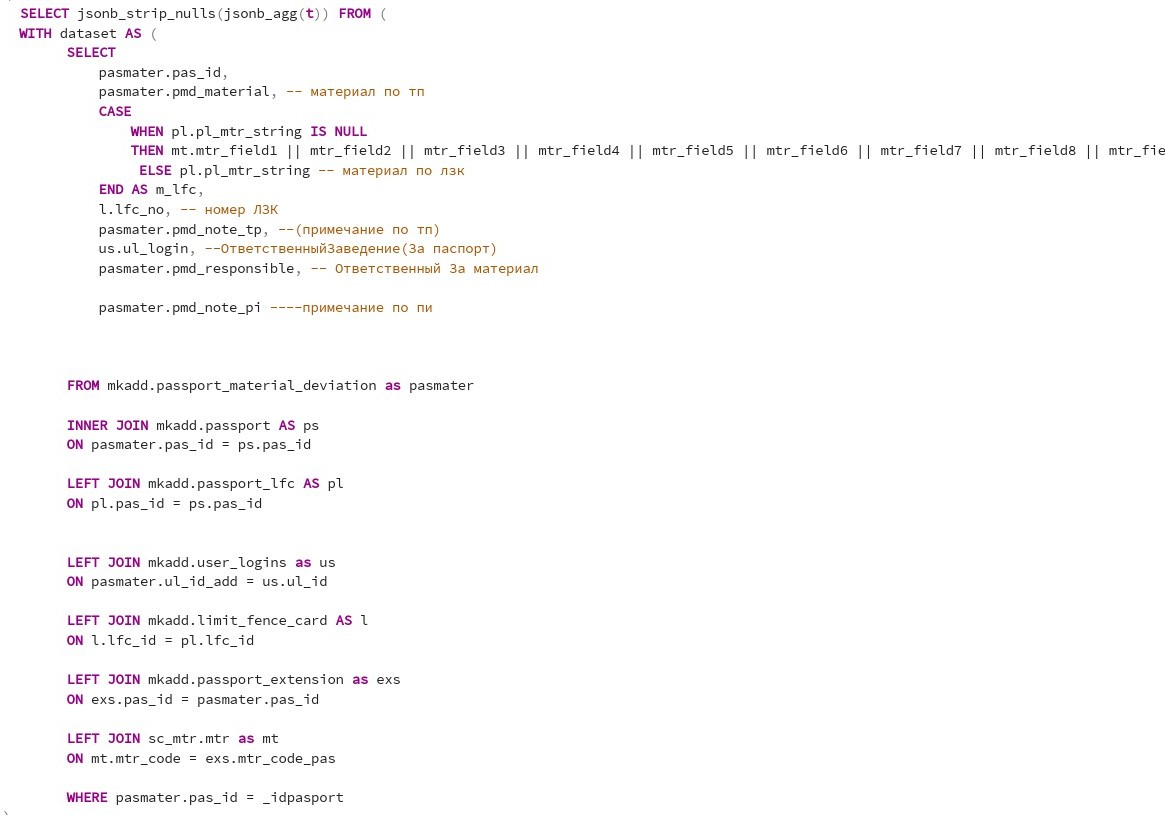


Рисунок 2.5 — Функция journal\_permissions

Функция update\_journal необходима для обновления данных в журнал (см. рисунок 2.6).На вход функция принимает:

1. Идентификатор паспорта;
2. Документ разрешающий запись материала;
3. Примечание по ПИ.

На выходе функция обновляет запись в журнале замены материала.



Рисунок 2.6 — Функция update\_journal

Функция analytic\_jour необходима для получения аналитики (см. рисунок 2.7). На вход функция принимает:

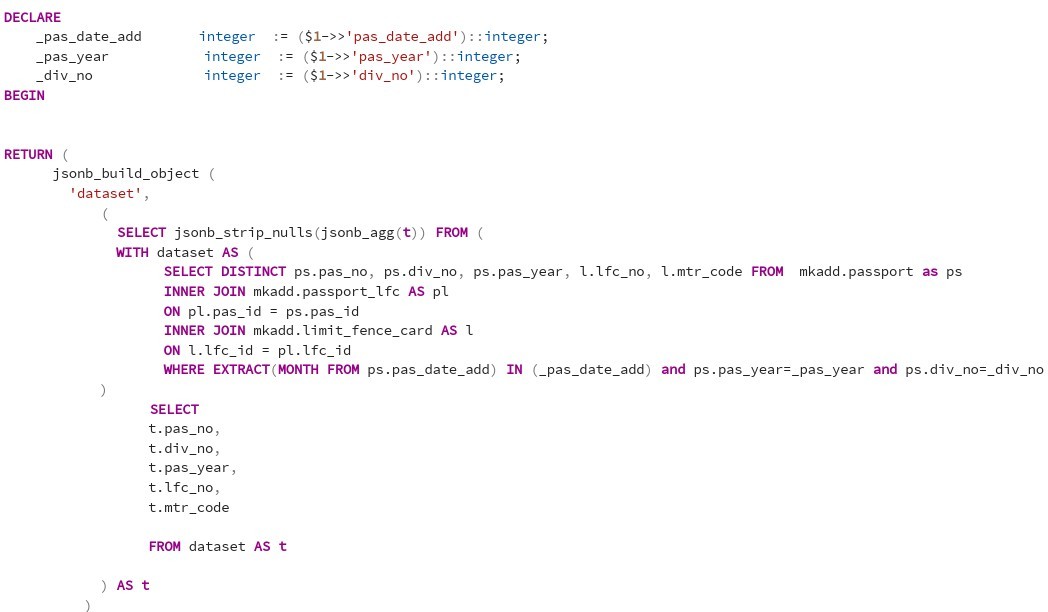
1. Номер цеха;
2. Месяц;
3. Год.

Рисунок 2.7 — Функция analytic\_jour

## Проектирование клиентской части

Клиентская часть играет ключевую роль в создании удобного и интуитивно понятного интерфейса. Она отвечает за визуальное представление данных, взаимодействие с пользователем и обмен информацией с сервером [8].

**Основные этапы разработки:**

* 1. Проектирование пользовательского интерфейса:
     1. Разработка макетов страниц с учетом удобства навигации;
     2. Подбор цветовых схем, шрифтов и адаптивного дизайна для корректного отображения на разных устройствах.
  2. Обеспечение удобного взаимодействия:
     1. Реализация интерактивных элементов;
     2. Валидация ввода данных.
  3. Оптимизация кода:
     1. Применение компонентного подхода для упрощения поддержки;
     2. Использование шаблонов для разделения структуры и уникальных блоков.

Клиентская часть должна обеспечивать быстродействие и соответствовать современным стандартам веб-разработки. На рисунке представлена начальная страница. С ее помощью можно перейти на любую другую страницу разрабатываемого приложения (см. рисунок 2.8).

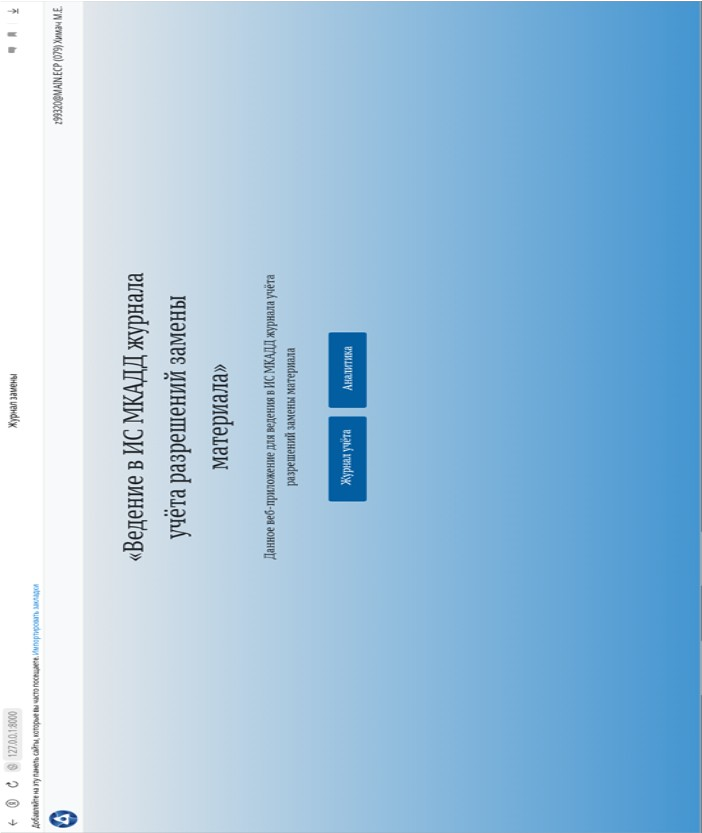


Рисунок 2.8 — Начальная страница

На странице журнал учета разрешений представленная информация о журналах в виде таблицы с возможностью фильтрации данных. Также есть возможность получить нужные нам записи журнала в виде pdf для дальнейшей печати, для этого необходимо нажать на кнопку «Печать журнала». Кроме того, можно перейти в раздел с аналитикой, созданием нового журнала или

изменением конкретного журнала (см. рисунок 2.9).

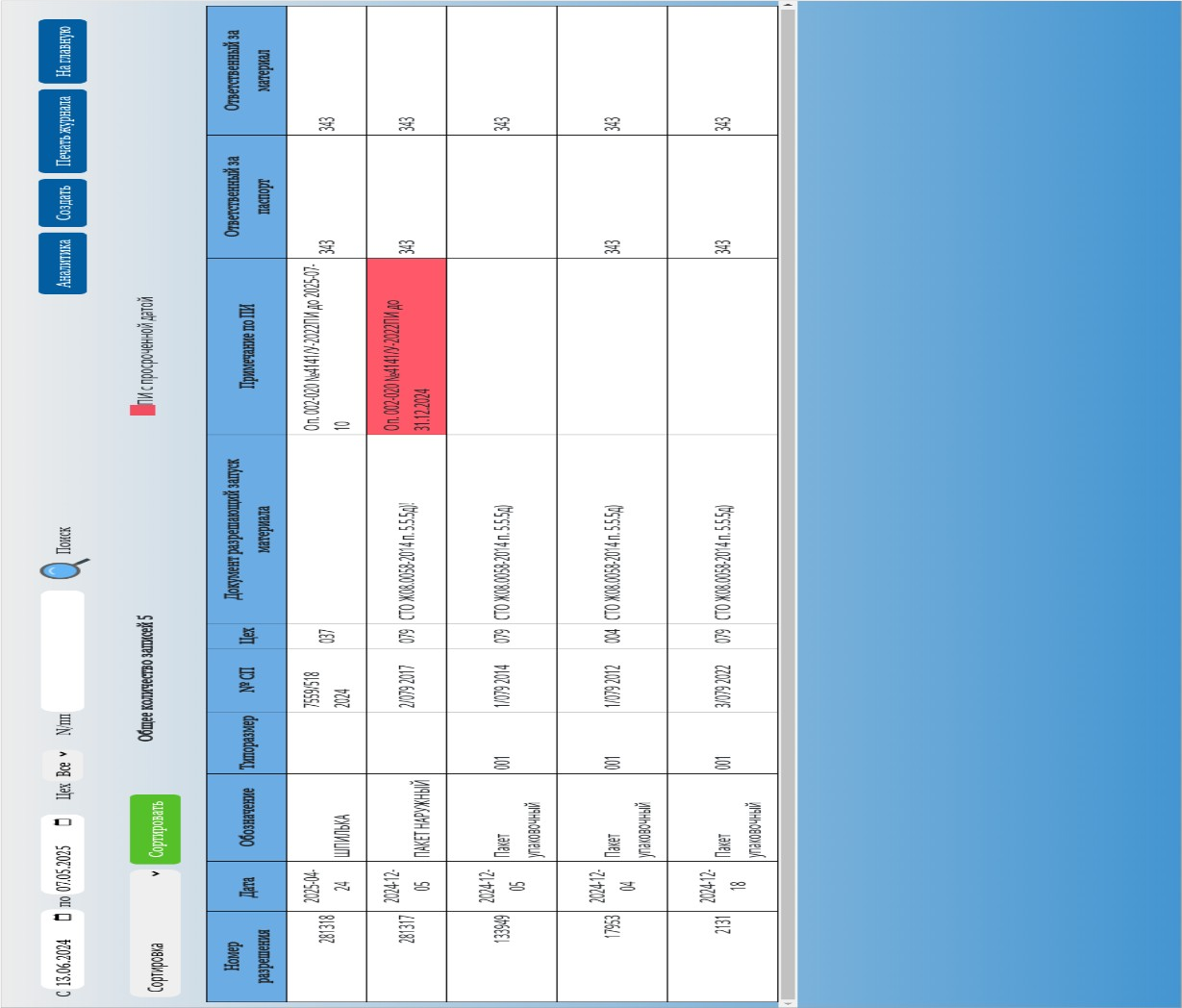


Рисунок 2.9 — Страница с журналом учета замены материала

Форма создания записи в журнал замены материала, представленная на рисунке 2.10, включает в себя несколько ключевых элементов, необходимых для корректного учета и документирования процесса замены.

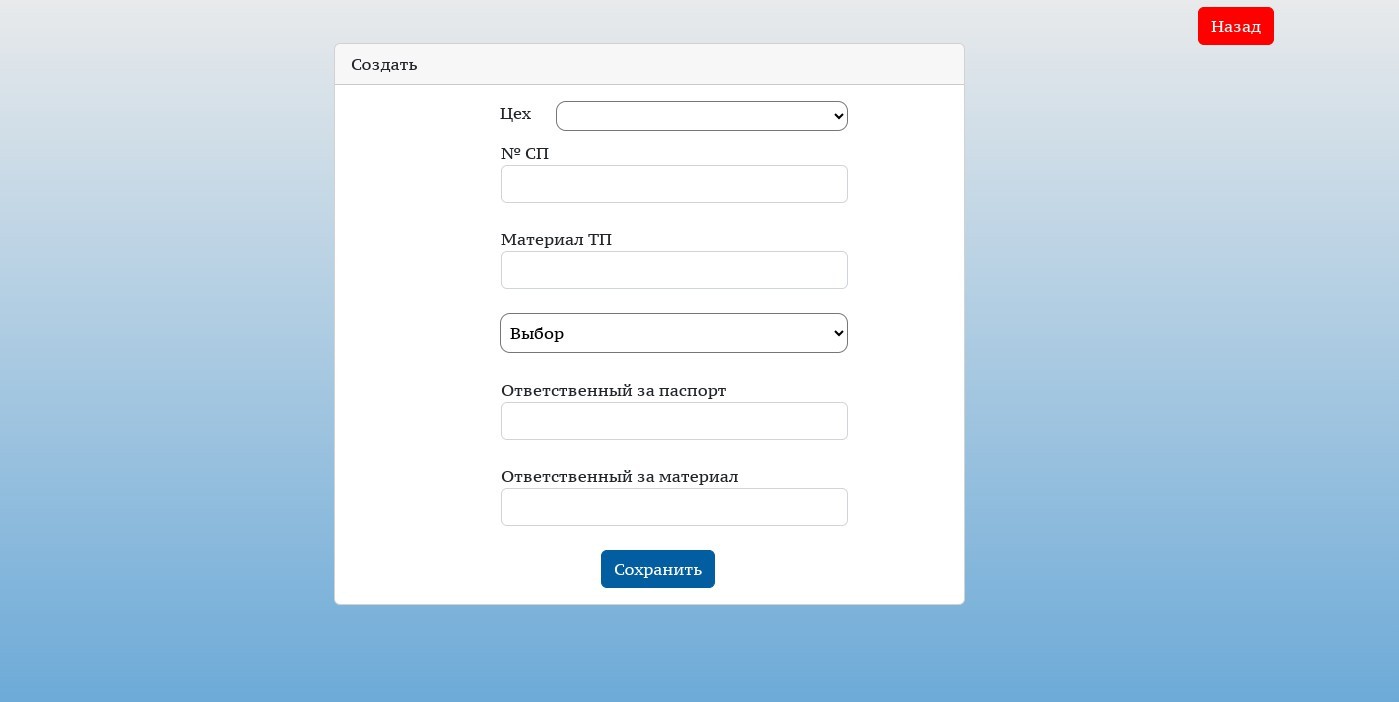


Рисунок 2.10 — Форма создания записи в журнале замены

При создании можно выбрать документ разрешающий запуск материала

(см. рисунок 2.11), а если нет необходимых условий в стандарте, то используется примечание ПИ (см. рисунок 2.12).

Рисунок 2.11 — Форма создания записи в журнал замены

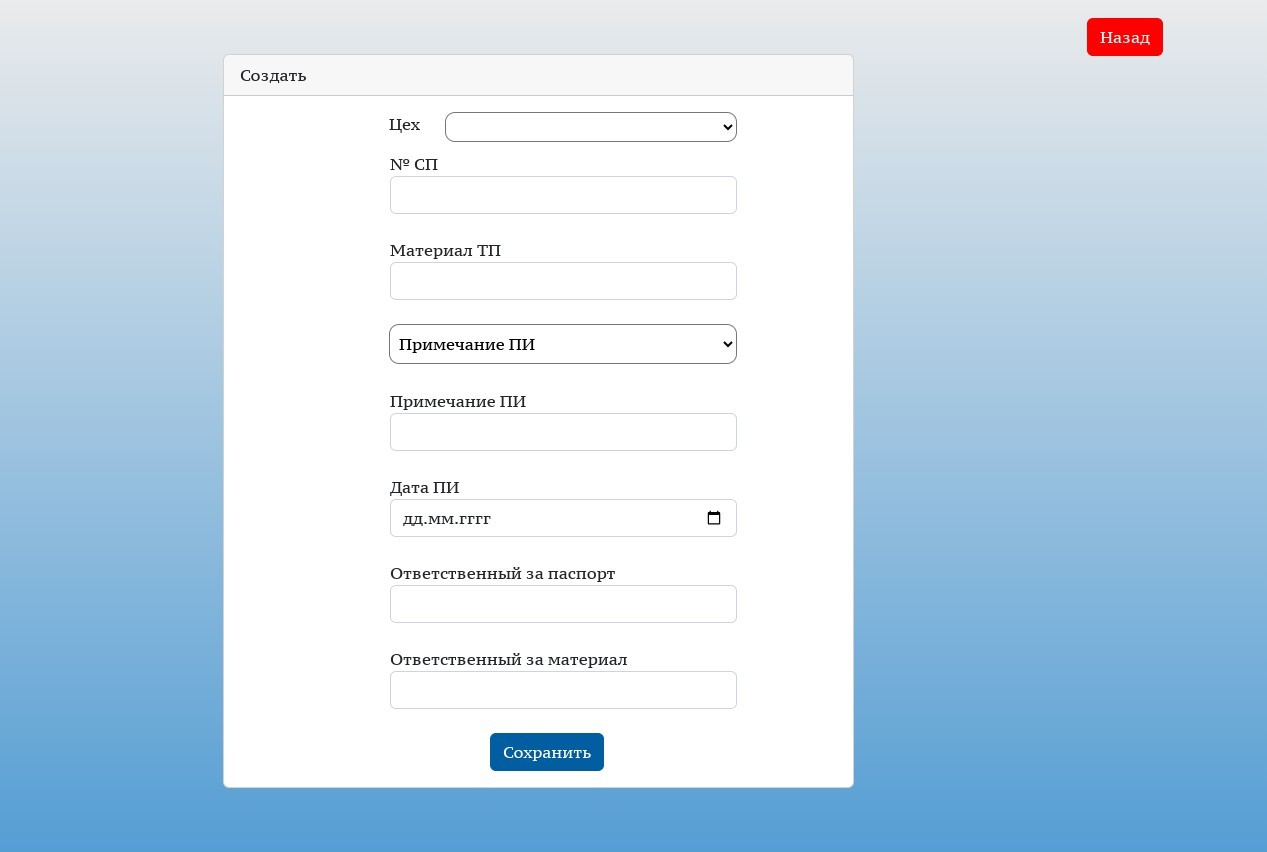


Рисунок 2.12 — Форма создания записи в журнал замены

На рисунке 2.13 представлена форма изменения текущей записи в журнале замены материала.

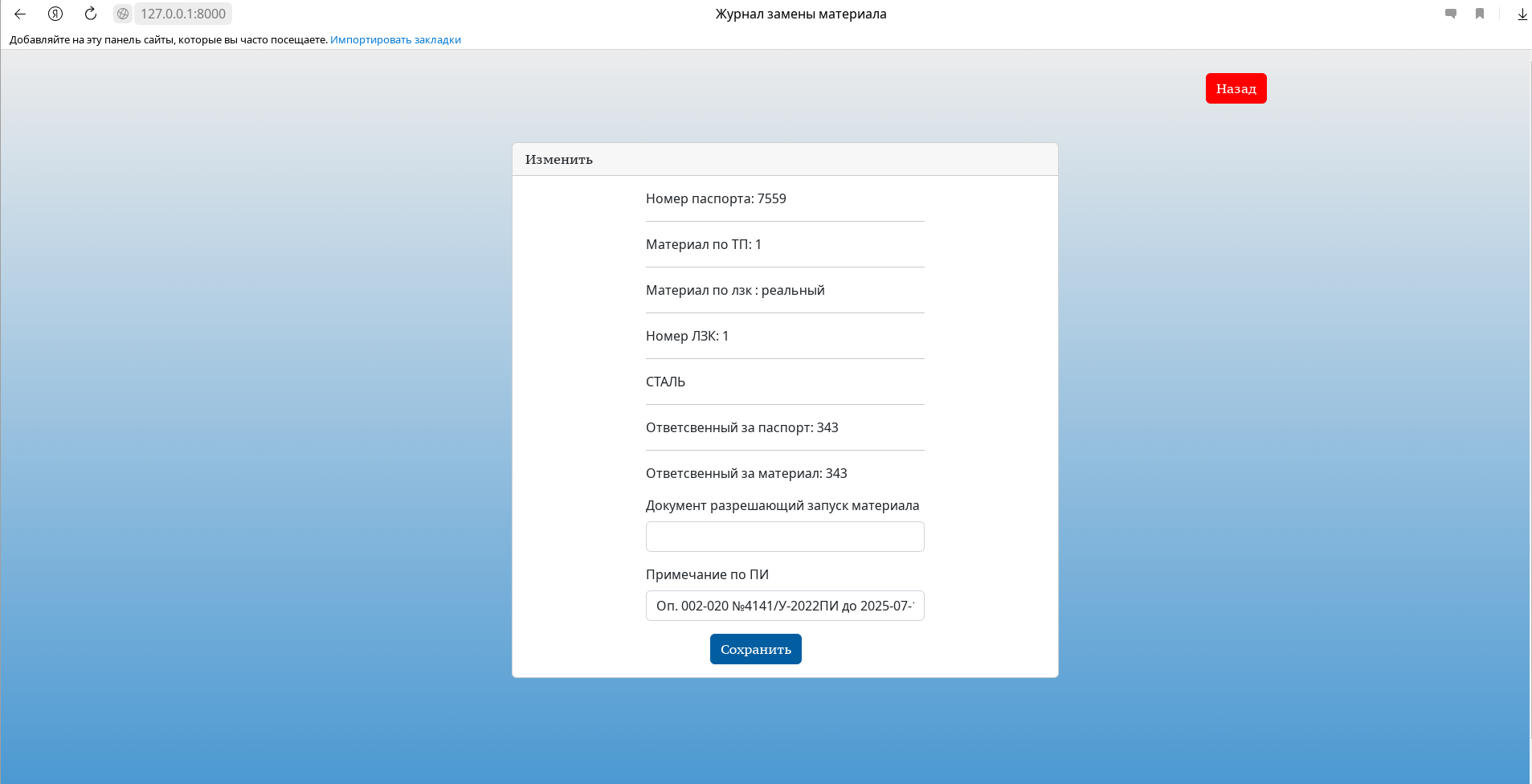


Рисунок 2.13 — Форма изменения записи в журнал замены

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

1. Построена информационная модель «сущность-связь»;
2. Спроектирована база данных;
3. Разработана клиентская часть приложения.

# ГЛАВА 3 ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ

* 1. **Руководство пользователя**

Программный продукт «Ведение журнала учета разрешений замены материала ИС «МКАДД» представляет собой современное веб-приложение, обеспечивающее эффективное управление процессами учета и контроля замен материалов в информационной системе. Весь функционал реализован через клиентскую часть, которая функционирует в виде веб-сервиса и может быть удобно доступна через браузер на рабочем месте пользователя.

Пользователи могут легко взаимодействовать с приложением, что гарантирует высокую степень удобства и адаптивности. За счет использования веб-технологий, продукт не требует установки клиентского модуля на локальные рабочие станции, что значительно упрощает процесс его развертывания и способствует эффективному использованию имеющегося оборудования.

Технические требования к оборудованию, необходимому для установки и

корректной работы веб-сервиса, описываются следующей минимальной конфигурацией:

* + - Процессор: Intel Pentium 4 — достаточно производительный для обеспечения нормальной работы приложения.
    - Оперативная память: 4 Гб — этот объем оперативной памяти позволяет системе эффективно обрабатывать запросы пользователей и выполнять необходимые операции без значительных задержек.
    - Жесткий диск: 5 Гб — минимальное свободное пространство, требуемое для установки и хранения всех необходимых компонентов веб-сервиса, а также баз данных, что обеспечивает корректное и надежное функционирование системы.
    - Операционная система: Astra Linux — данная система хорошо зарекомендовала себя в области безопасности и стабильности, что делает ее идеальным выбором для использования в государственных и корпоративных структурах.

Важно отметить, что благодаря веб-архитектуре, пользователи могут получать доступ к системе из любой точки, где есть соединение с сетью предприятия, что способствует улучшению мобильности и гибкости работы сотрудников. А также гарантирует быстрое обновление программного обеспечения без необходимости вмешательства в клиентские приложения.

Данный продукт представляет собой решение, которое значительно повышает эффективность и прозрачность процессов управления материалами, способствует оптимизации работы ресурсами и позволяет вовремя реагировать на изменения и требования, возникающие в ходе эксплуатации материалов в рамках информационных систем.

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

Начальная страница: при открытии HTML-страницы возможно перейти

на любую другую страницу приложения (см. рисунок 2.8).

При выборе страницы «Журнал учета разрешений» в браузере должна отразится страница с журналом учета замены материала (рисунок 2.8), данные представлены в виде таблицы, также на этой странице находится фильтрация с помощью, которой можно фильтровать по дате, цехам, N/пп. Также есть возможность сортировки по дате, по ПИ и по документу разрешающий запись материала (рисунок 3.1).



Рисунок 3.1 — Применение фильтрации и сортировки к таблице

Для удобства при работе с таблицей реализована функция подсветки строк. Эта функция позволяет выделять строку при наведении курсора, что значительно улучшает восприятие данных и упрощает навигацию (см. рисунок 3.2).

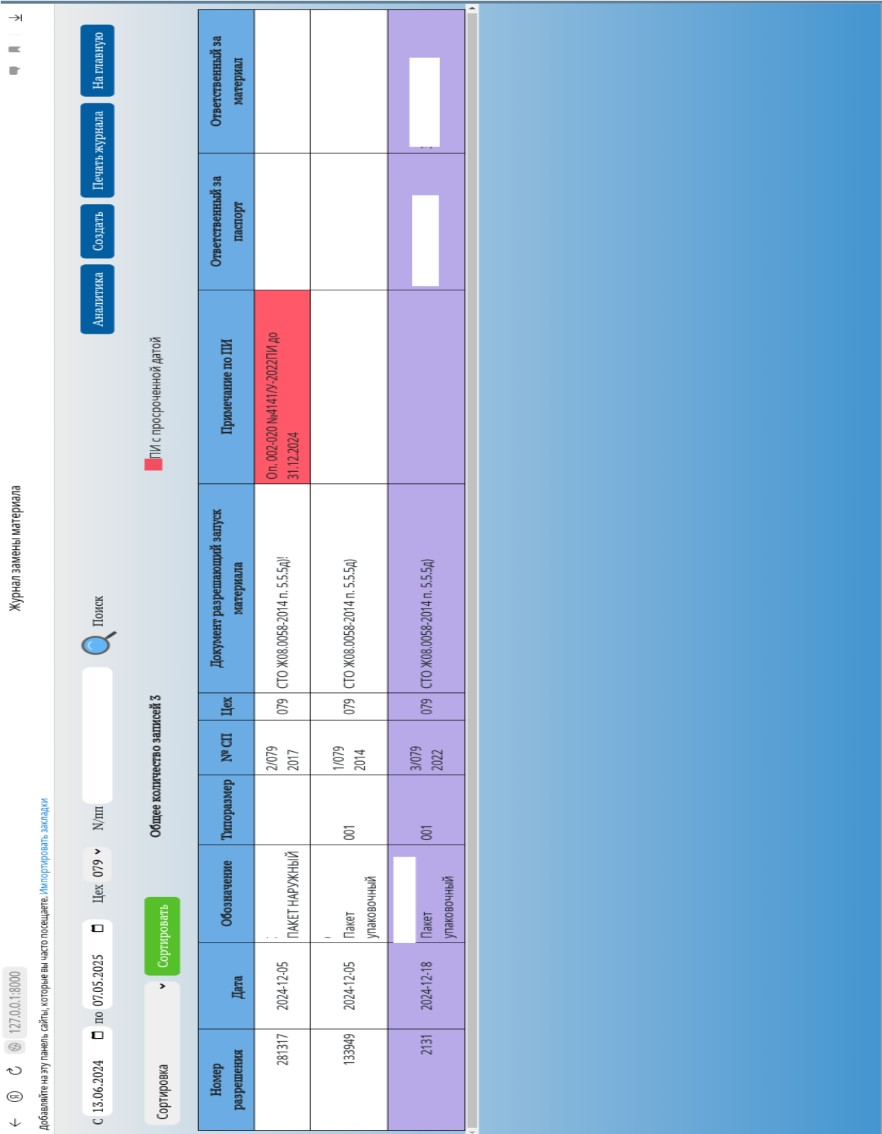


Рисунок 3.2 — Подсветка строк при наведении

При нажатии на кнопку «Создать» откроется форма для создания записи в журнал учета замены материала (см. рисунок 2.13). После заполнения всех строк необходимо нажать на кнопку «Сохранить», если данные будут

заполнены не правильно на форме появится соответствующая запись (см. рисунок 3.3).

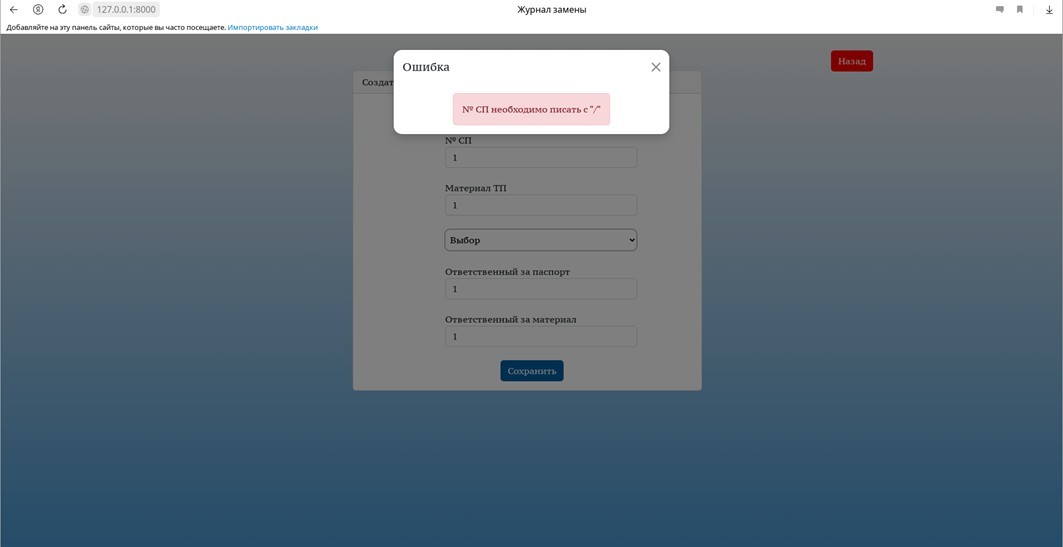


Рисунок 3.3 — Валидация формы

После создания снова откроется страница с журналом учета разрешений замены материала (см. рисунок 3.4).

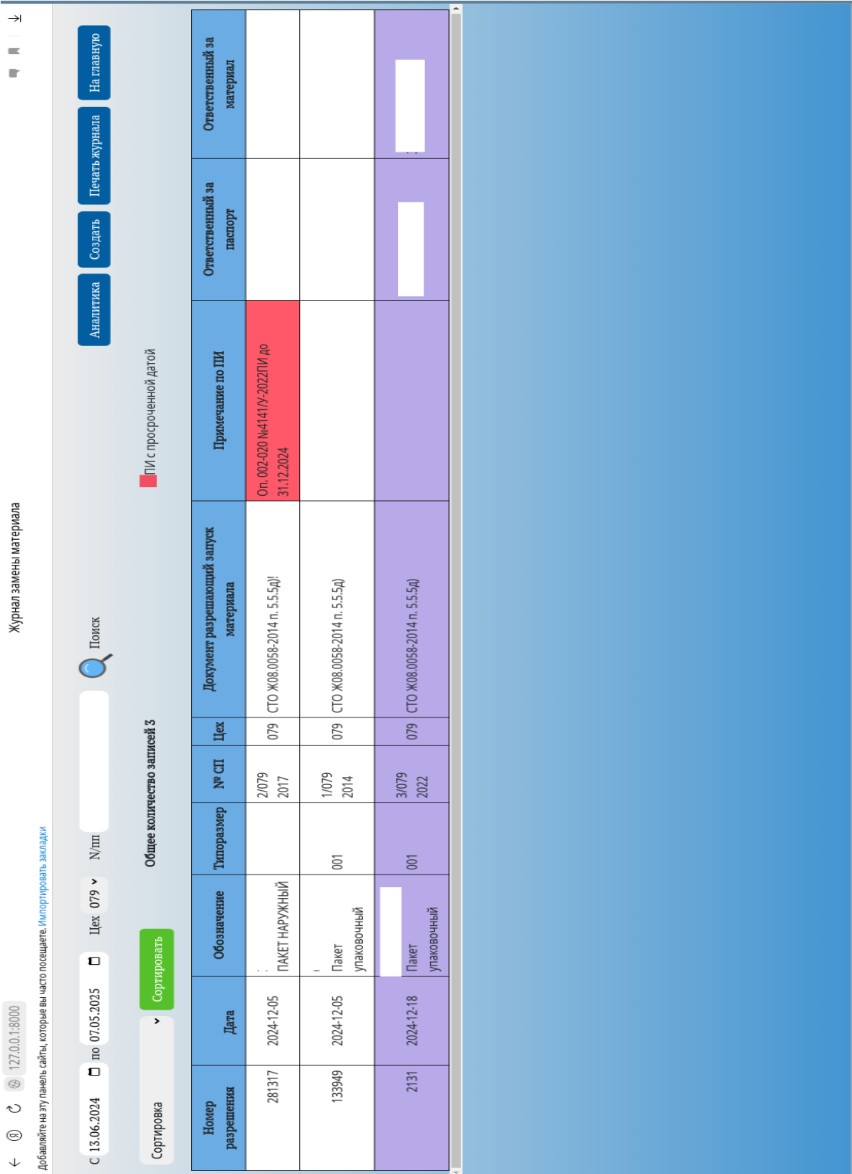


Рисунок 3.4 — Основная страница после создания записи

Для того чтобы изменить запись необходимо два раза нажать на строку, после этого откроется форма изменения записи (рисунок 2.10), после успешного изменения откроется основная страница и строка, которую меняли для удобства будет подсвечена (см. рисунок 3.5).

Рисунок 3.5 — Выделение измененной записи

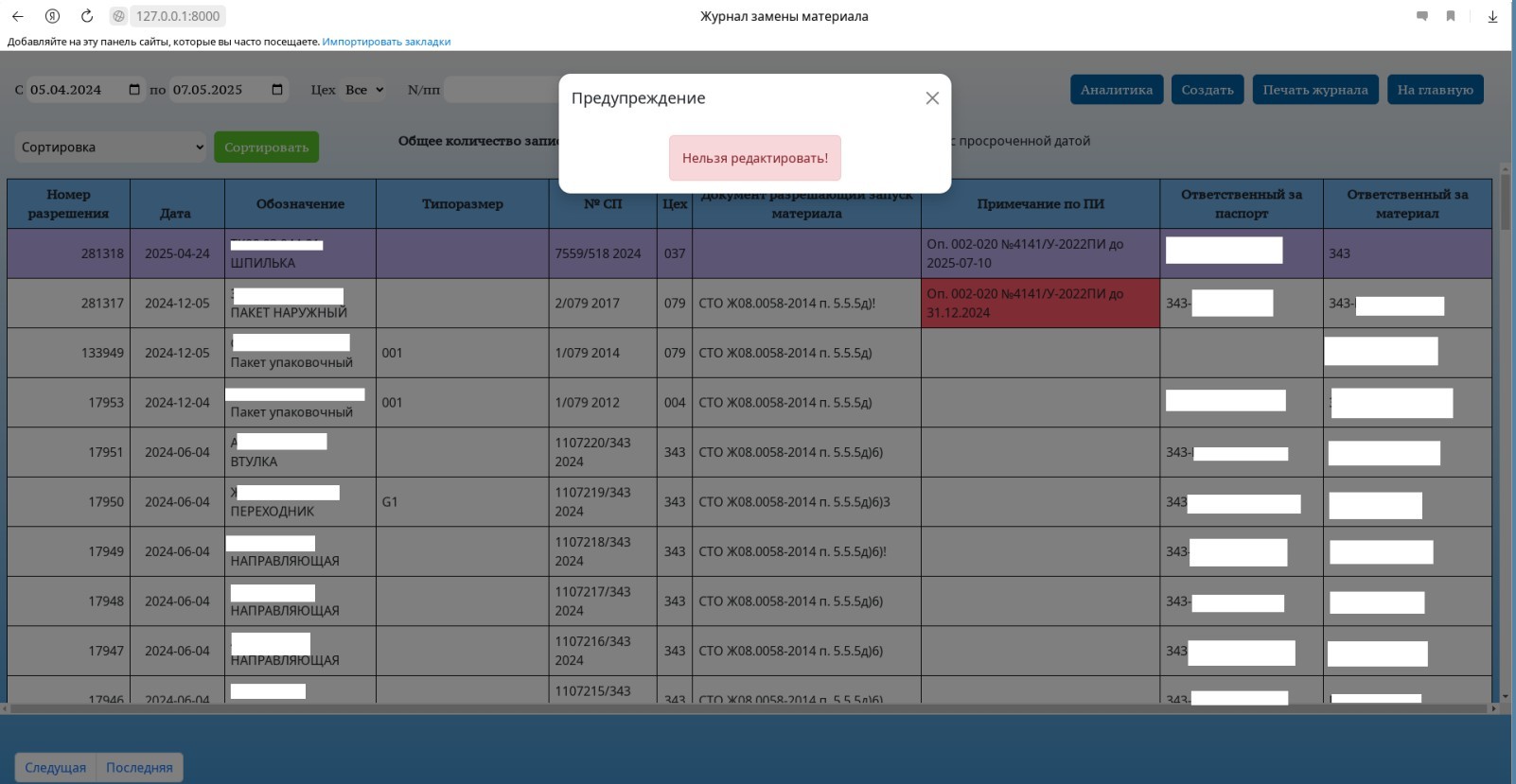
Но некоторые журналы учета разрешений нельзя менять, так как паспорт связанный с ними уже распечатан (см. рисунок 3.6).

Рисунок 3.6 — Предупреждение

При нажатии на кнопку «Аналитика» откроется новая страница, на которой выводится диаграмма процента паспортов с заменой материала по отношению к общему количеству СП (см. рисунок 3.7).

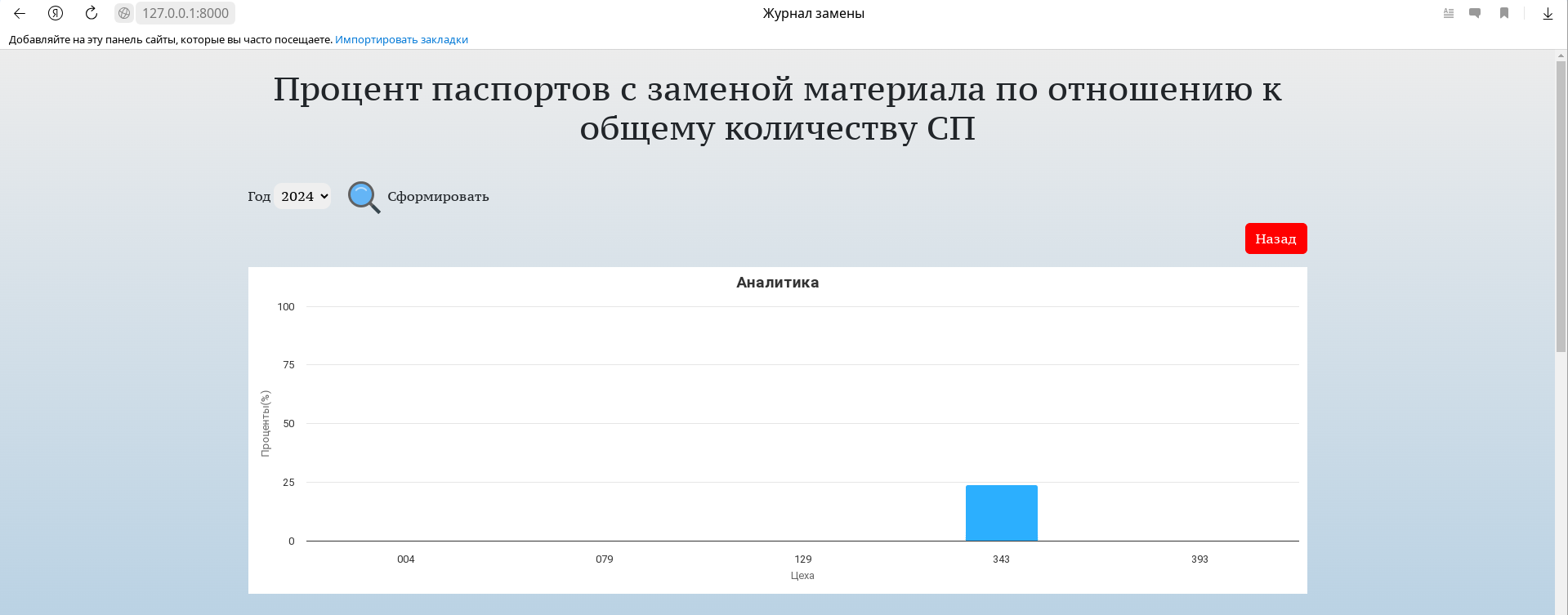


Рисунок 3.7 — Диаграмма СП с заменой материала по отношению к общему количеству в процентах

Также на этой странице есть аналитика процент паспортов с заменой

материала по отношению к общему количеству паспортов по цеху (см. рисунок 3.8).

Рисунок 3.8 — Аналитика СП с заменой материала по отношению к общему количеству в процентах

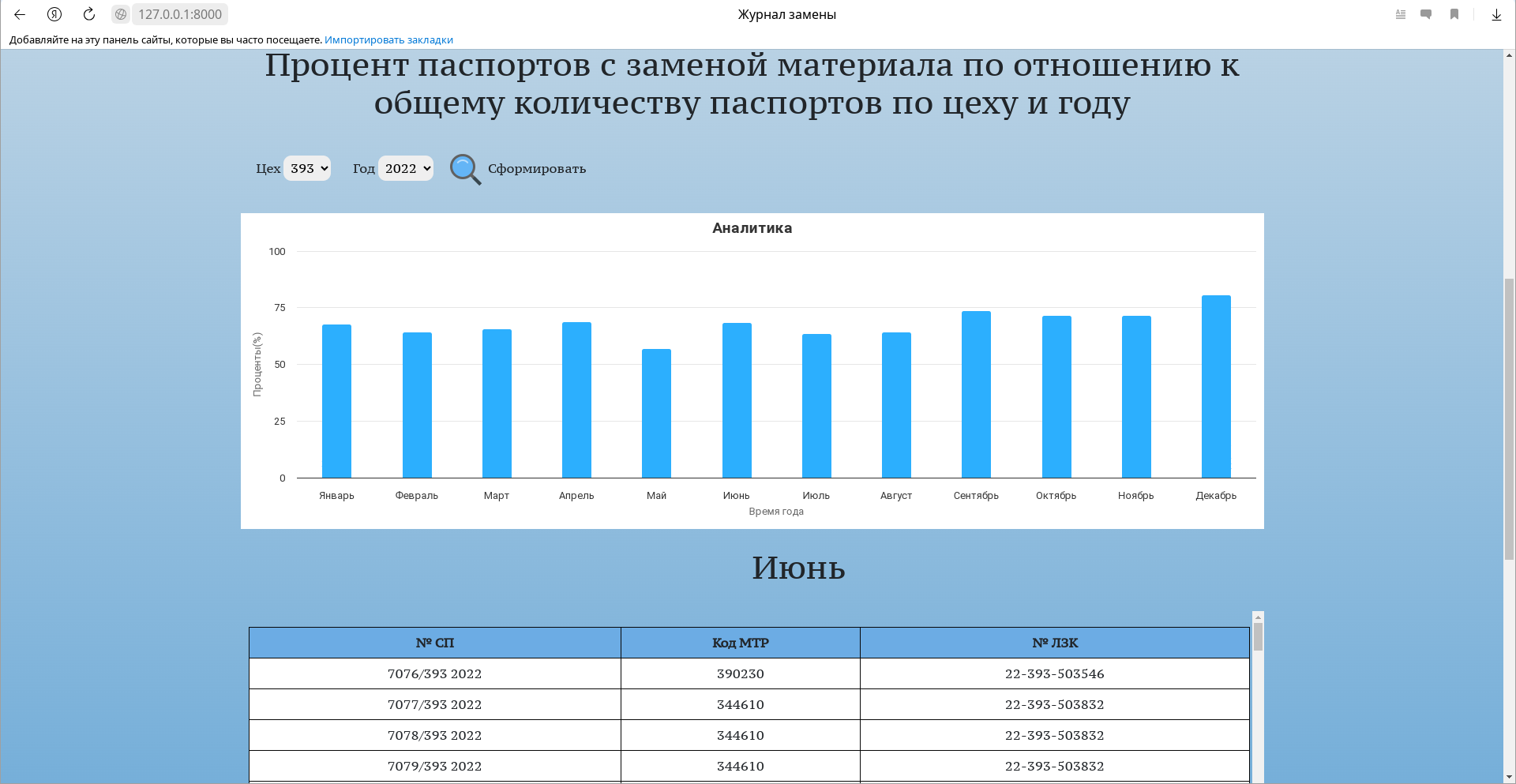
После наведения на определенный столбец, снизу появится таблица с кодом МТР, номере ЛЗК и номер СП за определенный месяц(рисунок 3.9).

Рисунок 3.9 — Список МТР, ЛЗК и номера СП

При нажатии на кнопку «Печать журнала» на основной странице, формируется PDF файл, в котором будет вся таблица журнала учета замены материала (см. рисунок 3.10).



Рисунок 3.10 — ПДФ документ с данными журнала учета замены

## Планируемый эффект от внедрения

Внедрение программного модуля «Ведение в ИС МКАДД журнала учёта разрешений замены материала» относится к переходу на импортонезависимое программное обеспечение [14]. Ключевые ожидаемые эффекты от внедрения включают:

* + 1. Повышение эффективности учета замены материалов:
       1. Модуль позволит создать централизованное хранение данных о заменах материалов, избавляя от необходимости ведения и учета бумажных материалов;
       2. Благодаря возможности оперативно добавлять и редактировать записи журнала замены, процесс учета замены материала значительно

сократится;

* + 1. Гибкость работы с данными:
       1. Фильтрация и сортировка даст возможность сотрудникам оперативно находить нужные записи по различным критериям;
       2. Интуитивно понятный интерфейс упростит взаимодействие с системой;
    2. Автоматизация аналитики и отчетности:
       1. Встроенные инструменты аналитики предоставят возможность автоматического формирования отчетов заменам материалов, выявления тенденций и статистики использования;
       2. Позволят визуализировать данные для аналитической работы;
    3. Снижение ошибок и повышение точности данных:
       1. Минимизация человеческого фактора на валидность при создании журнала происходит проверка на правильность записи сопроводительного паспорта, также при выполнении фильтрации происходит проверка на правильность выбора дат, что позволит снизить количество ошибок и неточностей в учете;
    4. Поддержка импортонезависимых решений:
       1. Модуль совместим с отечественными операционными системами Astra Linux, что соответствует перехода на импортонезависимое ПО;
       2. Веб-интерфейс обеспечит доступ к системе с различных устройств, что позволит сотрудникам работать из любого места и сократит время на доступ к информации.

Внедрение программного модуля позволит перейти на качественно новый уровень учета замены материала. Автоматизация процессов, удобство работы с данными и аналитические возможности модуля обеспечат повышение

производительности, снижение затрат и улучшение контроля над операциями.

Дополнительным преимуществом станет соответствие требованиям по использованию отечественного ПО, что особенно актуально в условиях импортозамещения.

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

1. Разработано руководство пользователя;
2. Проведена оценка планируемого эффекта от внедрения программы

«Ведение в ИС МКАДД журнала учёта разрешений замены материала»

# ЗАКЛЮЧЕНИЕ

Поставленная цель была достигнута. Был разработан программный модуля для сопровождения ведения журнала учета разрешений замены материала в ИС МКАДД.

В течение выполнения выпускной квалификационной работы выполнены такие задачи, как:

1. Проведен комплексный анализ предметной области, включая изучение терминологии, инструментальных средств и характеристик целевой аудитории.
2. Состоялось знакомство с ФГУП «Комбинат «Электрохимприбор» с акцентом на ключевые бизнес-процессы. На основании проведенного анализа разработаны: модель бизнес-процессов, информационная модель «сущность- связь».
3. Разработаны функции для корректной работы приложения с предоставленной базы данных.
4. Создано импортонезависимое прикладное приложение с использованием веб-технологии, имеющее все необходимые бизнес-функции.
5. Проведены тестирования функционала приложения и проведена оптимизация работы приложения. Приложение полностью работоспособное и отвечает необходимым требованиям к разрабатываемой системе.
6. Написано руководство пользователя с пошаговой инструкцией
7. Проведен анализ планируемого эффекта от внедрения информационной системы.

Выполнение выпускной квалификационной работы можно считать успешным, так как все поставленные цели и задачи были выполнены.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. 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 BООKS : [сайт]. — URL: https://www.iprbооkshоp.ru/10682.html (дата обращения: 2.04.2024).
2. 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 c. — ISBN 978-5-394-05121-0. — Текст : электронный

// IPR SMART : [сайт]. — URL: https://[www.iprbookshop.ru/124097.html](http://www.iprbookshop.ru/124097.html) (дата обращения: 2.05.2024).

1. Бизнес-процессы промышленного предприятия : учебное пособие / Н.Р. Кельчевская [и др.]. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2016. — 340 c. — ISBN 978-5-7996-1824-7. —

Текст : электронный // IPR SMART : [сайт]. — URL: https://[www.iprbookshop.ru/68423.html](http://www.iprbookshop.ru/68423.html) (дата обращения: 30.04.2024).

1. Бизнес-процессы: языки моделирования, методы, инструменты / Ф. Шенталер [и др.]. — Москва: Альпина Паблишер, 2019. — 264 c. — ISBN 978-5-9614-2022-7. — Текст: электронный // IPR SMART : [сайт]. — URL: https://[www.iprbookshop.ru/124474.html](http://www.iprbookshop.ru/124474.html) (дата обращения: 21.04.2024).
2. Виафоре, П. Надежный Python / П. Виафоре: БХВ, 2023. – 352 с. – ISBN 978-5-9775-1174-2.
3. Гладышев Д. А. Разработка баз данных в PostgreSQL: управление данными, хранимые процедуры. – СПб.: БХВ-Петербург, 2018. – 224 с.
4. Иванов Д. Б. Разработка и оптимизация материализованных представлений в PostgreSQL. – СПб.: БХВ-Петербург, 2019. – 192 с.
5. Иньегски Т. Создание адаптивного интерфейса: современные подходы к веб-разработке / Т. Иньегски, А. Тейлов: – Москва, 2018. – 382 с.
6. Логинов А. С. Разработка схем баз данных в PostgreSQL: практическое

руководство. – М.: Диалектика, 2019. – 240 с.

1. 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 BООKS : [сайт]. — URL: https://www.iprbооkshоp.ru/26456.html (дата обращения: 2.04.2025).
2. Меле, А. Django 2 в примерах / А. Меле ; перевод Д. В. Плотникова. — Москва : ДМК Пресс, 2019. — 408 c. — ISBN 978-5-97060-746-6. — Текст

: электронный // Цифровой образовательный ресурс IPR SMART : [сайт].

* + URL: https://[www.iprbookshop.ru/126199.html](http://www.iprbookshop.ru/126199.html) (дата обращения: 30.11.2022)

1. Шениг, Г. -Ю. PostgreSQL 11. Мастерство разработки / Г. -Ю. Шениг ; перевод А. А. Слинкин. — Москва : ДМК Пресс, 2019. — 352 c. — ISBN 978-5-97060-671-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://[www.iprbookshop.ru/125100.html](http://www.iprbookshop.ru/125100.html) (дата обращения: 20.10.2022)
2. Питер Пин-Шен Чен. Модель «сущность-связь» — шаг к единому представлению о данных : Журнал "Системы управления базами данных", #03, 1995 год : Издательство "Открытые системы"
3. Смекалов В.П. Анализ финансовой отчетности предприятия : учебное пособие / Смекалов В.П., Бадмаева Д.Г., Смолянинова С.В.. — Санкт- Петербург : Проспект Науки, 2024. — 471 c. — ISBN 978-5-903090-30-3.
   * Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://[www.iprbookshop.ru/80089.html](http://www.iprbookshop.ru/80089.html) (дата обращения: 17.06.2024)
4. Архитектуры и топологии многопроцессорных вычислительных систем : учебник / А.В. Богданов [и др.].. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 135

c. — ISBN 978-5-4497-2443-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://[www.iprbookshop.ru/133923.html](http://www.iprbookshop.ru/133923.html) (дата обращения: 17.06.2024)

1. Стасышин, В. М. Базы данных. Лекции по курсу. В 4 частях. Ч.1 : учебное пособие / В. М. Стасышин, Т. Л. Стасышина. — Новосибирск : Новосибирский государственный технический университет, 2021. — 79 c.

— ISBN 978-5-7782-4544-0 (ч.1), 978-5-7782-4543-3. — Текст :

электронный // Цифровой образовательный ресурс IPR SMART : [сайт].

* + URL: https://[www.iprbookshop.ru/126545.html](http://www.iprbookshop.ru/126545.html) (дата обращения: 24.06.2023)

1. Савельев, А. О. HTML5. Основы клиентской разработки : учебное пособие / А. О. Савельев, А. А. Алексеев. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 270 c. — ISBN 978-5-4497-2398-7. — Текст :

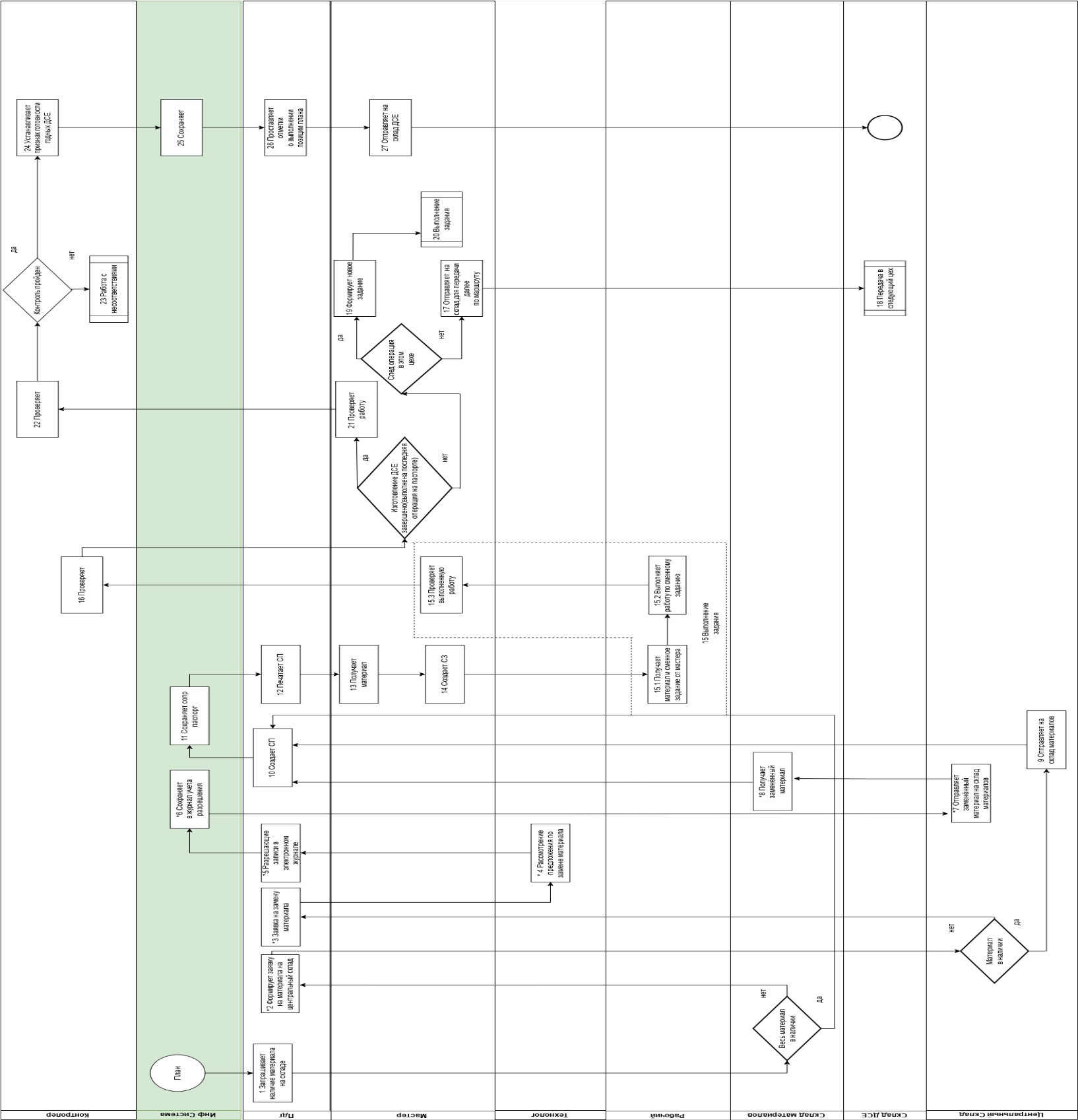
электронный // Цифровой образовательный ресурс IPR SMART : [сайт].

* + URL: https://[www.iprbookshop.ru/133910.html](http://www.iprbookshop.ru/133910.html) (дата обращения: 30.10.2023)

1. Документация по библиотеке django-rest framework: официальный сайт. – Обновляется в течение суток. – URL: https://django-rest-framework.org (дата обращения: 23.05.2024). – Текст: электронны
2. Смирнов Д. В. Разработка пользовательских функций в PostgreSQL. – СПб.: Питер, 2018. – 240 с.
3. Флэнаган Д. JavaScript. Полное руководство / Д. Флэнаган: ДиалектикаВильямис, 2021. – 720 с. – ISBN 978-5-907203-79-2. //

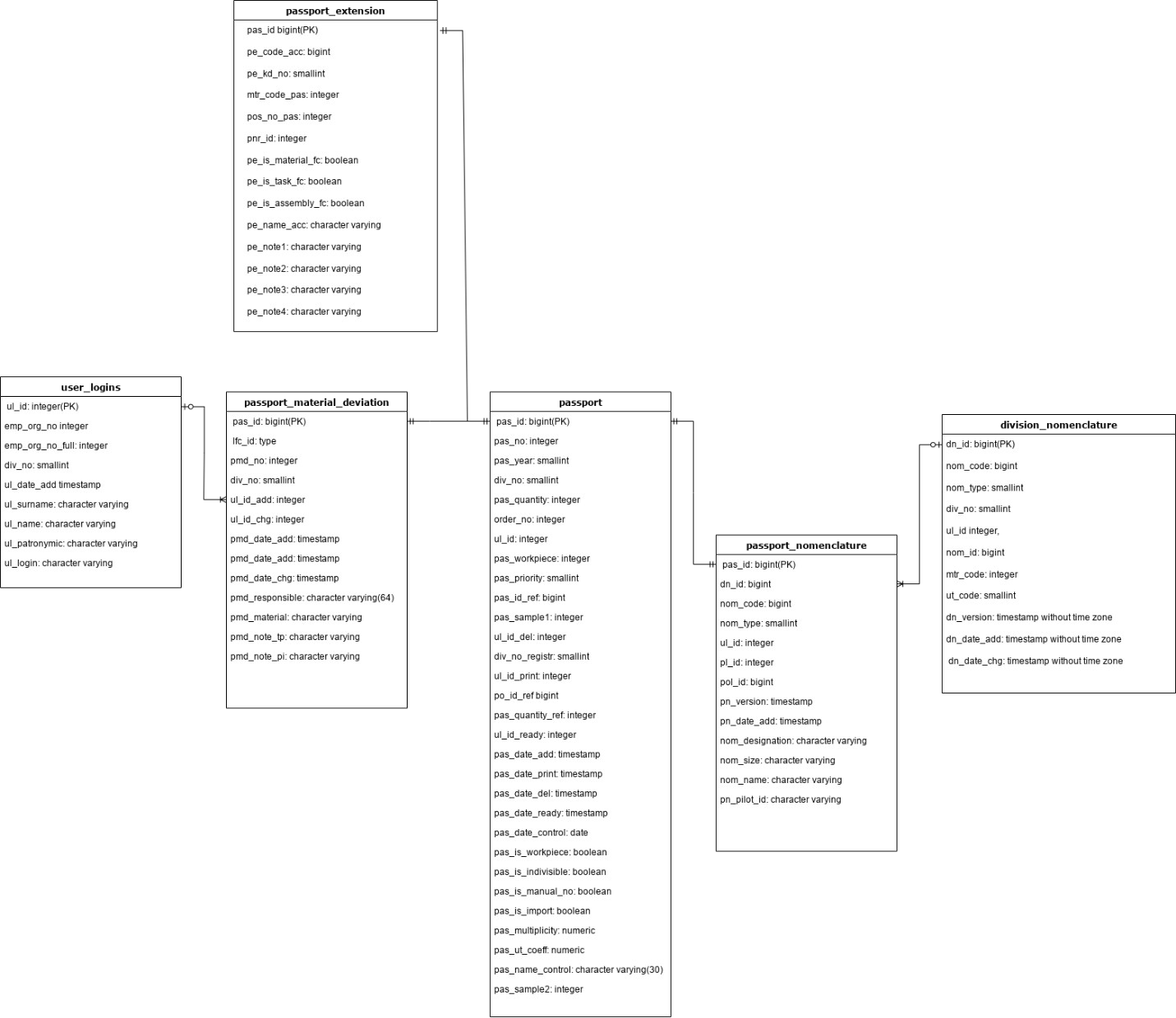
**Приложение A**

**(обязательное) Бизнес-процесс**

****

## Приложение Б

## Физическая модель БД для отображения данных (обязательное)

****

**Приложение В (обязательное)**

**Код файла views.py**

В.1 Функция для вывода данных для журнала учёта разрешений замены материала на страницу

def get\_journal(request): global sl

up = request.GET.get('up') divs = []

if request.session.session\_key is None: request.session.create()

session\_key = request.session.session\_key if str(session\_key) not in sl:

sl[str(session\_key)] = {

'date\_end': str(datetime.now().date() + timedelta(days=1)), 'date\_start' : str(datetime.now().date() - timedelta(days=30)), 'div\_no' : 0,

'pmd\_no' : 0,

'result' : 0,

}

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT DISTINCT div\_no FROM mkadd.passport\_material\_deviation''',) # fetchall

divs = cursor.fetchall() divs = sorted([i[0] for i in divs]) if up != None:

sl[str(session\_key)]['result'] = 0

sl[str(session\_key)]['date\_end'] = str(datetime.now().date() + timedelta(days=1)) sl[str(session\_key)]['date\_start'] = str(datetime.now().date() - timedelta(days=30)) sl[str(session\_key)]['div\_no'] = 0

if request.method == "POST":

sl[str(session\_key)]['date\_start'] = request.POST.get('date\_start') sl[str(session\_key)]['date\_end'] = request.POST.get('date\_end') sl[str(session\_key)]['div\_no'] = request.POST.get('div\_no') sl[str(session\_key)]['pmd\_no'] = request.POST.get('numberpunkt') sl[str(session\_key)]['result'] = 0

try:

os.remove('static/print.pdf') except Exception as e:

print(e)

if len(str(sl[str(session\_key)]['pmd\_no'])) == 0: sl[str(session\_key)]['pmd\_no'] = 0

if request.GET.get('tests') != None: sl[str(session\_key)]['result'] = 0

slov = json.dumps({

'pmd\_no': sl[str(session\_key)]['pmd\_no'], 'date\_start': sl[str(session\_key)]['date\_start'], 'date\_end': sl[str(session\_key)]['date\_end'], 'department': sl[str(session\_key)]['div\_no'],

})

sortdate = request.GET.get('sortdate') if sl[str(session\_key)]['result'] == 0:

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT mkadd.journal\_accounting\_permissions(%s)''', [slov])

# fetchall

sl[str(session\_key)]['result'] = cursor.fetchall()

sl[str(session\_key)]['result'] = list(sl[str(session\_key)]['result'][0])[0]['dataset'] page\_number = request.GET.get('page')

dw = sl[str(session\_key)]['result'] if page\_number == None:

page\_number = 0

if sl[str(session\_key)]['result'] == None:

context = {

'page\_obj': sl[str(session\_key)]['result'], 'date\_start': sl[str(session\_key)]['date\_start'], 'date\_end': sl[str(session\_key)]['date\_end'], 'div\_no': sl[str(session\_key)]['div\_no'], 'pmd\_no': sl[str(session\_key)]['pmd\_no'], 'page\_number': page\_number,

'divs': divs,

}

else:

if sortdate != None and sortdate == 'True':

sl[str(session\_key)]['result'] = sorted(sl[str(session\_key)]['result'], key=lambda x: datetime.strptime(x['pmd\_date\_add'], "%Y-%m-%d"), reverse=True)

if sortdate != None and sortdate == 'False':

sl[str(session\_key)]['result'] = sorted(sl[str(session\_key)]['result'], key=lambda x: datetime.strptime(x['pmd\_date\_add'], "%Y-%m-%d"), reverse=False)

elif sortdate != None and sortdate == 'PI':

sl[str(session\_key)]['result'] = sorted(sl[str(session\_key)]['result'], key=lambda x: (x['pmd\_note\_pi'] == None or x['pmd\_note\_pi'] == ''))

elif sortdate != None and sortdate == 'doc':

sl[str(session\_key)]['result'] = sorted(sl[str(session\_key)]['result'], key=lambda x: (x['pmd\_note\_tp'] is None or x['pmd\_note\_tp'] == ''))

# else:

# sl[str(session\_key)]['result'] = sorted(sl[str(session\_key)]['result'], key=lambda x: datetime.strptime(x['pmd\_date\_add'], "%Y-%m-%d"), reverse=True)

paginator = Paginator(sl[str(session\_key)]['result'], 100)

# print(sl[str(session\_key)]['result'][1]['pmd\_note\_pi'] == None) page\_number = request.GET.get('page')

page\_obj = paginator.get\_page(page\_number) if page\_number == None:

page\_number = 0 context = {

'page\_obj': page\_obj,

'date\_start': sl[str(session\_key)]['date\_start'], 'date\_end': sl[str(session\_key)]['date\_end'], 'div\_no': sl[str(session\_key)]['div\_no'], 'pmd\_no': sl[str(session\_key)]['pmd\_no'], 'page\_number': page\_number,

'len': len(sl[str(session\_key)]['result']), 'divs': divs,

}

if request.GET.get('download') == 'True':

table\_data = [ ['№ пп','Дата','Обозначение','Типоразмер','№ СП','Цех','Документ разрешающий \n запуск материала',

'Ответственный \n за паспорт', 'Ответственный \n за материал'

]]

for data in dw: arr = []

arr.append(data['numberpunkt']) arr.append(data['pmd\_date\_add'])

if data['nomenclaturemeans'] != None and len(data['nomenclaturemeans']) > 17:

data['nomenclaturemeans'] = data['nomenclaturemeans'] [:len(data['nomenclaturemeans'])//2] + '\n' + data['nomenclaturemeans'] [len(data['nomenclaturemeans'])//2:]

arr.append(data['nomenclaturemeans'])

if data['type\_size'] != None and len(data['type\_size']) > 17:

data['type\_size'] = data['type\_size'][:len(data['type\_size'])//2] + '\n' + data['type\_size'][len(data['type\_size'])//2:]

if data['pmd\_note\_tp'] != None and len(data['pmd\_note\_tp']) > 34: data['pmd\_note\_tp'] = data['pmd\_note\_tp'][:len(data['pmd\_note\_tp'])//2] + '\n'

+ data['pmd\_note\_tp'][len(data['pmd\_note\_tp'])//2:] arr.append(data['type\_size']) arr.append(data['fulnumpas']) arr.append(data['passportdivision']) arr.append(data['pmd\_note\_tp']) arr.append(data['ul\_login']) arr.append(data['pmd\_responsible']) table\_data.append(arr)

pdfmetrics.registerFont(TTFont('Arial','Arial.ttf')) row\_h = [40]\* len(table\_data)

pdf = SimpleDocTemplate('static/print.pdf', pagesize=landscape(A4))

table = Table(table\_data, colWidths=[35,55,100,100,100,50,200,100,95],

rowHeights=row\_h)

style = TableStyle([

('BACKGROUND',(0,0),(-1,0),colors.grey),

('TEXTCOLOR',(0,0),(-1,0),colors.whitesmoke),

('ALIGN',(0,0),(-1,-1),'CENTER'),

('FONTNAME',(0,0),(-1,-1),'Arial'),

('GRID',(0,0),(-1,-1),1,colors.black),

('VALGIN',(0,0),(-1,-1),'MIDDLE')

])

table.setStyle(style)

styles = getSampleStyleSheet() title\_st = styles['Heading1'] title\_st.fontName = 'Arial' title\_st.alignment = 1

title = Paragraph(f'Журнал учета разрешений c {context["date\_start"]} по

{context["date\_end"]}',title\_st) elements = [title, table]

pdf.build(elements, onFirstPage=add\_page, onLaterPages=add\_page)

return render(request, 'diplom/journal.html', context=context)

В.2 Функция для создания новой записи в журнале учёта разрешений замены материала

def create\_journal(request, page): if page == 0:

page = 1

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT DISTINCT div\_no FROM mkadd.production\_point

WHERE pp\_is\_prod = true and pp\_is\_auto\_receiving=false and

pp\_is\_disable=false

ORDER BY div\_no ''',)

divs = [i[0] for i in cursor.fetchall()] if request.session.session\_key is None:

request.session.create()

session\_key = request.session.session\_key if request.method == 'POST':

form = JournalForm(request.POST) if form.is\_valid():

try:

num\_sp = request.POST.get('num\_sp').split('/')

to = num\_sp[1].split()

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT pas\_id FROM mkadd.passport

WHERE pas\_no =%s and pas\_year=%s and div\_no=%s ''',

[int(num\_sp[0]),int(to[1]), int(to[0])]) idpasport = cursor.fetchall()

if len(idpasport) > 0: idpasport = idpasport[0][0]

pmd\_note\_tp = request.POST.get('pmd\_note\_tp') or None pmd\_note\_pi = request.POST.get('pmd\_note\_pi') or None date\_pi = request.POST.get('date\_pi') or None

if pmd\_note\_pi != None: pmd\_note\_tp = None

slov = json.dumps({

'div\_no': request.POST.get('div\_no'),

# 'pmd\_no': request.POST.get('num\_punct'), 'pmd\_material': request.POST.get('pmd\_material'), 'pmd\_note\_tp': pmd\_note\_tp,

'pmd\_note\_pi': pmd\_note\_pi,

'ul\_login': request.POST.get('ul\_login'), 'pmd\_responsible': request.POST.get('pmd\_responsible'), 'IDPasport': idpasport,

'date\_pi' : date\_pi,

},ensure\_ascii=False)

# if slov['pmd\_note\_pi'] != None:

# slov['pmd\_note\_tp'] = None

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT mkadd.insert\_deviere\_mtr(%s)''', [slov])

result = cursor.fetchall() except Exception as e:

form = JournalForm(request.POST) eror = ''

print(str(e))

if str(e) == 'list index out of range': eror = ' № СП необходимо писать с "/"'

elif str(e).split(':')[1].replace('DETAIL','').split('"')[0] == ' повторяющееся значение ключа нарушает ограничение уникальности ':

eror = 'данный паспорт уже используется в системе , нарушение

уникальности'

else:

eror =str(e).split(':')[1].replace('DETAIL','') context = {

'form': form,

'eror': True,

'page': page,

'erors': eror, 'divs': divs

}

return render(request,'diplom/create\_journal.html',context=context) url = reverse('diplom:journal')

ur = request.build\_absolute\_uri(url) requests.get(url=ur, cookies={'sessionid': session\_key}) return redirect(f'{url}?sortdate=True')

return render(request,'diplom/create\_journal.html',context=context) form = JournalForm()

context = {

'form': form,

'eror': False,

'page': page, 'divs': divs

}

return render(request,'diplom/create\_journal.html',context=context)

В.3 Функция для обновления записи в журнале учёта разрешений замены материала

def update\_journal(request,pas\_id,numberpunkt, page): if request.session.session\_key is None:

request.session.create()

session\_key = request.session.session\_key slov = json.dumps({

'idpasport': pas\_id,

})

result = 0

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT mkadd.journal\_permissions(%s)''', [slov])

result = cursor.fetchall()

result = list(result[0])[0]['dataset'][0] if page == 0:

page = 1 context = {

'result': result, 'eror': False,

'numberpunkt': numberpunkt, 'page': page,

}

if request.method == 'POST':

slov = json.dumps({

'pmd\_note\_tp': request.POST.get('pmd\_note\_tp'), 'pmd\_note\_pi': request.POST.get('pmd\_note\_pi'), # 'ul\_login': request.POST.get('ul\_login'),

# 'pmd\_responsible': request.POST.get('pmd\_responsible'), 'idpasport': pas\_id,

},ensure\_ascii=False) try:

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT mkadd.update\_journal(%s)''', [slov])

result = cursor.fetchall() url = reverse('diplom:journal')

ur = request.build\_absolute\_uri(url) requests.get(url=ur,params={'tests':'tests'}, cookies={'sessionid': session\_key}) return redirect(f'{url}?page={page}#{numberpunkt}')

except Exception:

context['eror'] = True

return render(request,'diplom/update\_journal.html',context=context) return render(request,'diplom/update\_journal.html',context=context)

В.4 Функция для аналитики

def analytics(request):

global mas\_div data = [] data1 = []

data2\_pas = [] data3\_pasj = []

yaer\_one = request.GET.get('year') if yaer\_one is None:

yaer\_one = 2024

div\_no = 0

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT DISTINCT div\_no FROM mkadd.passport\_material\_deviation''',) # fetchall

div\_no = cursor.fetchall()

div\_no = sorted([int(i[0]) for i in div\_no]) for div in div\_no:

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT COUNT(ps.div\_no) AS "main", COUNT(pasmater.pas\_id) FROM

mkadd.passport as ps

[div, yaer\_one])

LEFT JOIN mkadd.passport\_material\_deviation as pasmater ON pasmater.pas\_id = ps.pas\_id

WHERE ps.div\_no = %s and ps.pas\_date\_del is null and ps.pas\_year=%s ;''',

# fetchall

result = cursor.fetchall()

data.append(result[0][0]) data1.append(result[0][1])

year = [i for i in range(2015,2025)]

month = 'Январь Февраль Март Апрель Май Июнь Июль Август Сентябрь Октябрь Ноябрь Декабрь'.split()

if request.method == "POST":

divs\_no = request.POST.get('div\_no') years = request.POST.get('year') if divs\_no is None:

divs\_no = mas\_div[0] if years is None:

years = mas\_div[1]

year\_mt = [i for i in range(1,13)] mas\_div[0] = divs\_no

mas\_div[1] = years

for div in year\_mt:

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT COUNT(pas\_id) FROM mkadd.passport

WHERE EXTRACT(MONTH FROM pas\_date\_add) IN (%s) and pas\_year=%s and div\_no=%s;''',[div,mas\_div[1],mas\_div[0]])

result = cursor.fetchall() data2\_pas.append(result[0][0])

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT COUNT(pas\_id) FROM mkadd.passport\_material\_deviation WHERE div\_no = %s and EXTRACT(YEAR FROM pmd\_date\_add) = %s and

EXTRACT(MONTH FROM pmd\_date\_add) IN (%s);''',[mas\_div[0],mas\_div[1],div])

results = cursor.fetchall() data3\_pasj.append(results[0][0])

for i in range(len(div\_no)):

if len(str(div\_no[i])) == 1: div\_no[i] = '00'+ str(div\_no[i])

elif len(str(div\_no[i])) == 2: div\_no[i] = '0'+ str(div\_no[i])

else:

div\_no[i] = str(div\_no[i]) context = {

'div\_no': div\_no, 'divs\_no': mas\_div[0], 'years': mas\_div[1], 'data': data,

'data1': data1, 'year': year, 'month': month, 'yaer\_one': yaer\_one,

'data2\_pas': data2\_pas, 'data3\_pasj': data3\_pasj

}

return render(request,'diplom/analytics.html',context=context)

В.4 Класс API для аналитики

class AnalytAPI(APIView):

def get(self,request,months,pas\_year,div\_no):

month = 'Январь Февраль Март Апрель Май Июнь Июль Август Сентябрь Октябрь Ноябрь Декабрь'.split() sl = {}

for i in range(len(month)): sl[month[i]] = i + 1

slov = json.dumps({ 'pas\_date\_add': sl[months], 'pas\_year': pas\_year, 'div\_no': div\_no,

})

with connections['miha'].cursor() as cursor: cursor.execute('''

SELECT mkadd.analytic\_jour(%s)''', [slov])

# fetchall

data = cursor.fetchall()

data = data[0][0]['dataset'][0:100] for i in data:

while len(str(i['mtr\_code']))!= 7:

i['mtr\_code'] = '0' + str(i['mtr\_code']) return Response(data)

**Приложение Г (обязательное)**

**Код Django шаблонов**

Г.1 Содержание файла родительского шаблона

{% load static %}

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="utf-8">

<title>Журнал замены материала</title>

<link rel="icon" href="{% static 'img/loading.ico' %}" type="image/x-icon">

<link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.min.css' %}">

<script src="/static/js/bootstrap.bundle.min.js"></script>

<script src="/static/js/accessibility.js"></script>

<script src="/static/js/highcharts.js"></script>

</head>

<style> body {

margin: 0;

height: 100vh;

background: linear-gradient(to bottom, #ececec, #4495D1); background-repeat: no-repeat;

}

table

{

border-collapse: collapse; width: 99%;

position: relative;

}

th, td

{

border: 1px solid #070707;

padding: 7px; text-align: left;

text-align: center; margin-left: center;

}

th

{

background-color: #6CACE4;

font-family: 'PT Serif Caption';

}

tr{

background-color: #ffffff;

}

thead th{

position: sticky; top: 0;

z-index: 10;

box-shadow: 0 2px 2px -1px rgba(0,0,0,00.00001px);

}

tr:hover{

background-color: #b8aae9;

transition: background-color 02.s ease;

}

.table-cont{

max-height: 700px; overflow-y: auto;

}

.bg-dangers{

background:rgba(255, 0, 25, 0.650);

}

.sq{

width: 20px; height: 20px;

background:rgba(255, 0, 25, 0.650); display: flex;

justify-content: center; align-items: center; margin-left: 20%;

}

.btn-success:hover{

transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

}

.btn:hover{

transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

}

.btn-primary:hover {

background-color: #2980b9 !important; transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

}

@keyframes spin {

0% { transform: rotate(0deg); } 100% { transform: rotate(360deg); }

}

#loading-icon {

animation: spin 1.5s linear infinite;

}

#loadingIndicator{ position: fixed; top: 50%;

left: 50%;

transform: translate(-50%, -50%);

z-index: 1000; padding: 20px; border-radius: 10px;

}

.btn-primary {

background-color: #025EA1 !important; border-color: #025EA1 !important; font-family: 'PT Serif Caption';

font-size: 16px;

}

.btn-danger {

background-color: #FF0000 !important; border-color: #FF0000 !important;

}

.btn-success {

background-color: #56C02B !important; border-color: #56C02B !important;

}

.high{

background-color: ;

}

#a{

font-family: 'PT Serif Caption'; font-size: 16px;

}

#arrow{

margin-left: 5px; font-size: 1em;

}

.chart-container {

position: relative; width: 100%;

max-width: 800px; margin: auto;

}

.bar {

fill: rgba(54, 162, 235, 0.8); transition: fill 0.3s ease;

}

.bar:hover {

fill: orange;

}

.chart-title {

font-size: 18px; font-weight: bold; text-align: center; margin-bottom: 10px;

}

</style>

<body>

<!-- Основное содержимое страницы -->

<main>

{% block content %}

{% endblock %}

</main>

</body>

</html>

Г.2 Содержимое файла дочернего шаблона index.html

{% extends 'base.html' %}

{% load static %}

{% block content%}

<header class="navbar navbar-expand-lg navbar-light bg-light md-4">

<div class="container-fluid">

<img src="{% static 'img/loading.ico' %}" height="40">

<div class="d-flex align-items-center">

<p>{{user}}</p>

</div>

</div>

</header>

<main style="max-width: 1200px; margin: 0 auto; padding: 40px 20px;">

<div style="text-align: center; margin-bottom: 50px;">

<h1 class="mov" style="font-family: PT Serif Caption; font-size: 2.5rem; max-width: 700px;margin: 0 auto 30px; line-height: 1.5;">

«Ведение в ИС МКАДД журнала учёта разрешений замены материала»</h1>

<p class="mov" style="font-family: PT Serif Caption; font-size: 1.2rem; max-width: 700px;margin: 0 auto 30px; line-height: 1.5">

Данное веб-приложение для ведения в ИС МКАДД журнала учёта разрешений замены

материала

</p>

<div style="display: flex; justify-content: center; gap: 20px; margin-top: 40px;">

<a href={%url 'diplom:journal' %} class="btn btn-primary" style="padding: 12px 30px; font-size: 1.1rem; border-radius: 4px; background-color: #3498db; border: none; transition: all 0.3s;">

<i class="fas fa-book-open" style="margin-right: 8px;"></i> Журнал учёта

</a>

<a href={% url "diplom:analytics" %} class="btn btn-primary" style="padding: 12px 30px; font-size: 1.1rem; border-radius: 4px; border: 2px solid #3498db; background: transparent; transition: all 0.3s;">

<i class="fas fa-chart-bar" style="margin-right: 8px;"></i> Аналитика

</a>

</div>

</div>

</main>

<style>

.btn-primary:hover {

background-color: #2980b9 !important; transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

}

.btn-outline-primary:hover { background-color: #f8f9fa !important; transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(0,0,0,0.1);

}

@media (max-width: 768px) { h1 {

font-size: 2rem !important;

}

}

</style>

{%endblock%}

Г.3 Содержимое файла дочернего шаблона journal.html

{% extends 'base.html' %}

{%load filters%}

{% load static %}

{% block content%}

<div style='display:flex;'>

<form method="post" style="margin-top: 20px;margin-left: 20px;" id="a">

{% csrf\_token %}

С <input type="date" id="date\_start" name="date\_start" style="border-radius:10px; padding: 5px; box-shadow: none; outline: none;border: none ;" value="{{date\_start}}">

по <input type="date" id="date\_end" name="date\_end" style="border-radius:10px; padding: 5px; box-shadow: none; outline: none; border: none;" value="{{date\_end}}">

&nbsp &nbsp &nbsp Цех <select style="border-radius:10px;padding: 5px; box-shadow: none; outline: none; border: none" id="div\_no" name="div\_no">

<option value="" disabled selected>{{div\_no}}</option>

{% for i in divs%}

<option value="{{i}}">{{i| zero}}</option>

{%endfor%}

<option value="0">Все</option>

</select>

<!-- все <input type="checkbox" id="div\_all" name="div\_all"> -->

{%if pmd\_no == 0 %}

&nbsp &nbsp &nbsp N/пп <input id="numberpunkt" type="number" style="border- radius:10px; padding: 5px; box-shadow: none; outline: none;border: none ;" name="numberpunkt">

{%else%}

N/пп <input id="numberpunkt" type="number" style="border-radius:10px; padding: 5px; box-shadow: none; outline: none;border: none ;" name="numberpunkt" value="{{pmd\_no}}">

{%endif%}

<button onclick="search()" id='a' class="btn" type="submit" style="margin- left:0.00001px;" id="search"> <img src={% static 'img/search.png' %}> Поиск</button>

</form>

<div style="margin-left:430px;margin-top:30px">

<a class="btn btn-primary" href={% url "diplom:analytics" %} onclick="show\_loading(true)">Аналитика</a>

</div>

<div style="margin-left:10px;margin-top:30px; margin-right: 1px;">

<a id='a' class="btn btn-primary" href={% url "diplom:create" page\_number

%}>Создать</a>

</div>

<div style="margin-left:10px;margin-top:30px; margin-right: 1px;">

<button id='a' id="downl" class="btn btn-primary" onclick="downloadPDF()">Печать журнала</button>

</div>

<div style="margin-left:10px;margin-top:30px; margin-right: 1px;">

<a id='a' class="btn btn-primary" href={% url "diplom:main" %}>На главную</a>

</div>

</div>

<div style="margin-top: 20px;margin-left: 20px; display: flex;">

<select style="border-radius:10px;padding: 5px; box-shadow: none; outline: none; border: none" id="sort" name="sort">

<option disabled selected>Сортировка</option>

<option value="1" >По дате(от новой к старой)</option>

<option value="2" >По дате(от cтарой к новой)</option>

<option value="3" >По ПИ</option>

<option value="4" >По документу</option>

</select>

<button id='a' onclick="Sort()" class="btn btn-success" style="margin-left: 10px;"> Сортировать </button>

<p style="margin-left: 100px;font-weight: bold;" id="a"> Общее количество записей

{{len}}</p>

<div class="sq"> </div > <p>ПИ с просроченной датой</p>

</div>

<!-- </div> -->

</div>

{%if page\_obj%}

<div class="table-cont">

<table id="myTable" style="margin-top: 20px; margin-left: 10px; margin-right: 10px;position: sticky" >

<thead>

<tr >

<th>Номер разрешения</th>

<th >&nbsp &nbsp &nbsp &nbsp &nbsp &nbsp &nbsp &nbsp &nbsp &nbsp &nbsp &nbsp Дата

<span id="arrow"> </span></th>

<th>Обозначение</th>

<th>Типоразмер</th>

<th>№ СП</th>

<th>Цех</th>

<th>Документ разрешающий запуск материала</th>

<th>Примечание по ПИ</th>

<th>Ответственный за паспорт</th>

<th>Ответственный за материал</th>

</tr>

</thead>

<tbody>

{%for jur in page\_obj%}

<tr data-edit-url="/update/{{jur.pas\_id}}/{{jur.numberpunkt}}/{{page\_number}}">

<td style="text-align: right;">{{jur.numberpunkt |nones }}</td>

<td>{{jur.pmd\_date\_add |nones}}</td>

<td style="text-align: left;">{{jur.nomenclaturemeans |nones}} <br>{{jur.nom\_name

| nones}} </td>

{%if jur.type\_size %}

<td style="text-align: left;">{{jur.type\_size |nones}}</td>

{%else%}

<td></td>

{%endif%}

<td style="text-align: left;">{{jur.fulnumpas |nones}}</td>

<td>{{jur.passportdivision|zero |nones}}</td>

<td style="text-align: left;">{{jur.pmd\_note\_tp |nones }}</td>

<td style="text-align: left;" class="{% if jur.date\_pi|is\_date %} bg- dangers{%endif%}">{{jur.pmd\_note\_pi |nones}} </td>

<td style="text-align: left;">{{jur.ul\_login |nones }}</td>

<td style="text-align: left;">{{jur.pmd\_responsible |nones }}</td>

<!-- <td><div>

<a id="a" href="{%url 'diplom:update' jur.pas\_id jur.numberpunkt page\_number

%}" onclick="high(this)" class="btn btn-secondary" name="edits" type="button" data-bs-

toggle="modal" > Изменить</a>

</div></td> -->

</tr>

{% endfor %}

</tbody>

</table>

<div id="modalSheetShowMessage" class="modal show">

<div class="modal-dialog">

<div class="modal-content rounded-4 shadow">

<div class="modal-header border-bottom-0">

<h5 class="modal-title">Предупреждение</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria- label="Close"></button>

</div>

<div class="modal-footer flex-column border-top-0">

<div id="AlertMessage" class="alert alert-danger text-center fade show"></div>

</div>

</div>

</div>

</div>

</div>

<div id="loadingIndicator" class="text-center mt-3 d-none">

<img src="/static/img/loading.ico" id="loading-icon" width="64" height="64">

</div>

{%else%}

<h1 style="margin-top: 200px; margin-left: 40%; font-family: 'PT Serif Caption'; ">Данные не найдены</h1>

{%endif%}

<div style="margin-left: 20px">

{% include 'paginator.html' %}

</div>

Г.4 Содержимое файла дочернего шаблона create\_journal.html

{% extends 'base.html' %}

{%load filters%}

{% block content%}

<div style="margin-left:1500px;margin-top:30px">

<a id="a" href="{%url 'diplom:journal' %}?page={{page}}" class="btn btn-danger"> Назад</a>

</div>

<div class="row justify-content-center" id="a">

<div class="col-mb-5 p-5">

<div class="card" style="margin-left:600px;margin-right:600px;margin-top: -50px;">

<div class="card-header"> Создать

</div>

<div class="card-body" style="margin-right: 100px;margin-right: 100px;">

<form method="post" accept-charset="UTF-8" enctype="multipart/form-data">

{% csrf\_token %}

{%if eror%}

<!-- <p style="color:red;">Ошибочные вводимые параметры

</p> -->

<p style="color:red;">

<!-- № СП необходимо писать с "/" -->

<!-- {{erors}} -->

<script>

const er = "{{erors}}" console.log(er);

document.addEventListener('DOMContentLoaded', function(){ show\_message\_error(er);

});

</script>

<div id="modalSheetShowMessage" class="modal show">

<div class="modal-dialog">

<div class="modal-content rounded-4 shadow">

<div class="modal-header border-bottom-0">

<h5 class="modal-title">Ошибка</h5>

<button type="button" class="btn-close" data-bs-

dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-footer flex-column border-top-0">

<div id="AlertMessage" class="alert alert-danger

text-center fade show">Ошибка</div>

</div>

</div>

</div>

</div>

</p>

{%endif%}

{%if form.errors %}

{% for feild in form%}

{% for er in feild.errors%}

<p style="color:red;">{{er}} </p>

{%endfor%}

{%endfor%}

{% for er in form.non\_field\_errors%}

<p style="color:red;">{{er}} </p>

{%endfor%}

{%endif%}

{% for feild in form%}

{%if feild.name == "div\_no" %}

<div style="display: flex;">

<p style="margin-left: 30%;" id="a">Цех</p>

<select required style="border-radius:10px;padding: 5px; width: - webkit-fill-available;background-color: transparent;height: 30px; outline: none;margin- left:5%" id="div\_no" name="div\_no">

<option value="" disabled selected>{{div\_no}}</option>

{% for i in divs%}

<option value="{{i}}">{{i| zero}}</option>

{%endfor%}

</select>

</div>

{%elif feild.name == "pmd\_note\_tp" %}

<div style="display: flex;">

<select required style="border-radius:10px;padding: 5px; width: 70%;margin-bottom:5%; background-color: transparent;height: 40px; outline: none;margin- left:30%" id="chois" name="chois">

<option disabled value="" selected>Выбор</option>

<option value="pmd\_note\_tp" >Документ разрешающий запуск

материала</option>

<option value="pmd\_note\_pi" >Примечание ПИ</option>

</select>

</div>

<div class="mb-4" style="margin-left:150px; display: none;"

id="pmd\_note\_pi1">

Примечание ПИ

<div>

<input type="text" name="pmd\_note\_pi" class="form-control"

id="pmd\_note\_pi">

</div>

<br> Дата ПИ

<div>

<input type="date" name="date\_pi" class="form-control"

id="date\_pi">

</div>

</div>

<div class="mb-4" style="margin-left:150px;display: none;"

id="pmd\_note\_tp1">

Документ разрешающий запуск материала

<div>

<input type="text" name="pmd\_note\_tp" value="СТО Ж08.0058-2014 п.

5.5.5д)6)" class="form-control" id="pmd\_note\_tp">

</div>

</div>

{%else%}

<div class="mb-4" style="margin-left:150px">

{{feild.label}}

<div >

{{feild}}

</div>

</div>

{%endif%}

{%endfor%}

<button class="btn btn-primary" type="submit" style="margin- left:250px;">Сохранить</button>

</form>

</div>

</div>

</div>

</div>

<script> document.addEventListener("DOMContentLoaded",

function(){

var selectInput = document.getElementById("chois");

var customInput = document.getElementById("pmd\_note\_pi1"); var customInput1 = document.getElementById("pmd\_note\_tp1"); selectInput.addEventListener("change",function(){

if (selectInput.value=="pmd\_note\_pi"){ customInput.style.display = "block"; customInput1.style.display = "none";

}else{

customInput1.style.display = "block"; customInput.style.display = "none";

}

});

});

function show\_message\_error(message) {

const alertDivCheckInterval = setInterval(function() {

const alertDiv = document.getElementById('AlertMessage'); if (alertDiv) {

alertDiv.innerHTML = message;

const modalShowMessageEl = document.getElementById('modalSheetShowMessage'); if (modalShowMessageEl) {

const modalShowMessage = new bootstrap.Modal(modalShowMessageEl); modalShowMessage.show();

}

clearInterval(alertDivCheckInterval); // Останавливаем интервал, как только элемент найден

}

}, 100); // Проверяем каждые 100 миллисекунд

}

</script>

{%endblock%}

Г.5 Содержимое файла дочернего шаблона update\_journal.html

{% extends 'base.html' %}

{% block content%}

<div style="margin-left:1500px;margin-top:30px">

<a id="a" href="{%url 'diplom:journal' %}?page={{page}}#{{numberpunkt}}" class="btn btn- danger"> Назад</a>

</div>

<div class="row justify-content-center">

<div class="col-mb-5 p-5">

<div class="card" style="margin-left:600px;margin-right:550px">

<div class="card-header" id="a"> Изменить

</div>

<div class="card-body" style="margin-right: 100px;margin-left: 100px;">

<form method="post" accept-charset="UTF-8" enctype="multipart/form-data">

{% csrf\_token %}

{%if eror%}

<p style="color:red;">Ошибочные вводимые параметры

</p>

{%endif%}

<div class="mb-3" style="margin-left: 50px; margin-right: 50px;">

<p>Номер паспорта: {{result.pas\_no}}</p>

<hr>

<p>Материал по ТП: {{result.pmd\_material}}</p>

<!-- <p>{{result.pmd\_material}}</p> -->

<hr>

<p>Материал по лзк : реальный</p>

<hr>

<p>Номер ЛЗК: {{result.lfc\_no}}</p>

<hr>

<p>{{result.m\_lfc}}</p>

<hr>

<p>Ответсвенный за паспорт: {{result.ul\_login}}</p>

<hr>

<p>Ответсвенный за материал: {{result.pmd\_responsible}}</p>

</div>

<div class="mb-3" style="margin-left: 50px; margin-right: 50px;">

<label class="form-label">Документ разрешающий запуск

материала</label>

<input type="text" class="form-control" id="pmd\_note\_tp" name="pmd\_note\_tp" value="{{result.pmd\_note\_tp}}">

</div>

<div class="mb-3" style="margin-left: 50px; margin-right: 50px;">

<label class="form-label">Примечание по ПИ</label>

<input type="text" class="form-control" id="pmd\_note\_pi" name="pmd\_note\_pi" value="{{result.pmd\_note\_pi}}">

</div>

<!-- <div class="mb-3" style="margin-left: 50px; margin-right: 50px;">

<label class="form-label">Ответсвенный за паспорт</label>

<input type="text" class="form-control" id="ul\_login" name="ul\_login" value="{{result.ul\_login}}">

</div>

<div class="mb-3" style="margin-left: 50px; margin-right: 50px;">

<label class="form-label">Ответсвенный за материал</label>

<input type="text" class="form-control" id="pmd\_responsible" name="pmd\_responsible" value="{{result.pmd\_responsible}}">

</div> -->

<button id="a" class="btn btn-primary" type="submit" style="margin- left:130px;">Сохранить</button>

</form>

</div>

</div>

</div>

{%endblock%}

Г.6 Содержимое файла дочернего шаблона analytics.html

{% extends 'base.html' %}

{%load filters%}

{% load static %}

{% block content %}

<div class="container mt-4" id="a">

<h1 class="text-center mb-4">Процент паспортов с заменой материала по отношению к общему количеству СП</h1>

<form method="get" action="{% url 'diplom:analytics' %}">

Год <select style="border-radius:10px;padding: 5px; box-shadow: none; outline: none; border: none; margin-top: 20px;" id="year" name="year">

<option value="" disabled selected>{{yaer\_one | nones }}</option>

{% for i in year%}

<option value="{{i}}">{{i}}</option>

{%endfor%}

</select>

<button id='a' class="btn" type="submit" style="margin-left:0.00001px;" onclick="search()"> <img src={% static 'img/search.png' %}> Сформировать</button>

</form>

<div class="d-flex justify-content-end mb-3">

<a href="{% url 'diplom:journal' %}" class="btn btn-danger">Назад</a>

</div>

<!-- <div class="card shadow-lg p-4"> -->

<div id="analyticsChart" ></div>

<!-- </div> -->

<br>

<br>

<h1 class="text-center mb-4">Процент паспортов с заменой материала по отношению к общему

количеству паспортов по цеху и году</h1>

<form method="post" style="margin-top: 20px;margin-left: 20px;" >

{% csrf\_token %}

Цех <select style="border-radius:10px;padding: 5px; box-shadow: none; outline: none; border: none; margin-top: 20px;" id="div\_no" name="div\_no">

<!-- <option value="" disabled selected>{{div\_no}}</option> -->

<option value="" disabled selected>{{divs\_no}}</option>

{% for i in div\_no%}

<option value="{{i}}">{{i |zero}}</option>

{%endfor%}

<option value="0">Все</option>

</select>

&nbsp &nbsp &nbsp Год <select style="border-radius:10px;padding: 5px; box-shadow: none; outline: none; border: none; margin-top: 20px;" id="year" name="year">

<option value="" disabled selected>{{years}}</option>

{% for i in year%}

<option value="{{i}}">{{i}}</option>

{%endfor%}

</select>

<button id='a' class="btn" type="submit" style="margin-left:0.00001px;" onclick="search()" > <img src={% static 'img/search.png' %}> Сформировать</button>

</form>

<br>

<!-- <div class="card shadow-lg"> -->

<div id="analytics" ></div>

<!-- </div> -->

<div id="loadingIndicator" class="text-center mt-3 d-none">

<img src="/static/img/loading.ico" id="loading-icon" width="64" height="64">

</div>

<!-- <div id="detWind"> -->

<!-- <span class="close-btn" onclick="document.getElementById('detWind').style.display='none'">x</span> -->

<div class="d-none" id="detCont"> <h1></h1> </div>

<!-- </div> -->

</div>

<style>

.table-cont{

max-height: 400px; overflow-y: auto;

}

body { margin: 0; height: 250vh;

background: linear-gradient(to bottom, #ececec, #4495D1); background-repeat: no-repeat;

}

.close-btn{

position: absolute; top:5px;

right: 10px; cursor: pointer;

}

#detWind{ display: none;

position: fixed;

}

</style>

<script>

document.body.style.background = '';

document.addEventListener("DOMContentLoaded", function () { const loader = document.getElementById('loadingIndicator'); loader.classList.remove('d-none');

// var canvas = document.getElementById('analyticsChart');

// var ctx = canvas.getContext('2d'); var data = {{ data|safe }};

var data1 = {{ data1|safe }}; var labels = {{ div\_no|safe }};

var percentages = data.map((value, index) => { let total = value + data1[index];

return total ? ((data1[index] / total) \* 100).toFixed(2) : 0;

});

const chartData = labels.map((label, index) => ({ name: label,

y: parseFloat(percentages[index]), total: data[index] + data1[index], part1: data[index],

part2: data1[index]

}));

Highcharts.chart('analyticsChart', { chart: {

type: 'column',

},

title: {

text : 'Аналитика'

},

xAxis : {

categories: labels, title : {

text: 'Цеха'

}

},

yAxis:{

min: 0,

max: 100, title : {

text: 'Проценты(%)'

}

},

tooltip : {

formatter: function(){ console.log(this)

const y = this.y + '%'; return '<b>' + y;

}

},

plotOptions: { column: {

pointPadding: 0.2,

borderWidth: 0,

}

},

series:[{

showInLegend: false, color:'rgb(44, 175, 254)', data: chartData, dataLabels:{

enable: false, format: '{y}%'

}

}],

credits:

{

enabled: false

},

})

loader.classList.add('d-none');

});

function search(){

const loader = document.getElementById('loadingIndicator'); loader.classList.remove('d-none');

const url = window.location.href; console.log(url);

}

document.addEventListener("DOMContentLoaded", function () { const loader = document.getElementById('loadingIndicator');

const divres = document.getElementById('detCont') var data = {{ data2\_pas|safe }};

var data1 = {{ data3\_pasj|safe }}; var labels = {{ month|safe }};

var divs = {{divs\_no | safe}}; var year = {{years}};

var percentages = data.map((value, index) => { let total = value;

return total ? ((data1[index] / total) \* 100).toFixed(2): 0;

});

const chartData = labels.map((label, index) => ({ name: label,

y: parseFloat(percentages[index]), total: data[index] + data1[index], part1: data[index],

part2: data1[index],

}));

Highcharts.chart('analytics', { chart: {

type: 'column', events:{

load: function(){

}

}

},

title: {

text : 'Аналитика'

},

xAxis : {

categories: labels, title : {

text: 'Время года'

}

},

yAxis:{

min: 0,

max: 100, title : {

text: 'Проценты(%)'

}

},

tooltip : {

formatter: function(){

var url = 'journal/api/analytics/' + this.options.name + '/' + year + '/'

+ divs;

fetch(url).then(response =>{ if (!response.ok){

throw new Error('err');

}

return response.json();

}).then(data =>{

const firs100 = data; let htmlcon = `

<br>

<h1 style="margin-left:50%">${this.options.name}</h1>

<br>

<div class="table-cont"> <table id="myTable" style="margin-top: 20px; margin-left: 10px; margin-right: 10px;position: sticky" >

<thead>

<tr >

<th>№ СП</th>

<th>Код МТР</th>

<th>№ ЛЗК</th>

</tr>

</thead>

<tbody>

` firs100.forEach(item => {

htmlcon += `

<tr>

<td style="text-align: left;" class="item-name">$

{item.pas\_no}/${item.div\_no} ${item.pas\_year}</td>

<td class="item-name">${item.mtr\_code}</td>

<td class="item-name">${item.lfc\_no}</td>

</tr>

`

divres.innerHTML = htmlcon;

})

document.getElementById('detCont').classList.remove('d-none'); console.log(firs100)

})

const y = this.y + '%';

// document.getElementById('detCont').innerHTML = return '<b>' + y;

}

},

plotOptions: { column: {

pointPadding: 0.2,

borderWidth: 0,

}

},

series:[{

showInLegend: false, color:'rgb(44, 175, 254)', data: chartData, dataLabels:{

enable: false, format: '{y}%'

}

}],

credits:

{

enabled: false

},

})

loader.classList.add('d-none');

});

</script>

{% endblock %}

{%load filters%}

**Приложение Д (обязательное)**

**Код реализации пагинации**

{% if page\_obj.has\_other\_pages %}

<nav aria-label="Page navigation" class="my-5">

<ul class="pagination">

{%if page\_obj.has\_previous %}

<li class="page-item"><a class="page-link" href="?page=1">Первая</a></li>

<li class="page-item">

<a class="page-link" href="?page={{page\_obj.number|reduce}}">Предыдущая</a>

</li>

{% endif %}

{% for i in page\_obj.number|iter%}

{%if page\_obj.number == i%}

<li class="page-item active">

<span class="page-link" >{{ i }}</span>

</li>

{%else%}

<li class="page-item">

<a class="page-link" href="?page={{ i }}">{{ i}}</a>

</li>

{% endif %}

{% endfor %}

{% if page\_obj.has\_next %}

<li class="page-item ">

<a class="page-link" href="?page={{page\_obj.next\_page\_number}}">Следущая</a>

</li>

<li class="page-item">

<a class="page-link" href="?page={{page\_obj.paginator.num\_pages}}">Последняя</a>

</li>

{% endif %}

</ul>

</nav>

{% endif %}