|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное государственное АВТОНОМНОЕ образовательное учреждение высшего образования  «Национальный исследовательский ядерный университет «МИФИ» |
| **Технологический институт –**  филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»  **(ТИ НИЯУ МИФИ)** |

**Кафедра информационных технологий и прикладной математики**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

направление подготовки 09.03.01 «Информатика и вычислительная техника»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Тема:** | Разработка программного модуля для сопровождения переноса | | | | |
|  | затрат между сопроводительными паспортами в ИС МКАДД | | | |  |
|  | |  | | |  |
| **Руководитель** | |  | | |  |
| Инженер-программист 2 категории | |  |  |  | С.А.Коненков |
| *(должность)* | |  | *(подпись)* |  | *(И.О. Фамилия)* |
| **Студент** | |  |  |  |  |
| ИВТ-41Д | |  |  |  | П.А.Пустовалов |
| *(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

[ВВЕДЕНИЕ 3](#__RefHeading___Toc8883_462899921)

[ГЛАВА 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ 6](#__RefHeading___Toc8855_462899921)

[1.1 Анализ предметной области 6](#__RefHeading___Toc8857_462899921)

[1.1.1 БД и технологии. 9](#__RefHeading___Toc8859_462899921)

[1.2 Описание бизнес-процесса перенос фактических затрат с одного сопроводительного паспорта на другой 11](#__RefHeading___Toc8861_462899921)

[1.3 Средства разработки программного обеспечения 15](#__RefHeading___Toc8863_462899921)

[1.4 Сценарии выполнения основных действий 19](#__RefHeading___Toc8865_462899921)

[ГЛАВА 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ 24](#__RefHeading___Toc8867_462899921)

[2.1 Исследование базы данных 24](#__RefHeading___Toc8869_462899921)

[2.2 Разработка серверной части приложение 36](#__RefHeading___Toc8871_462899921)

[2.3 Проектирование клиентской части 43](#__RefHeading___Toc8873_462899921)

[ГЛАВА 3 ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ 46](#__RefHeading___Toc8875_462899921)

[3.1 Руководство пользователя. Тестирование приложения. 46](#__RefHeading___Toc8877_462899921)

[3.2 Планируемый эффект от внедрения 46](#__RefHeading___Toc8879_462899921)

[Обозначения и сокращения 47](#__RefHeading___Toc8881_462899921)

# ВВЕДЕНИЕ

В современном обществе цифровизация и автоматизация бизнес-процессов занимают одно из главных мест в стратегии развития практически всех отраслей экономики. Использование информационных технологий позволяет повысить эффективность, сокращение затрат и снижение риска, связанные с человеческим фактором. С ростом усложнения производственных процессов и роста объёмов данных переход на кроссплатформенные[описать что такое кроссплатформенные], импортонезависимые ИТ-решения становится необходимым условием для стабильной работы предприятий.

Одной из ключевых задач цифровой трансформации является интеграция инновационных решений, способных обеспечить гибкость и адаптивность информационных систем. В настоящее время многие предприятия стремятся отказаться от устаревших закрытых систем и перейти на открытые платформы, что позволяет не только оптимизировать внутренние процессы, но и повысить уровень безопасности. Такой подход становится особенно актуальным в свете необходимости защиты корпоративных данных от внешних угроз и непредвиденных сбоев, связанных с использованием программного обеспечения, разработанного для узкоспециализированных операционных систем.

Особую важность приобретает переход на открытую операционную систему Linux. Программное обеспечение, разработанное для Windows, зачастую функционирует корректно в своей исходной среде, однако при эксплуатации на Linux могут возникать проблемы, поскольку сами приложения не создавались как кроссплатформенные. В связи с этим осуществляется поэтапный переход на веб-технологии, что позволяет обеспечить независимость от конкретной платформы и повысить надежность работы систем.

Отдел «Управление информационных технологий и связи» (УИТиС) ФГУП «Комбинат «Электрохимприбор» разрабатывает и поддерживает программное обеспечение, необходимое для автоматизации бизнес-процессов предприятия. С увеличением требований к информационной безопасности и надежности управления критически важной аппаратурой происходит переход системы управления на импортонезависимые компоненты. Это обеспечивает гарантированную защиту системы в любых условиях и минимизирует риски, связанные с эксплуатацией программного обеспечения, разработанного для закрытых платформ.

Целью данной работы является разработка программного модуля для сопровождения переноса затрат между сопроводительными паспортами. Данный модуль представляет собой часть ИС МКАДД, которое автоматизирует учёт и контроль переноса затрат между сопроводительными паспортами, обеспечивая простоту эксплуатации, высокую точность данных и возможность анализа информации. Разработка этого модуля является важным элементом поэтапного миграции на веб-технологии, что позволит обеспечить гибкость, точность и прозрачность работы системы.

Техническое задание определяет стек технологий, необходимых для реализации программного продукта. В качестве серверной части используется язык программирования Python с фреймворками Django и Django Rest Framework. Для разработки клиентской части применяются HTML, CSS, JavaScript и библиотека Bootstrap5. Этот современный стек позволяет создавать надёжные, масштабируемые и адаптивные веб-приложения, соответствующие современным требованиям информационной безопасности и удобства использования.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ предметной области;
2. Изучить существующую на предприятии систему для дальнейшей интеграции и составить модель бизнес-процесса;
3. Спроектировать базу данных;
4. Разработать клиентскую часть приложения;
5. Реализовать веб-сервис;
6. Протестировать программный продукт;
7. Составить инструкцию пользователя;
8. Проанализировать эффект от внедрения разработки.

# ГЛАВА 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ

## 1.1 Анализ предметной области

Основным документом, обеспечивающим организацию и контроль последовательности выполнения операций по маршруту, установленному ТП, является СП, который сопровождает партию изделий в течение всего жизненного цикла на комбинате. Партия изделий может включать как одно изделие, так и несколько. СП содержит данные о партии изделий (одной номенклатуры), включая количество, а также последовательность выполнения операций (маршрут). Главной характеристикой СП является информация об изделии, которая описывается через его номенклатуру. Номенклатура определяет основные свойства изделия, такие как название, обозначение и размерные параметры. Учётная точка фиксирует этапы выполнения операций в конкретных производственных зонах. Каждая операция представляет собой отдельное производственное действие, направленное на обработку ДСЕ. Для выполнения производственной задачи может понадобиться целый набор операций, которые должны быть выполнены в строгой последовательности. Маршрут последовательность выполнения операций, определяющая движение ДСЕ между цехами. В данном случае маршрут представляет собой последовательность цехов (цехозаходов), через которые проходит ДСЕ в процессе производства. В каждом цехозаходе установлен порядок выполнения операций. СЗ формируется мастером и представляет собой документ, формируемый мастером рабочему с указанием конкретной задачи по обработке или сборке ДСЕ. Накладная используется для учёта перемещения ДСЕ между цехами или другими подразделениями, содержа в себе информацию об передаваемом ДСЕ.

В случае необходимости может потребоваться перенос затрат — документ, фиксирующий перераспределение затрат по паспортам. Пользователь – это сотрудник предприятия, который участвует в производственном процессе. Полномочия пользователя определяют, какие действия пользователь может совершать в ИС.

Таким образом, в предметной области можно выделить следующие объекты и их параметры:

* Паспорт: номер паспорта, год, цех, номер заказа, количество ДСЕ в партии, материалы, особые замечания, ДСЕ (объект Номенклатура), маршрут (объект Маршрут).
* Номенклатура: название ДСЕ, обозначение ДСЕ, размерные параметры ДСЕ, пятнадцатизначный код.
* Операция: порядковый номер операции, название операции, код операции, участок.
* Маршрут: перечень цехозаходов (объект Цехозаход).
* Цехозаход: номер цеха, порядковый номер в маршруте, список операций (объект Операция).
* Накладная: номер накладной, наименование ДСЕ, количество получено, количество отправлено, количество брака, цех, дата отправки, дата приёма, ответственный за получение (объект Пользователь), ответственный за отправку (объект Пользователь).
* Сменное задание: номер задания, дата выдачи, дата приёма, плановая дата окончания, ДСЕ (объект Номенклатура), список операций (объект Операции), мастер (объект Пользователь), рабочий (объект Пользователь), учётная точка (объект Учётная точка). Учётная точка: цех, участок, здание, номер учётной точки.
* Перенос затрат: количество, дата добавления, дата подтверждения, паспорт основной (объект Паспорт), паспорт источник (объект Паспорт), ДСЕ (объект Номенклатура).
* Пользователь: табельный номер, фамилия, имя, отчество, логин.
* Полномочия пользователя: перечень полномочий для работы системой, установленных для пользователя, пользователь (объект Пользователь).

Необходимость автоматизированной системы переноса фактических затрат между сопроводительными паспортами в ИС МКАДД внутри одного заказа основного производства, объясняется тем, что при запросе материала на склад, необходимого материала для изготовления ДСЕ может не быть, но на складах может хранится продукция, снятая с производства. В случаях острой производственной необходимости, в которых допускается использование заготовок или полуфабрикатов одной позиции для производства другой, есть возможность произвести перенос затрат между сопроводительными паспортами, для этого необходимо получить подтверждение, что затраты можно перенести от начальника ПДУ, разрешение у технолога на замену, технолог выпускает ПИ на доработку или/и изменения маршрута ТД. Лицо, ответственное за оформление СП, на основание ПИ выпускает новый СП, после чего создаёт заявку в ИС, контролёр УТК заверяет достоверность внесённой информации и сохраняет отметку от подтверждения в ИС.

Автоматизация системы переноса затрат в ИС МКАДД является неотъемлемой частью обеспечения непрерывной работы предприятия, данная система должна обеспечить простоту использованию, точность данных и возможность анализа данных, а также решить следующие задачи:

* Просмотр списка паспортов с переносом затрат;
* Создание записи для переноса затрат;
* Подтверждение переноса затрат и сам перенос затрат;
* Аналитика: Диаграмма % паспортов с перенесенными затратами внутри одного заказа по отношению к общему количеству паспортов по заказу за период (месяц, год).

### 1.1.1 БД и технологии.

Для хранения данных в приложении была выбрана база данных PostgreSQL Pro. Это объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом, которая отличается высокой производительностью и масштабируемостью, а также сертифицирована ФСТЭК, что обеспечивает соответствие требованиям безопасности для государственных и критически важных систем. PostgreSQL Pro полностью поддерживает стандарт SQL и предоставляет расширенные возможности для работы с большими объёмами данных.

Основные характеристики PostgreSQL Pro:

* Открытый исходный код: PostgreSQL Pro является проектом с открытым исходным кодом, доступным для использования, модификации и распространения.
* Соответствие стандартам SQL: СУБД поддерживает полный набор функций стандарта SQL, включая сложные запросы, транзакции и обработку больших объёмов данных.
* Расширяемость: PostgreSQL Pro позволяет пользователям добавлять собственные функции, типы данных и операторы, что делает её гибкой для решения широкого спектра задач.
* Надёжность и согласованность: Поддержка транзакций на основе ACID гарантирует целостность данных и надёжную обработку запросов.
* Высокая производительность: СУБД демонстрирует отличные результаты при обработке больших объёмов данных и выполнении сложных запросов. В тестах, таких как TPC-H, PostgreSQL Pro часто сравнима или превосходит другие СУБД, достигая свыше 1 000 000 операций чтения в секунду.
* Масштабируемость: PostgreSQL Pro поддерживает репликацию для горизонтального масштабирования, что обеспечивает высокую доступность, эффективное распределение нагрузки и географическую децентрализацию данных. Однако репликация требует тщательной настройки для обеспечения согласованности и упрощения управления.
* Сертификация ФСТЭК: Наличие сертификата ФСТЭК является важным преимуществом PostgreSQL Pro, гарантируя соответствие высоким стандартам безопасности и требованиям государственных заказчиков.
* Таким образом, PostgreSQL Pro была выбрана для реализации проекта благодаря своей надёжности, производительности и безопасности, что особенно важно для работы с большими данными в критически важных системах.

## 1.2 Описание бизнес-процесса перенос фактических затрат с одного сопроводительного паспорта на другой

Бизнес-процесс — последовательность взаимосвязанных операций, которые выполняются на предприятии для достижения определенной цели или результата. Бизнес-процесс включает различные этапы и задачи, которые должны быть выполнены в определенном порядке и с определенными ресурсами.

Основанием выписки СП является план-график, выданный ПДО. На основе позиции плана ПДГ формирует СП, СП сохраняется в ИС, после чего ПДГ печатает СП.

На основании СП мастер создаёт заявку на получение материала. На основание СП (заявки) кладовщик склада материалов выдаёт материал мастеру. После получения материала, мастер создаёт СЗ и вместе с материалом передаёт его рабочему.

Рабочий от мастера получает СЗ, материалы и начинает выполнять операции по СЗ. После выполнения СЗ, мастер проверяет выполнение задания и в случае необходимости передаёт контролеру.

Если ДСЕ готово и выполнена последняя операции маршрута паспорта, мастер передаёт ДСЕ контролёру. Контролёр проводит контрольные мероприятия, если контроль пройден, устанавливает признак готовности ДСЕ и сохраняет запись об этом в ИС. Отметка о выполнении позиции плана отправляется в плановую систему, после чего ДСЕ передаётся на склад ДСЕ.

Если контроль не пройдён, то ДСЕ переводится на работу с несоответствиями.

Если ДСЕ не готово и если следующая операция в текущем цехе, то мастер формирует новое задание и передает его рабочему.

Если следующая операция не в текущем цехе, то мастер отправляет ДСЕ на склад ДСЕ для передачи, кладовщик в свою очередь отправляет в следующий цех, для продолжения выполнения операций по СП.

В некоторых случаях на складах может хранится продукция, снятая с производства. В случаях острой производственной необходимости, в которых допускается использование заготовок или полуфабрикатов одной позиции для производства другой. Начальник ПДУ подтверждает, что затраты можно перенести и предаёт информацию о позициях и необходимом количестве технологу. Технолог в свою очередь выпускает ПИ на доработку и/или изменения маршрута ТД. На основании ПИ лицо, ответственное за оформление СП, выпускает новый СП и создаёт заявку на перенос СП в ИС. Контролёр УТК заверяет достоверность внесённой информации и подтверждает заявку о переносе затрат в ИС.

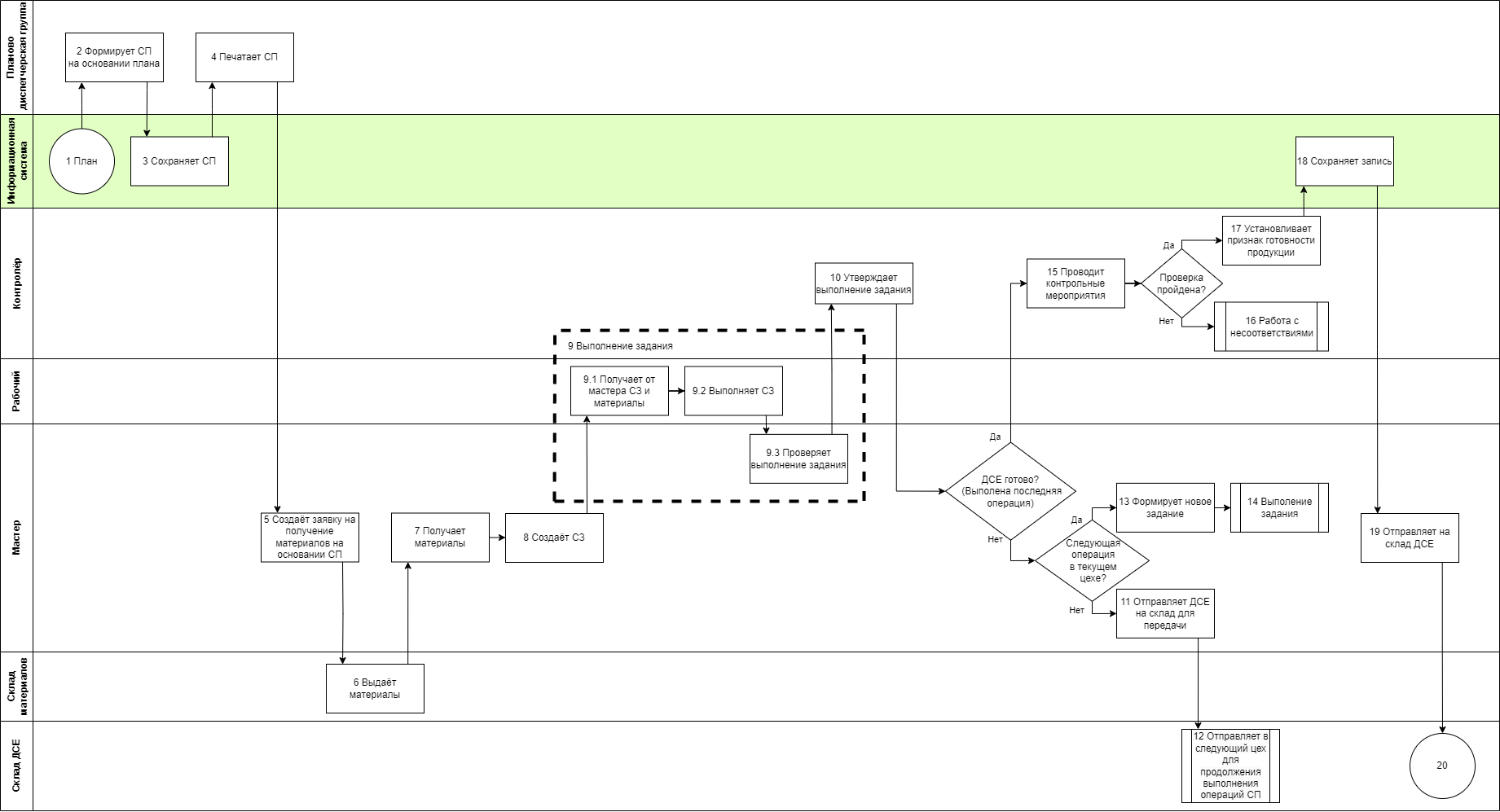


Рисунок 1.1 Описание бизнес-процесс производства детали

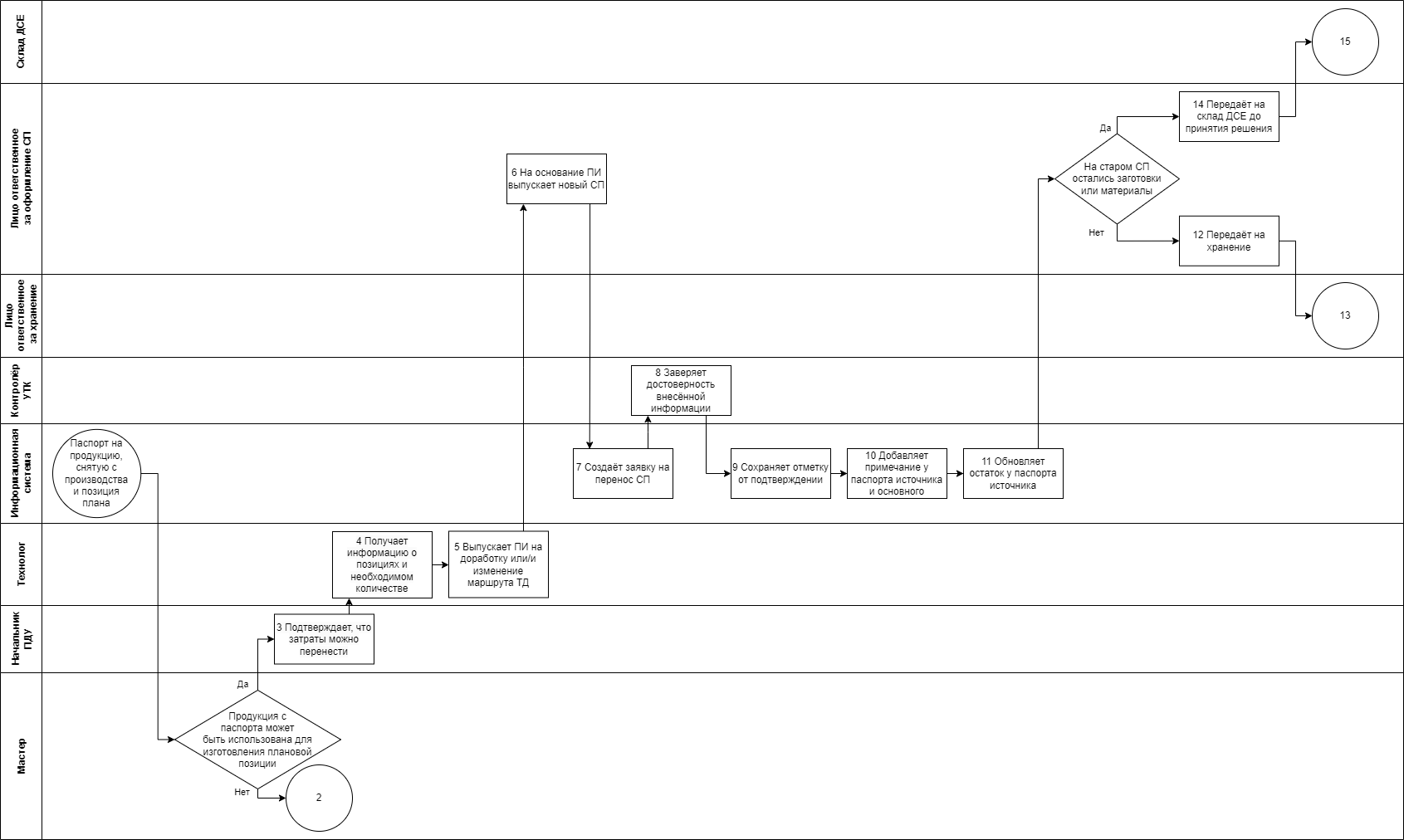


Рисунок 1.2 Описание бизнес-процесса переноса затрат между паспортами

## 1.3 Средства разработки программного обеспечения

В соответствии с техническим заданием для разработки веб-приложения требуется использовать набор веб-технологий (HTML, CSS, JavaScript и Python) наряду с импортонезависимым программным обеспечением.

Серверная часть:

* Python – высокоуровневый интерпретируемый язык программирования, известный своей читаемостью и простотой. Он поддерживает различные парадигмы, такие как процедурное, объектно-ориентированное и функциональное программирование, что делает его универсальным инструментом для разработки.
* Django – это открытый веб-фреймворк, предназначенный для создания надёжных и качественных веб-приложений в кратчайшие сроки. Фреймворк построен на концепции архитектуры MVC (Model-View-Controller). В реализации Django данный паттерн преобразован в Model-View-Template (MVT):
  + Model описывает структуру данных и реализует бизнес-логику посредством ORM, обеспечивая взаимодействие с базой данных.
  + View служит посредником, обрабатывая запросы пользователей, получая данные из моделей и определяя, какой шаблон использовать для отображения.
  + Template отвечает за визуальное представление информации конечному пользователю.  
    Таким образом, в Django модель отражает данные, а представление и шаблоны формируют пользовательский интерфейс, а роль контроллера берёт на себя логика middleware.
* Django Rest Framework (DRF) – это расширяемый набор инструментов для создания RESTful API на основе Django, упрощающий процесс разработки веб-сервисов.
* API (Application Programming Interface) представляет собой совокупность правил и протоколов, позволяющих различным программным компонентам взаимодействовать друг с другом.
* PostgreSQL – система управления реляционными базами данных с открытым исходным кодом, известная своей надёжностью, гибкостью и поддержкой расширенных возможностей SQL и разнообразных типов данных.
* HTTP (Hypertext Transfer Protocol) – протокол прикладного уровня, используемый для передачи гипертекстовых документов в сети Интернет. Он работает по модели запрос-ответ, где клиент посылает запрос (например, GET, POST, PUT и др.), а сервер возвращает соответствующую информацию.

Клиентская часть:

* HTML (HyperText Markup Language) – стандартный язык разметки для создания и структурирования веб-страниц, задающий их базовую структуру.
* CSS (Cascading Style Sheets) – язык таблиц стилей, предназначенный для определения внешнего вида и оформления HTML-документов.
* JavaScript – универсальный язык программирования высокого уровня, который позволяет создавать интерактивный и динамический контент на веб-сайтах.
* Bootstrap – популярный фреймворк с открытым исходным кодом, облегчающий разработку веб-интерфейсов с использованием готовых компонентов HTML, CSS и JavaScript.

Для работы с данным технологическим стеком выбран редактор исходного кода Visual Studio Code. Он обеспечивает удобное редактирование на различных языках, поддерживает подсветку синтаксиса и автоматическую корректировку ошибок, что существенно ускоряет процесс создания проекта. Среда разработки представлена на рисунке 1.2.

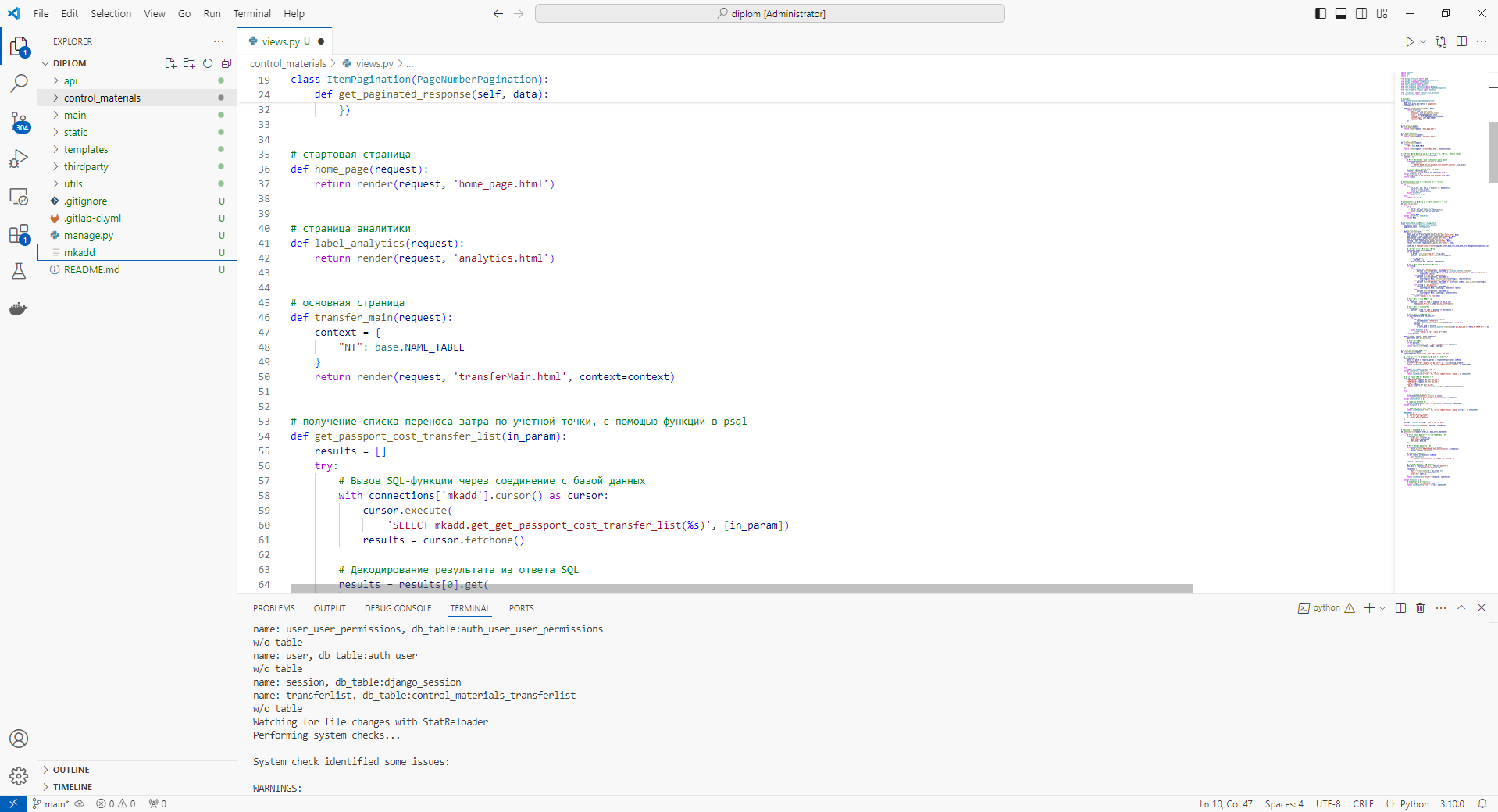
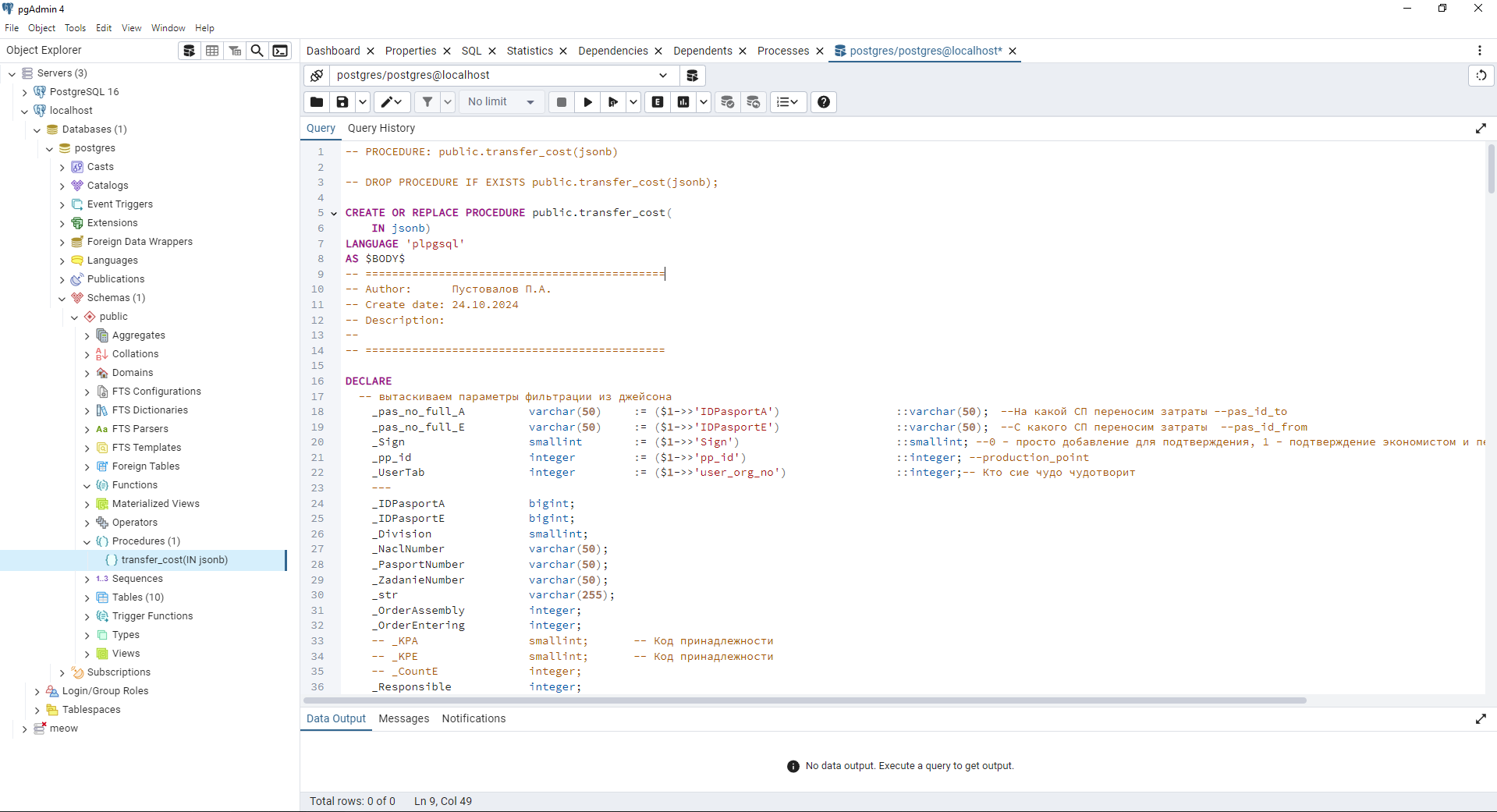


Рисунок 1.3 – Редактор кода Visual Studio Code

Для разработки базы данных, необходимой для работы приложения, была использована СУБД PostgreSQL, являющаяся стандартной для предприятия и указанной в техническом задании. Управление СУБД осуществляется с помощью специализированного графического клиента pgAdmin(см. рисунок 1.3).

Рисунок 1.4 – графический клиент pgAdmin

Вывод: в соответствии с техническим заданием на разработку веб-сервиса «Разработка программного модуля для сопровождения переноса затрат между сопроводительными паспортами 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.

## 1.4 Сценарии выполнения основных действий

**Сценарий - 1. «Создание заявки на перенос затрат на СП»**

|  |  |
| --- | --- |
| Автор | Пустовалов П.А. |
| Дата создания | 18.12.2024 |
| Основное действующее лицо | Лицо, ответственное за формирование СП. |
| Описание | Лицо, ответственное за формирование СП, создаёт в ИС заявку на перенос затрат СП. |
| Условие-триггер | Возникла необходимость создать заявку на перенос затрат и материалов. |
| Последовательность выполнения | * 1. **Лицо, ответственное за формирование СП, добавляет заявку на перенос затрат.**      1. На основании ПИ лицо, ответственное за формирование СП, вносит в форму данные: №СП основного, №СП источника      2. Система создаёт заявку на перенос затрат. |
| Условия | * + 1. Номера паспортов введены корректно.     2. Паспорт приёмник находится не в замороженном состоянии.     3. Оба паспорта должны находится на одной точке.     4. Перенос затрат по данным паспортам уже не выполнен.     5. На СП источнике должна стоять дата УТК.     6. Количество на паспортах больше нуля.     7. Паспорта не отделены друг от друга.     8. Паспорт не находится в непринятой накладной.     9. Паспорт не находится в непринятом задании.     10. Паспорт не удалён.     11. Заказ сборочного паспорта присутствует в справочнике открытых заказов.     12. Заказ входящего паспорта присутствует в справочник открытых заказов.     13. Паспорт-источник по заказам 0х0777 и 0х0888 проходит через СПФ.     14. Заказ кредита входящей равен заказу головы.     15. Заказ потребления паспорта-источника равен заказу основного паспорта.     16. У основного паспорта нет ЛЗК на шапке паспорта.     17. Заявка на перенос затрат по данным паспортам не создана. |

# 

**Сценарий - 2. «Подтверждения заявки на перенос затрат»**

|  |  |
| --- | --- |
| Автор | Пустовалов П.А. |
| Дата создания | 19.12.2024 |
| Основное действующее лицо | Контролёр УТК. |
| Описание | Контролёр УТК, подтверждает заявку на перенос затрат. |
| Условие-триггер | Необходимо подтвердить перенос затрат. |
| Последовательность выполнения | * 1. **Контролёр УТК, подтверждает заявку на перенос затрат.**      1. Находит/выбирает нужную запись в системе.      2. Подтверждает заявку на перенос затрат. |
| Условия выполнения | * + 1. Паспорт приёмник находится не в замороженном состоянии.     2. Оба паспорта должны находится на точке.     3. Перенос затрат по данным паспортам ещё не выполнен.     4. На СП источнике должна стоять дата УТК.     5. Количество на паспортах больше нуля.     6. Паспорта не отделены друг от друга.     7. Паспорт не находится в непринятом накладной.     8. Паспорт не находится в непринятом задании.     9. Паспорт не удалён.     10. Заказ сборочного паспорта присутствует в справочнике открытых заказов.     11. Заказ входящего паспорта присутствует в справочник открытых заказов.     12. Паспорт-источник по заказам 0х0777 и 0х0888 проходит через СПФ.     13. Заказ кредита входящей равен заказу головы.     14. Заказ потребления паспорта-источника равен заказу основного паспорта.     15. У основного паспорта нет ЛЗК на шапке паспорта.     16. Заявка на перенос затрат по данным паспорта существует.     17. Есть разрешения на данную операцию. |

**Сценарий - 3. «Удаление заявки на перенос затрат»**

|  |  |
| --- | --- |
| Автор | Пустовалов П.А. |
| Дата создания | 19.12.2024 |
| Основное действующее лицо | Лицо, ответственное за формирование СП. |
| Описание | Лицо, ответственное за формирование СП, удаляет созданную заявку на перенос затрат СП в ИС. |
| Условие-триггер | Необходимо удалить заявку перенос затрат. |
| Последовательность выполнения | * 1. **Лицо, ответственное за формирование СП, удаляет заявку на перенос затрат.**      1. Находит/выбирает нужную запись в системе.      2. Удаляет заявку на перенос затрат. |
| Условия выполнения | * + 1. Заявка на перенос затрат по данным паспортам создана.     2. Перенос затрат по данным паспортам ещё не выполнен. |

# ГЛАВА 2 РАЗРАБОТКА ПРИЛОЖЕНИЯ

## 2.1 Исследование базы данных

В процессе разработки программного продукта возникла необходимость обработки значительных объёмов информации. Для эффективного выполнения данной задачи были реализованы функции на языке PL/pgSQL в среде управления базами данных PostgreSQL.

Функции в PostgreSQL представляют собой заранее написанный код, предназначенный для автоматизированного выполнения операций с данными. Их использование позволяет сократить дублирование кода, повысить читаемость и упростить сопровождение программного решения. В отличие от хранимых процедур, функции обладают рядом преимуществ:

* Гибкость в использовании: функции могут принимать параметры, что делает их универсальными.
* Интеграция со встроенными операторами и функциями PostgreSQL.
* Возвращение значений, что позволяет использовать их непосредственно в SQL-запросах.

Применение функций способствует оптимизации обработки информации, снижая нагрузку на систему и повышая производительность.

В рамках реализации функциональности были разработаны следующие функции:

* Функция для получения списка переноса затрат (листинг 1).
* Функции для создания, подтверждения и удаления заявки на перенос (листинг 2).
* Функция для аналитического расчёта паспортов с перенесёнными затратами в разрезе заказов за определённый период (листинг 3).

Листинг 1 — Функция вывода списка переноса затрат

*(код функции)*

Листинг 2 — Функция обработки заявок на перенос затрат

*(код функции)*

Листинг 3 — Функция аналитики паспортов с перенесёнными затратами

*(код функции)*

Следует отметить, что проектирование структуры базы данных не входило в рамки данной работы. Была осуществлена только разработка функций для обработки данных, основанная на анализе и адаптации существующих решений, реализованных ранее в MySQL. Перенос функциональности в PostgreSQL потребовал переосмысления подходов к организации кода и оптимизации выполнения операций.

## 2.2 Разработка серверной части приложение

Для реализации web-приложения «Сопровождения переноса затрат между сопроводительными паспортами в ИС МКАДД», был определён и настроен Django проект.

[photo]

Django — это мощный web-фреймворк, который позволяет разделять код на приложения (apps), представляющие собой независимые модули с собственными моделями, представлениями, шаблонами и статическими файлами. Такой подход делает систему более гибкой, упрощает поддержку и дальнейшее масштабирование.

При проектировании серверной части в файле settings.py были определены настройки, включая подключение модуля control\_materials. Ниже представлен фрагмент списка установленных приложений (листинг 4).

Листинг 4 — значения INSTALLED\_APPS

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'transfer\_cost',

    'rest\_framework',

]

Статические файлы (изображения, стили CSS, JavaScript-скрипты) хранятся в директории /static/, структура которой определяется на этапе проектирования. Для корректной обработки этих файлов в settings.py был добавлен параметр STATICFILES\_DIRS (листинг 5).

Листинг [5] – Параметр STATICFILES\_DIRS

STATICFILES\_DIRS = (os.path.join(BASE\_DIR, 'static'),)

Связь с базой данных устанавливается через переменную DATABASES, в которой указываются следующие параметры:

* ENGINE – драйвер для работы с СУБД;
* NAME – имя базы данных;
* USER – имя пользователя БД;
* PASSWORD – пароль пользователя;
* HOST – адрес сервера базы данных;
* PORT – порт подключения.

Пример настройки базы данных (листинг 6):

Листинг 6 – Пример подключения к базе данных

DATABASES = {

'default': {

‘ENGINE’: ‘django.db.backends.postgresql’,

‘HOST’: ‘’,

‘NAME’: ‘’,

‘SCHEME’: ‘auth’,

‘PORT’: ‘5432’,

‘USER’: ‘django\_auth\_user’,

‘PASSWORD’: ‘django\_auth\_user’,

},

}

В основе серверной части лежит API, которое выполняет роль посредника между пользователем и базой данных. Основная модель, используемая для работы с сервером, представлена в листинге 7.

Листинг 7 — Основная модель.

from django.db import models

class transferList(models.Model):

psE\_id = models.CharField(max\_length=255,blank=True, null=True) #номер паспорта Основной

pas\_no\_full\_to = models.CharField(max\_length=255,blank=True, null=True) #номер паспорта Основной

pas\_no\_full\_from = models.CharField(max\_length=255,blank=True, null=True) #номер паспорта Источник

nom\_designation = models.CharField(max\_length=255,blank=True, null=True) #ОбозначениеНОВОЕ

nom\_name = models.CharField(max\_length=255,blank=True, null=True) #Наименование

nom\_size = models.CharField(max\_length=255,blank=True, null=True) #ТипоРазмер

pct\_quantity = models.IntegerField(max\_length=255,blank=True, null=True) #количество

pct\_date\_add = models.DateTimeField(max\_length=255,blank=True, null=True) #Добавил Дата

who\_add = models.CharField(max\_length=255,blank=True, null=True) #Добавил Отвественный

pct\_date\_confirm = models.DateTimeField(max\_length=255,blank=True, null=True) #Подтвердил Дата

who\_confirm = models.CharField(max\_length=255,blank=True, null=True) #Подтвердил Отвественный

class Meta:

managed = False

Обмен данными между клиентом и сервером осуществляется через Django REST Framework (DRF). Для работы со списком затрат был создан класс TransferCostList, отвечающий за:

* получение списка затрат;
* обработку фильтрации, сортировки и пагинации;
* использование кеширования для оптимизации работы (листинг 9).

Листинг 8 — Класс для работы со списком затрат

Листинг 9 – Класс для работы со списком затрат

# Класс для работы со списком переноса затрат

class TransferCostList(generics.ListAPIView):

serializer\_class = transfer\_list\_serialize

pagination\_class = ItemPagination

# Получение данных с учётом фильтров

def get\_queryset(self):

pp\_id = self.request.query\_params.get('pp\_id', '60')

start\_date\_str = self.request.query\_params.get('start\_date', None)

end\_date\_str = self.request.query\_params.get('end\_date', None)

designation = self.request.query\_params.get('desig', None)

pas\_no = self.request.query\_params.get('pas\_no', None)

sorting = self.request.query\_params.get('sort\_by', None)

revers = bool(self.request.query\_params.get('revers', None))

update = bool(self.request.query\_params.get('update', None))

cache\_key = f"passport\_cost\_transfer\_{pp\_id}\_{start\_date\_str}\_{end\_date\_str}\_{designation}\_{pas\_no}\_{sorting}\_{revers}\_{update}"

# Работа с кэшом и получения списка

queryset = cache.get(cache\_key)

if not queryset or update:

in\_param = json.dumps({"pp\_id": int(pp\_id)})

queryset = get\_passport\_cost\_transfer\_list(in\_param)

if not queryset:

queryset = []

cache.set(cache\_key, queryset, timeout=300)

# Фильтрация данных по заданным параметрам

if sorting:

try:

if sorting in ('pct\_date\_add', 'pct\_date\_confirm'):

queryset = sorted(queryset, key=lambda x: datetime.datetime.strptime(

x[sorting] if x[sorting] is not None else "01.01.0001 00:00:00", "%d.%m.%Y %H:%M:%S"),

reverse=not revers)

elif sorting in ('who\_add', 'who\_confirm'):

queryset = sorted(queryset, key=lambda x: (

x[sorting] is None, full\_user\_split(x[sorting])), reverse=revers)

elif sorting in ('pas\_no\_full\_to', 'pas\_no\_full\_from'):

queryset = sorted(queryset, key=lambda x: (x[sorting] is None, pass\_no\_split(x[sorting])),

reverse=not revers)

elif sorting == 'pct\_quantity':

queryset = sorted(queryset, key=lambda x: (

x[sorting] is None, x[sorting]), reverse=not revers)

else:

queryset = sorted(queryset, key=lambda x: (

x[sorting] is None, x[sorting]), reverse=revers)

except Exception as e:

print(f'Ошибка в сортировке: {e}')

# Фильтрация по номеру паспорта

if pas\_no:

queryset = [item for item in queryset if pas\_no in (

item['pas\_no\_full\_to'], item['pas\_no\_full\_from'])]

# Фильтрация по номенклатуре

if designation:

queryset = [item for item in queryset if designation ==

item['nom\_designation']]

# Фильтрация по диапазону дат

if start\_date\_str and end\_date\_str:

try:

start\_date = datetime.datetime.strptime(

start\_date\_str, "%Y-%m-%d")

end\_date = datetime.datetime.strptime(end\_date\_str, "%Y-%m-%d")

queryset = [

item for item in queryset

if start\_date <= datetime.datetime.strptime(item['pct\_date\_add'], "%d.%m.%Y %H:%M:%S") <= end\_date

]

except Exception as e:

print(f'Ошибка в форматировании даты: {e}')

global clean\_queryset

clean\_queryset = queryset

# create\_pdf(self.request) #meow

return queryset

Для удобного представления данных используется сериализатор transfer\_list\_serialize, который преобразует объекты модели в формат JSON (листинг 10).

Листинг 10 - сериализатор transfer\_list\_serialize

from rest\_framework import serializers

from .models import transferList

class transfer\_list\_serialize(serializers.ModelSerializer):

pas\_no\_full\_to = serializers.CharField(allow\_null=True) # Номер паспорта Основной

pas\_no\_full\_from = serializers.CharField(allow\_null=True) # Номер паспорта Источник

nom\_designation = serializers.CharField(allow\_null=True) # ОбозначениеНОВОЕ

nom\_name = serializers.CharField(required = False, allow\_null=True) # Наименование

nom\_size = serializers.CharField(required = False, allow\_null=True) # ТипоРазмер

pct\_quantity = serializers.IntegerField(allow\_null=True) # Количество

pct\_date\_add = serializers.DateTimeField(allow\_null=True) # Добавил Дата

who\_add = serializers.CharField(allow\_null=True) # Добавил Отвественный

pct\_date\_confirm = serializers.DateTimeField(allow\_null=True) # Подтвердил Дата

who\_confirm = serializers.CharField(allow\_null=True) # Подтвердил Отвественный

class Meta:

read\_only = "true"

model = transferList

fields = ['pas\_no\_full\_to', 'pas\_no\_full\_from','nom\_designation','nom\_designation','nom\_name',

'nom\_size','pct\_quantity','pct\_date\_add','who\_add','pct\_date\_confirm','who\_confirm']

firm']

Выделение ресурсов на этапе проектирования играет ключевую роль в создании масштабируемых и гибких веб-приложений. В рамках разработки модуля transfer\_cost были определены основные ресурсы и API-ресурсы обеспечивающие работу сервиса.

Основные API-ресурсы:

* /TransferMain/ — просмотр списка переноса затрат;
* /TransferMain/Analytics/ — просмотр аналитики.

Внутренние API-ресурсы (служебные):

* /TransferMain/LoadTable/ — отправка данных на сервер и получение списка переноса затрат через Django REST Framework;
* /TransferMain/TransferCost/ — обработка запросов, связанных с переносом затрат;
* /TransferMain/Analytics/data/ — передача данных для аналитики и их получение в формате JSON;
* /TransferMain/pdf/ — генерация и скачивание PDF-файла со списком переноса затрат.

## 2.3 Проектирование клиентской части

Разработка клиентской части играет ключевую роль в создании удобного и эффективного веб-приложения. Основная задача клиентской части — обеспечить интуитивно понятный интерфейс и корректное взаимодействие пользователя с сервером.

Основные этапы разработки клиентской части:

1. Создание пользовательского интерфейса (UI)
   * Разработка макетов веб-страниц.
   * Описание стилей с использованием CSS.
   * Формирование структуры интерфейса для удобного взаимодействия пользователя с системой.
2. Настройка пользовательского опыта (UX)
   * Обеспечение корректного поведения элементов интерфейса.
   * Проверка и валидация вводимых данных.
   * Организация удобного взаимодействия пользователя с системой.
3. Настройка взаимодействия с сервером
   * Передача данных между клиентом и сервером с использованием Fetch API.
   * Реализация логики фильтрации, сортировки и пагинации.

Использование шаблонов Django

Django позволяет использовать родительские и дочерние шаблоны, что упрощает организацию кода интерфейса. Родительский шаблон содержит базовую структуру HTML-документа, а дочерние шаблоны дополняют его и переопределяют отдельные блоки кода. Это позволяет сократить дублирование и сделать проект более удобным в поддержке.

Примеры родительских и дочерних шаблонов приведены в [Приложение].

Стартовая страница

Стартовая страница представляет собой первую страницу, с которой начинается работа пользователя с приложением. Она содержит:

* Логотип «Росатома», размещённый в центральной части экрана.
* Кнопку перехода на основную страницу, выполненную в виде крупного интерактивного элемента, обеспечивающего быструю навигацию.

Дизайн стартовой страницы минималистичен, чтобы не перегружать пользователя информацией и сосредоточить его внимание на переходе к основному функционалу.

Основная страница

Основная страница представляет собой рабочее пространство приложения, в котором отображается информация о переносах затрат.

Вывод данных осуществляется в виде таблицы, где предусмотрены:

* Фильтрация по номеру сопроводительного паспорта и обозначению.
* Пагинация (по 100 записей на страницу).
* Сортировка данных по убыванию или возрастанию.

Функциональность клиентской части

* Фильтры — кнопка «Показать/скрыть фильтры» открывает список фильтров.
* Добавление записей — реализовано через модальное окно с полями для ввода номеров сопроводительных паспортов. Введённые данные проходят валидацию.
* Удаление и подтверждение записи — кнопки остаются неактивными (заблокированы и отображаются с уменьшенной контрастностью) до выбора пользователем нужной записи. После выбора запись выделяется, а кнопки активируются.
* Скачивание PDF — кнопка формирует PDF-файл с данными, учитывая фильтры и сортировку.
* Аналитика — кнопка открывает страницу аналитики.

Страница аналитики

На странице аналитики пользователь может ввести номер заказа и временной промежуток, чтобы:

* Получить диаграмму соотношения паспортов с перенесёнными затратами к их общему количеству.
* Включить специальный флажок, который переключает отображение с процентного отношения на абсолютные числовые значения (так как процент перенесённых затрат обычно мал).

Вывод

В соответствии с техническим заданием в данной главе:

* Спроектирована база данных.
* Разработана серверная часть приложения.
* Реализована клиентская часть с необходимыми функциями.

# ГЛАВА 3 ИСПОЛЬЗОВАНИЕ ПРИЛОЖЕНИЯ

## 3.1 Руководство пользователя. Тестирование приложения.

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

Программный продукт «Сопровождения переноса затрат между сопроводительными паспортами в ИС МКАДД» представляет собой приложение, содержащие в себе клиентскую часть в виде [web-приложения], функционирующее в [браузере](как-то по умному можно по идее описать) на рабочем месте [сотрудника].

Установка клиентской части на компьютер не требуется, так как система функционирует как web-приложение.

Запуск модуля осуществляется путём перехода по ссылке [установленной администратором] на рабочем месте.

[home page]

Основная страница: после успешного после успешного входа в систему (перехода по ссылке), в браузере должна отобразиться HTML-страница [(рисунок)].

Открыв ссылку в браузере, встречает стартовая страница, имеющая общее описание модуля и кнопку перехода на страницу списка переноса затрат.

Страница переноса затрат представляет собой таблицу с данными о переносах затрат, кнопки фильтрации добавления, удаления, подтверждения заявки на перенос затрат, так же кнопку на переход страницу аналитики и кнопки для скачивания pdf файла содержащий в себе полный список переноса затрат по указанным фильтрам, сортировки. В таблице имеется возможность сортировки содержимого по столбцам таблицы, а именно: по возрастанию, убыванию, в алфавитном порядке.

Кнопка показать/скрыть фильтры, [выдвигает] пользователю меню фильтров. Главный фильтр, по которым выводится таблица с данными это учётная точка. Поля ввода номера паспорта и обозначения представляют собой фильтры, в частности поле ввода номера паспорта требует ввода полного правильного номера паспорта, если же пользователь ошибётся в написание паспорта, выведется сообщение, где предположительно находится ошибка[фото]. Так же в фильтры входят поля ввода даты, дату можно ввести как с клавиатуры, так и в интерактивном календаре нажав на нужную дату и в случае чего можно отредактировать выбранную дату, путём ввода нужных чисел. Кнопка отфильтровать применит нужные фильтры, и таблица обновится. Кнопка сбросить все фильтры, не трогая учётную точку и обновит таблицу согласно ведённой учётной точке. Так же в этом меню предусмотрены отслеживание нажатия кнопки Enter и Esc, нажатие Enter выполняет роль нажатия кнопки отфильтровать, нажатие Esc скрывает меню фильтров.

По нажатию на кнопку Добавить открывает модальное окно [скриншот], окна с полями ввода двух полных номеров паспортов для создания заявки на перенос. Поля ввода так же требуют правильности ввода полного номера паспорта, аналогично как с поле фильтра по номеру паспорта в меню фильтров. Нажатие на свободное поле от окна, кнопки крестика справа сверху окна, Отмены и нажатия клавиши Esc закрывают данное окно сохраняя введённые данные. После нажатия на кнопку Добавить в окне, окно закроется и отобразится индикация загрузки. После чего если всё успешно справа снизу вылезет уведомление, заявка успешно создана, и таблица обновится, и новая запись будет выделена жёлтым цветом т.к. данная заявка ещё не подтверждена. Если что-то пойдёт не так, вылезет модально окно с ошибкой.

Кнопки Удалить и Подтвердить распространяются, на не подтверждённые записи, такие записи имеют пустые поля под Подтвердил Дата/Ответственный и выделяются жёлтым цветом. После нажатия на любую часть не подтвержденной записи, перестают быть заблокированы (становятся нажимаемыми и возвращают свой цвет обратно). После нажатию на любую из этих кнопок вылезет окно с подтверждением действия, удалить либо подтвердить заявку на перенос. Отмена подтверждения действия может выполнится как нажатием на свободную область от окна, так и нажатием на кнопку Отмена. После чего справа снизу появится уведомление от том, что отверждения отменено. В случае подтверждения действия, окно закроется и появится иконка загрузки, после ответ от сервера вы получите либо уведомление от том, что всё успешно справа снизу и таблица обновится, либо сообщение в окне об ошибке.

Кнопка Аналитика, перекидывает на страницу аналитики.

Кнопка Скачать pdf, даёт загрузить пользователю pdf содержащий в себе таблицу с данными о переносах затрат по указанным фильтрам и сортировкам.

Кнопки Предыдущая и Следующая. Кнопка Следующая разблокирована, если список по текущей учётной точке содержит больше 100 записей и, если на следующей странице есть записи. Кнопа Предыдущая заблокирована, до тех пор\, пока не будет нажата кнопка Следующая и, если это не первая страница.

Страница аналитики. Предоставляет диаграмму паспортов с перенесенными затратами внутри одного заказа по отношению к общему количеству паспортов, после ввода в соответствующие поля номера заказа, временного промежутка и нажатия на кнопку Показать аналитику, после чего появится иконка загрузки до ответа сервера, если всё хорошо, то отобразится диаграмма, если что-то не так, в окне будет показано советующие уведомление. Так же под полями ввода даты, находится флаг по умолчанию в включенном состоянии, что означает что, в диаграмме будут отображено процентное соотношение если же переключить данный флаг, будут отображены численные значения столбцов. Так же в этой области по нажатию на клавишу Enter, будет аналогичное действие как нажатие на кнопку Вывести аналитику. Кнопка выйти, перекидывает обратно на страницу переноса затрат.

## 3.2 Планируемый эффект от внедрения

Внедрение программы «Сопровождение переноса затрат между сопроводительными паспортами в ИС МКАДД» является частью перехода на импортонезависимое программное обеспечение и обеспечения кроссплатформенности системы.

Внедрение модуля может быть осуществлено сразу после завершения разработки и защиты дипломного проекта, что позволит частично подготовить инфраструктуру к переходу на импортонезависимые операционные системы, такие как Astra Linux. Альтернативным сценарием является ожидание полной миграции всей системы МКАДД на веб-платформу, после чего интеграция разработанного модуля станет частью комплексного обновления системы.

Таким образом, основные ожидаемые эффекты от внедрения модуля включают:

* Повышение доступности и удобства использования, так как пользователи смогут работать с модулем через веб-интерфейс без привязки к конкретной операционной системе;
* Создание задела для дальнейшей модернизации системы, что упростит возможную миграцию других компонентов ИС МКАДД в веб-среду.

Для оценки эргономичности пользовательского интерфейса можно отметить следующее:

* 1. Стартовая страница [скриншот] содержит логотип организации и кнопку для перехода в основное приложение. Она выполнена в минималистичном стиле, что соответствует требованиям удобства использования.
  2. Основная рабочая область [скриншот] представляет собой таблицу с функциональными кнопками (добавление, удаление, подтверждение, аналитика, экспорт). Структура таблицы логична, однако возможны улучшения в части визуального выделения ключевых данных.
  3. Раздел аналитики [скриншот] предусматривает ввод параметров и отображение диаграммы, что способствует быстрому анализу данных. Возможны доработки в части визуального представления результатов анализа.

В целом интерфейс программы соответствует требованиям эргономичности, однако для повышения удобства работы могут быть внедрены дополнительные визуальные и функциональные улучшения.

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

* Разработано руководство пользователя;
* Оценен планируемый эффект от внедрения.

**ЗАКЛЮЧЕНИЕ**

[В процессе работы над проектом была изучена предметная область, основные термины, инструментальные средства и целевая аудитория. Состоялось знакомство с ФГУП «Комбинат «Электрохимприбор» и частью его бизнес-процессов. Построены модели бизнес-процессов, сделаны выводы о необходимости создания приложения, предназначенного для управления задачами, стоящими перед пользователями системы.

Для реализации приложения была разработана база данных, на ее основе – необходимые хранимые процедуры, клиентская часть, состоящая из HTML-документов. В результате выполнения выпускной квалификационной работы было разработано импортонезависимое прикладное приложение с использованием веб-технологии, имеющее все необходимые бизнес-функции.]

# Обозначения и сокращения

* ИС — информационная система
* СП — сопроводительный паспорт
* ПДУ — планово диспетчерский участок
* СЗ — сменное задание
* Маршрут — документ, определяющий последовательность операций изготовления продукции
* ПИ — предварительное извещение
* ТД — техническая документация
* УТК — участок технического контроля
* СТО — стандарт организации
* МОЛ — материально ответственное лицо
* ЛЗК — лимитно заборная карта