

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-ОСЕТИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ КОСТА ЛЕВАНОВИЧА ХЕТАГУРОВА»**

Факультет: Физико-технический
Кафедра: Геофизики и геоинформатики

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«Проектирование базы данных, и разработка сервиса для хранения и обработки данных о взаимосвязи компонентов установки SPD»

Исполнитель:

Бакалавр, 4 курс, очная форма обучения,

Направление подготовки: 03.03.02.
Физика

Гуссалов Сослан Вадимович

Научный руководитель:

д.ф.м.н., профессор, заведующий
кафедры геофизики и
геоинформатики

Заалишвили Владислав Борисович

Научные консультанты:

к.ф.м.н., с.н.с. лаборатории ядерных
проблем ОИЯИ

Прокошин Фёдор Валерьевич

к.ф.м.н., Тваури Инга Васильевна

«Допущен к защите»

Зав. кафедрой _____ (профессор, доктор физико-математических наук,
заведующий Магкоев Тамерлан Таймуразович)

«_____» _____ 2025г.

Владикавказ 2025

СОДЕРЖАНИЕ

РЕФЕРАТ.....	3
ВВЕДЕНИЕ.....	4
Глава 1. Эксперименты в физике элементарных частиц и их установки.....	6
1.1. Введение в экспериментальную физику элементарных частиц.....	6
1.2. Основные типы экспериментальных установок. Ускорители частиц.....	7
1.3. Ускорительный комплекс NICA.....	8
Вывод по главе 1.....	13
Глава 2. Базы данных.....	14
2.1. Типы базы данных.....	14
2.1.1. Реляционные базы данных.....	14
2.1.2. Нереляционные базы данных.....	15
2.1.3. Графовые базы данных.....	16
2.2. PostgreSQL.....	16
2.3. Атомарные транзакции.....	17
Вывод по главе 2.....	19
Глава 3. SPD: информационные системы и базы данных.....	20
3.1. Детектор SPD.....	20
3.2. Информационная система эксперимента SPD.....	26
3.3. Информационная система Haradware Database.....	27
Вывод по главе 3.....	28
Глава 4. Сервиса для хранения и обработки данных о взаимосвязи компонентов установки SPD.....	29
4.1. Информационная система Hardware Databases.....	29
4.1.1. Frontend-часть HWDB.....	29
4.1.2. Backend: Сервис для хранения информации о компонентах установки SPD.....	31
4.2. Запросы к базе данных.....	32
4.3. Уровни устройства сервиса.....	33
4.4. Структура базы данных.....	35

4.5. Бизнес логика.....	37
4.6. Обработка HTTP запросов.....	38
4.7 Frontend: Визуализация связей между компонентами.....	41
Вывод по главе 3.....	44
ЗАКЛЮЧЕНИЕ.....	45
СПИСОК ЛИТЕРАТУРЫ.....	46

РЕФЕРАТ

Дипломная работа на тему «Проектирование базы данных, и разработка сервиса для хранения и обработки данных о взаимосвязи компонентов установки SPD» состоит из введения, 3 разделов, выводов, заключения и списка используемых источников, который включает в себя n ссылок. Работа изложена на 45 страницах печатного текста, таблиц и рисунков.

В работе рассматривается эксперимент Spin Physics Detector (SPD), предназначенной для исследований спиновой физики на коллайдере NICA.

Целью данного исследования является проектирование базы данных, и разработка сервиса для хранения и обработки данных о взаимосвязи компонентов установки SPD.

Основные задачи исследования выглядят следующим образом:

1. Изучение экспериментальных установок для детектирования частиц.
2. Проектирование базы данных для хранения информации о взаимосвязях компонентов.
3. Разработка веб интерфейса для визуализации связей компонентов.

Объектом исследования является сервис для хранения информации о взаимосвязях компонентов установки SPD.

Предмет исследования: системы разрабатываемые для хранения и предоставления данных о взаимосвязях компонентов установки SPD.

Ключевые слова: hardware, database, SPD, NICA, mapping, детекторы, ускоритель частиц, физика высоких энергий.

ВВЕДЕНИЕ

Spin Physics Detector (SPD) — это эксперимент на ускорительном комплексе NICA в ОИЯИ, предназначенный для исследований в области спиновой физики с поляризованными пучками. Он должен состоять из нескольких небольших систем, которые должны использовать различные типы детекторов: дрейфовые камеры и трубки, MRPC, Черенковские счетчики, сцинтилляционные и кремниевые детекторы. Сигналы с детекторов должны собираться системой сбора данных, имеющей несколько сотен тысяч каналов. Каждый компонент детектора и системы DAQ должен будет иметь свои наборы параметров и настроек, которые необходимо сохранять для использования при эксплуатации и обслуживании этих систем и особенно для передачи знаний между членами команды. Некоторые из этих данных также должны использоваться для обработки и анализа собранных данных.

Разрабатывается система для хранения информации о соединениях между компонентами и предоставления ее пользователям и другим системам. Она необходима для обеспечения правильной идентификации происхождения сигналов от компонентов детектора.

Основные задачи исследования заключаются в следующем:

1. Изучение экспериментальных установок для детектирования частиц;
2. Проектирование базы данных для хранения информации, где указаны все взаимосвязи компонентов;
3. Разработка средства автоматизированного заполнения данных;

Предметом исследования представляет собой методы и программно-технические решения (архитектура, интерфейсы, модули), необходимые для реализации основных задач исследования.

Результаты проведенных исследований были представлены в виде докладов на Весенней школе по информационным технологиям ОИЯИ 2024 [1, 2, 3].

Глава 1. Эксперименты в физике элементарных частиц и их установки

1.1 Введение в экспериментальную физику элементарных частиц

Элементарная физика элементарных частиц изучает фундаментальные составляющие материи и их взаимодействия, используя высокоточные измерения и сложные технические установки. Основной целью является проверка теоретических моделей: Стандартной модели, описывающей элементарные частицы и фундаментальные взаимодействия (электрослабые и сильные), а также поиск явлений, выходящих за ее пределы, включая суперсимметрию, темную материю или дополнительные измерения. Эксперименты направлены на определение физических характеристик частиц, таких как масса, заряд, спин, время жизни, а также параметров их взаимодействий: сечений процессов, констант связи и вероятности распадов. При помощи этих измерений можно уточнять фундаментальные параметры Стандартной модели.

Экспериментальная физика элементарных частиц связана с квантовой теорией поля, которая предоставляет теоретическую основу для предсказания свойств частиц и их взаимодействий. Эксперименты проектируются для проверки предсказаний квантовой электродинамики, квантовой хромодинамики и электрослабой теории. Основными целями являются:

- 1) Достижение высоких энергий для создания массивных частиц
- 2) Подавление фоновых процессов, обусловленных космическими лучами или вторичными взаимодействиями
- 3) Обработку больших объемов данных

Для решения этих задач применяются ускорительные комплексы, способные разгонять частицы до релятивистских энергий, и детекторные системы, регистрирующие продукты столкновений с высокой точностью.

Современные экспериментальные установки включают ускорители частиц, детекторы и специализированные обсерватории, такие как нейтринные детекторы. При помощи происходят столкновения частиц при

высоких энергиях, что позволяет изучать процессы, недоступные в естественных условиях. Детекторы идентифицируют вторичные частицы, фиксируют их траектории, определяют энергии и другие параметры. Обработка данных происходит при помощи статических методов для выделения сигнала на фоне шума, интерпретации результатов.

Особо важным является достижение высокой точности измерений, что требует минимизации систематических и статистических погрешностей. К примеру, измерение массы бозона Хиггса с точностью до десятых долей процента позволяет проверить согласованность Стандартной модели. Эксперименты также направлены на изучение редких процессов с малыми сечениями, таких как распады с нарушением CP-инвариантности, что требует высокой интенсивности пусков и чувствительности детекторов.

1.2 Основные типы экспериментальных установок Ускорители частиц

Ускорители частиц являются основным инструментом экспериментальной физики элементарных частиц. Ускорители делятся на циклические и линейные ускорители, различающихся по конструкции, принципам работы и областям применения.

Линейные ускорители ускоряют частицы вдоль прямолинейных траекторий с использованием высокочастотных электрических полей, создаваемых в резонаторных камерах. Из-за отсутствия кольцевой геометрии минимизируются потери энергии на синхротронное излучение. Энергия частиц в линаронах определяется длиной ускорительной трасы и градиентом ускоряющего поля измеряемого в МэВ/м.

Примером линейного ускорителя является Стэнфордский линейный ускоритель (SLAC) в США, имеющий длину около 3 км. SLAC ускоряет электроны и позитроны до энергий порядка 50 ГэВ и используется для изучения структуры адронов, электрослабых взаимодействий и редких процессов. Эксперименты на SLAC подтвердили кварковую модель протонов

и нейтронов, а также внесли вклад в изучение асимметрии в слабых взаимодействиях.

Циклические ускорители имеют кольцевую геометрию, где частицы удерживаются на орбитах при помощи сверхпроводящих магнитов, генерирующие магнитные поля порядка нескольких Тесла. Засчет многократного ускорения удается достичь очень высоких энергий, существенно больше чем на линейных ускорителях.

Коллайдеры представляют собой ускорительные комплексы, в которых два пучка частиц ускоряются в противоположных направлениях и сталкиваются в специально спроектированных точках. Высокая энергия в системе центра масс позволяет создать массивные частицы, такие как бозон Хиггса, топ-кварк или гипотетические частицы за пределами Стандартной модели. Основными параметрами коллайдера являются:

- 1) Энергия столкновений
- 2) Интенсивность пучков
- 3) Фокусировка пучков для увеличения вероятности взаимодействия

Примером коллайдера является Большой адронный коллайдер (LHC) в ЦЕРН, Швейцария. LHC Ускоряет протоны до энергий ТэВ и сталкивает их в четырех точках, оснащенных детекторами ATLAS, CMS, ALICE и LHCb. Детекторы ATLAS и CMS предназначены для поиска новой физики, включая частицы суперсимметрии и прецизионных свойств бозона Хиггса. ALICE специализируется на изучении кварк-глюонной плазмы, формирующейся при столкновениях тяжелых ионов, а LHCb фокусируется на физике b- кварков и нарушении CP-инвариантности.

1.3 Ускорительный комплекс NICA

Ускорительный комплекс NICA (Nuclear-based Ion Collider Facility) является амбициозным проектом, приводящим к реализации в Объединенном институте ядерных исследований (ОИЯИ) в Дубне. Запуск NICA откроет новые горизонты в физике высоких энергий и исследованиях

фундаментальных характеристик материи. Сложные экспериментальные комплексы и международное сотрудничество нацеливает на изучение структуры веществ в экстремальных условиях, аналогичных тем, что были в ранней Вселенной.

NICA представляет собой современный ускорительный комплекс, основой которого является нуклотрон - сверхпроводящий синхротрон, созданный в ОИЯИ еще в 1990-х годах. Нуклотрон стал отправной точкой для развития комплекса, обеспечив базу для реализации более масштабных задач. Комплекс включает в себя ряд основных частей: инжекционную систему, бустер для начального ускорения частиц, сам нуклотрон, и два кольцевых ускорителя, образующие вместе коллайдер для столкновения ионов. Такая конфигурация позволяет проводить исследования тяжелых ионов, например, золотых ионов, при высоких энергиях, достигая энергии столкновения до 11 ГэВ на нуклон для ионов с массовым числом $A \approx 197$.

Основная цель NICA - исследование фазовых переходов в ядерной материи, особенно перехода от адронной фазы к кварк-глюонной плазме. Кварк-глюонная плазма - агрегатное состояния вещества, при котором кварки и глюоны, обычно связанные внутри протонов и нейтронов, становятся свободными. Считается, что такое состояние существовало в первые микросекунды после большого взрыва. Чтобы воссоздать такие условия в NICA сталкиваются тяжелые ионы на высоких энергиях, что приводит к экстремальным плотностям и температурам.

Комплекс NICA (Рис. 1) включает три основных эксперимента: MPD (Multi-Purpose Detector), SPD (Spin Physics Detector) (Рис. 2) и BM@N (Baryonic Matter at Nuclotron). Каждый из них решает свои задачи и вносит вклад в общую научную программу. MPD - это универсальный детектор, предназначенный для изучения столкновений тяжелых ионов. Оснащен трековыми системами, таких как TPC (Time Projection Chamber), и калориметрами, которые распознают тысячи треков частиц, получаемых в каждом событии. SPD разрабатывается для спиновой физики и исследования

поляризованных частиц, важных для понимания нуклонной структуры и спиновых свойств. BM@N исследует материю с помощью фиксированной мишени, что дает возможность исследовать реакции с высокой плотностью барионного заряда.



Рисунок 1. Ускорительный комплекс NICA, г. Дубна

Одним из основных характеристик NICA является высокая светимость коллайдера, которая обеспечивает большое количество столкновений частиц в единицу времени. Это критически важно для сбора статистически значимых данных. Для достижения высокой светимости применяются новые технологии, такие как стохастическое и электронное охлаждение пучков, минимизирующие рассеяние частиц и увеличивающие плотность пучка. Комплекс оборудован современной системой электронного съема данных, способной обрабатывать гигантские потоки информации, поступающие с детекторов в реальном времени.

Процесс обработки данных в NICA является сложной задачей, требующей использования высших достижений статистического анализа. Реконструкция событий - процедура восстановления траекторий частиц

(треков) по сигналам с детекторов - является одной из центральных задач. В условиях высокой множественности событий, когда при каждом столкновении в результате возникают тысячи треков, традиционные алгоритмы, например, фильтр Калмана, сталкиваются с пределами по скорости и масштабируемости. Поэтому разрабатываются алгоритмы глубокого обучения, включая нейросетевые модели, например, графовые нейронные сети. Эти методы позволяют эффективно отфильтровать шумовые сигналы, распознавать треки и находить вторичные вершины.

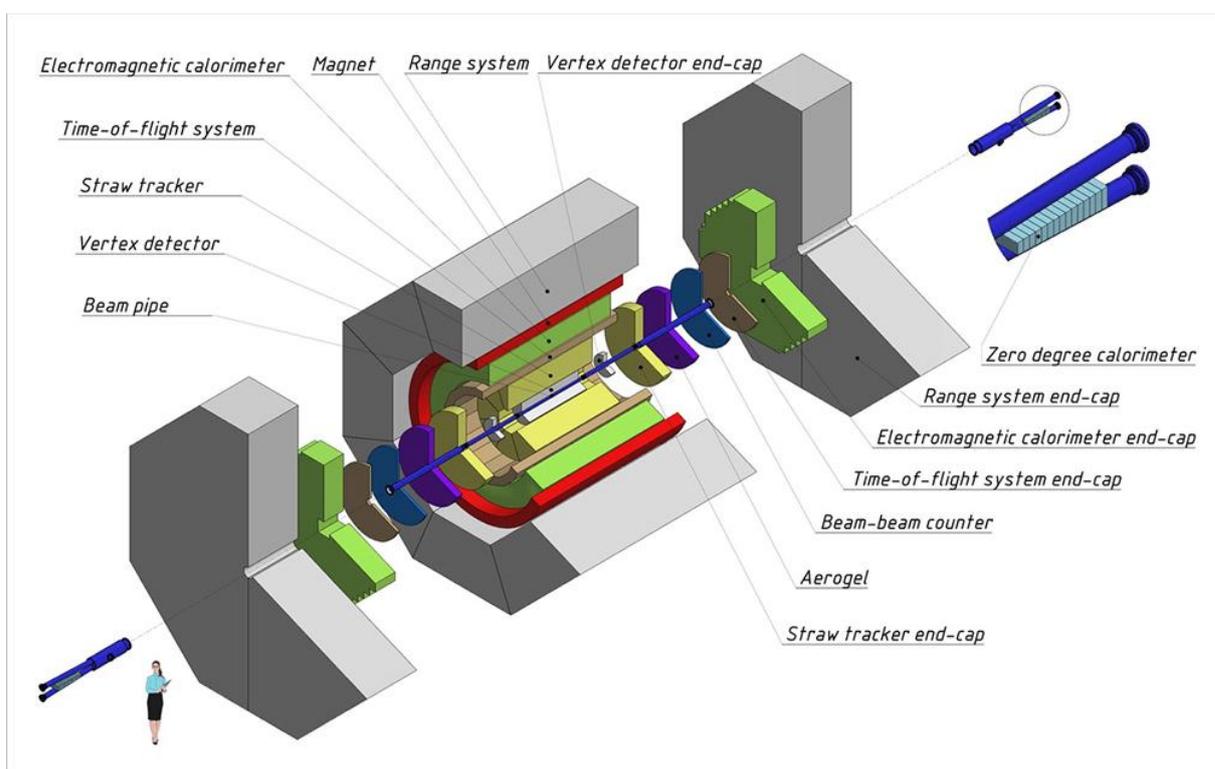


Рисунок 2. Устройство детектора SPD

Еще одной важной составляющей NICA является распределенная вычислительная инфраструктура. Для обработки и хранения огромных объемов данных используются технологии GRID и облачные вычисления, аналогичные тем которые используются в глобальной сети WLCG (Worldwide LHC Computing Grid). Это позволяет распределять вычислительные задачи между различными центрами, обеспечивая высокую скорость обработки и надежность хранения данных. В ОИЯИ также активно развиваются

суперкомпьютерные мощности, включая кластеры на базе GPU, которые ускоряют выполнение ресурсоемких задач, таких как моделирование физических процессов и обучение нейросетей.

NISA решает не только основные научные задачи, но и имеет практическое значение. Технологии, которые разрабатываются для комплекса, применяются в медицине, материаловедении и других областях. Например, доступные для экспериментов NISA методы обработки данных и алгоритмов машинного обучения можно использовать для анализа медицинских изображений. Работы в физике высоких энергий также способствуют развитию новых материалов и технологий, которые могут быть применены в промышленности.

Вывод по главе 1

В первой главе рассмотрены основные аспекты экспериментальной физики частиц, включая ее цели, методы и основные установки. Эксперименты в этой области направлены на проверку предсказаний Стандартной модели, а также поиска новой физики (суперсимметрия, темная материя)

Основными инструментами исследований являются ускорители частиц (линейные и кольцевые ускорители) а также детекторные системы. Коллайдеры такие как LHC обеспечивают высокие энергии столкновений, которые необходимы для рождения массивных частиц, в то время как линейные ускорители (к примеру SLAC) минимизируют энергетические потери и предназначены для изучения структуры адронов и редких процессов.

Особое внимание уделено ускорительному комплексу NICA, который является современной установкой для изучения кварк-глюонной плазмы и фазовых переходов в ядерной материи. Комплекс включает в себя детекторы MPD, SPD и BM@N. Каждый из них решает отдельные задачи связанные с физикой тяжелых ионов, спиновыми эффектами и барионной материей. NICA имеет высокую светимость, передовые методы обработки данных (включая машинное обучение) и распределение вычислительной системы.

Глава 2. Базы данных

2.1 Типы базы данных

База данных (БД) — это некий объем перенесенной в электронный вид информации, которую хранят на специальных компьютерных серверах. По ряду причин работать с такими базами удобнее и проще, чем с традиционными таблицами:

1) БД позволяют эффективно обрабатывать, хранить и структурировать объемы информации, которые стали настолько большими, что их не «вытягивают» обычные таблицы.

2) БД доступны большому количеству пользователей. Удаленный доступ к базе данных и используемая система запросов обеспечивает возможность одновременной работы большого количества человек. Электронные таблицы также позволяют одновременно работать нескольким пользователям, но в случае с БД это происходит быстрее и проще.

3) Базы данных управляются при помощи специального ПО или (СУБД).

2.1.1 Реляционные базы данных

Реляционными называются базы данных, в основе построения которых лежит реляционная модель.

Данные в реляционных структурах организованы в виде набора таблиц, называемых отношениями, состоящих из столбцов и строк. Каждая строка таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности. Каждая строка в таблице может быть помечена уникальным идентификатором, называемым первичным ключом, а строки из нескольких таблиц могут быть связаны с помощью внешних ключей.

Особенности реляционных БД:

- 1) Модель данных в реляционных БД определена заранее и является строго типизированной
- 2) Данные хранятся в таблицах, состоящих из столбцов и строк
- 3) На пересечении каждого столбца и строки допускается только одно значение
- 4) Каждый столбец переименован и имеет определенный тип, которому следуют значения со всех строк в данном столбце
- 5) Столбцы располагаются в определённом порядке, который определяется при создании таблицы
- 6) В таблице может не быть ни одной строчки, но обязательно должен быть хотя бы один столбец

2.1.2 Нереляционные базы данных (NoSQL)

Нереляционная база данных — это база данных, в которой в отличие от большинства традиционных систем баз данных не используется табличная схема строк и столбцов. В этих базах данных применяется модель хранения, оптимизированная под конкретные требования типа хранимых данных. Например, данные могут храниться как простые пары "ключ — значение", документы JSON или граф, состоящий из ребер и вершин.

Термин NoSQL применяется к хранилищам данных, которые не используют язык запросов SQL. На практике NoSQL означает "не реляционная база данных", даже несмотря на то, что многие из этих баз данных поддерживают запросы, совместимые с SQL. Однако базовая стратегия выполнения запросов обычно отличается от того, как традиционная система управления реляционными базами данных (RDBMS) будет выполнять тот же SQL-запрос.

2.1.3 Графовые базы данных

Графовая база данных – это нереляционный тип баз данных, основанный на топографической структуре сети. Идея этой БД восходит к математической

теории графов. Графы представляют наборы данных в виде узлов, ребер и свойств.

1) Узлы, или точки (nodes) – это экземпляры или сущности данных; ими является любой объект, который вы планируете отслеживать. Например, люди, заказчики, подразделения и т.д.

2) Ребра, или линии (edges) – это важнейшие концепции в графовых БД. Они отображают взаимосвязь между узлами. Эти связи имеют направление и могут быть одно- или двунаправленными.

3) Свойства (properties) содержат описательную информацию, связанную с узлами. В некоторых случаях свойства бывают и у ребер.

2.2 PostgreSQL

PostgreSQL — это объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире. Имеет открытый исходный код и является альтернативой коммерческим базам данных.

СУБД PostgreSQL позволяет гибко управлять базами данных. С ее помощью можно создавать, модифицировать или удалять записи, отправлять транзакцию — набор из нескольких последовательных запросов на языке SQL.

Задачами PostgreSQL являются:

- 1) Гибкий доступ к базам данных, их организация и хранение.
- 2) Управление записями в базах данных: создание, редактирование и удаление
- 3) Просмотр нужной информации из базы данных по запросу
- 4) Настройка и контроль доступа к той или иной информации, группировка пользователей по уровню прав
- 5) Защита информации от утечек и потерь
- 6) Контроль состояния базы

Еще одна из особенностей PostgreSQL — поддержка большого количества типов записи информации. Это не только стандартные

целочисленные значения, числа с плавающей точкой, строки и булевы значения («да/нет»), но и денежный, геометрический, перечисляемый, бинарный и другие типы. PostgreSQL «из коробки» поддерживает битовые строки и сетевые адреса, массивы данных, в том числе многомерные, композитные типы и другие сложные структуры. В ней есть поддержка XML, JSON и NoSQL-баз. При необходимости к СУБД можно подключить поддержку типов данных, которые нужны в конкретном проекте. В PostgreSQL есть несколько внутренних форматов, которые используются только в ней.

PostgreSQL работает со сложными, составными запросами. Система справляется с задачами разбора и выполнения трудоемких операций, которые подразумевают и чтение, и запись, и валидацию одновременно. Она медленнее аналогов, если речь заходит только о чтении, но в других аспектах превосходит конкурентов.

2.3 Атомарные транзакции

Атомарные транзакции представляют собой фундаментальный механизм обеспечения целостности данных в системах, где критически важна согласованность операций. Под атомарностью понимается свойство транзакции, при котором она выполняется как единое целое - либо все ее действия успешно завершаются, либо ни одно из них не применяется. Это исключает возможность частичного выполнения, которое могло бы привести к противоречивому состоянию системы. Термин атомарность заимствован из физики, подчеркивая неделимость процесса, аналогично атому, как неделимой единице вещества.

Основное назначение атомарных транзакций, это гарантировать корректность данных в условиях многозадачности, сбоя оборудования или программных ошибок.

Атомарные транзакции являются ключевым компонентом модели ACID, которая описывает требования к надежным СУБД. В рамках этой модели

атомарность тесно связана с другими свойствами: изоляция гарантирует, что параллельные транзакции не влияют друг на друга, а устойчивость обеспечивает сохранение результатов завершенных транзакций даже после сбоев.

Реализация атомарности основана на механизмах журналирования и отката. Перед выполнением транзакции система фиксирует ее начальное состояние в журнале изменений, если в процессе выполнения возникает ошибка, СУБД использует журнал для отмены всех промежуточных изменений. В распределенных системах, где транзакции затрагивают несколько узлов, применяются протоколы координации, такие как двухфазный коммит. На первом этапе все участники подтверждают готовность к выполнению операции, а на втором либо фиксируют изменения, либо откатывают их при обнаружении проблем.

Вывод к главе 2

Реляционные БД, основанные на табличной структуре с использованием первичных и внешних ключей, обеспечивают строгую типизацию и удобную работу с данными, возвращая результат запросов в виде таблиц. Нереляционные базы данных, предлагают гибкие модели хранения, такие как пары “ключ-значение”, JSON-документы или графы, что делает их подходящими для задач, где традиционные таблицы неэффективны. Графовые базы данных, как подвид NoSQL, оптимизированы для работы со сложными взаимосвязями, представляя данные в виде узлов и ребер.

PostgreSQL – реляционная СУБД с открытым исходным кодом, которая сочетает гибкость, поддержку множества типов данных и высокую производительность при обработке данных.

Глава 3. SPD: информационные системы и базы данных

3.1. Детектор SPD

Детектор спиновой физики (SPD) - одна из ведущих экспериментальных установок, разрабатываемых для коллайдера NICA (Nuclon-based Ion Collider Facility) Объединенного института ядерных исследований (ОИЯИ, Дубна). Основной задачей эксперимента SPD является исследование спиновой структуры нуклонов в взаимодействиях поляризованных протонов, дейтронов, а также ядер водорода. Проект SPD объединяет более 300 ученых из 32 институтов 15 стран. Эксперимент частично предназначен для исследования основных процессов, связанных со спиновыми асимметриями, структурой нуклонов и ролью орбитальных моментов кварков в спин протона.

Основной целью эксперимента SPD является исследование спин зависимых и спин независимых структурных функций нуклона в процессах взаимодействия поляризованных частиц. Ключевые направления физической программы включают измерение спиновых асимметрий (SSA): SPD направлено на измерение фотонных распределений TMD (Transverse Momentum Dependent) и глюонных TMD, которые дают информацию об орбитальном движении кварков и глюонов внутри нуклона. Эти измерения позволяют уточнить вклад орбитальных моментов в спин протона, решая проблему так называемого “спинового кризиса”, когда вклад кварков составляет лишь 30 %, а остальное приписывается к глюонам и орбитальным моментам.

Реакции с рождением чарма: Эксперимент сосредоточен на реакциях с выработкой скрытого (Например: $J/\psi \rightarrow \mu^+\mu^-$) и открытого (Например: $D^0 \rightarrow K^-\pi^+$) чарма. Для идентификации J/ψ используется Range система, а для открытого чарма - электромагнитный калориметр и вершинный детектор, которые позволяют определять первичные и вторичные вершины.

Измерение кварковых TMD, которые в 1970-х годах подверглись исследованию в экспериментах в Фермилабе и Аргонне, обеспечит проверку

правильности работы установки SPD и расширение фазового пространства с помощью детектора большего размера и высотой триггерной частоты.

На первом этапе эксперимента планируется изучить упругое и квазиупругое рождение частиц, таких как пионы, что актуально для калибровки детектора и статистического накопления.

Эксперимент разбит на 2 этапа. Первый этап предполагает эксплуатацию при пониженной светимости и энергии пучков до 3 ГэВ с добавленным вниманием к упругому рассеянию. Второй этап требует полной энергии (до 27 ГэВ) и светимости, что должно обеспечить возможность выполнения исследований с высокой статистикой для реакции с чармом и глюонным TMD

Детектор SPD расположен в южной точке пересечения пучков коллайдера NICA. Предназначен для регистрации частиц с высокой триггерной частотой (до 1 МГц) и малой множественности (около 10 треков на событие). Основные детекторные подсистемы включают в себя:

1) Магнит SPD - сверхпроводящий соленоид диаметром 4 м и чуть более 4 м длиной, созданный по образцу магнита эксперимента PANDA, но увеличенный на 1,5 м в диаметре. Магнит состоит из трех модулей центрального катушечного модуля (150 витков) и двух крайних катушечных модулей (по 300 витков). Поле в центре достигает 1,2 Тл, уменьшаясь на 20% к краю трековой системы. Сверхпроводящий кабель - аналогичен используемому в PANDA, а конструкция включает фланцы с треугольными растяжками, как в ATLAS.

2) Вершинный детектор - предназначен для определения точных координат точки взаимодействия. Для первой части эксперимента запланировано использование цилиндрической микростриповой камеры, изготовленной в ЛЯП (ОИЯИ), с разрешением лучше 10 мкм. На последующее расширение рассматриваются два варианта: стриповый детектор (DSSD) или пиксельный детектор (MAPS).

3) Основной трековый детектор состоит из строу трубок (около 26 000 каналов), разделенных на восемь секторов(октантов). Каждый октант включает 60 двойных слоев трубок, ориентированных под углами 3 градуса для создания стереопроекции, необходимой для восстановления треков. Разрешение по импульсу составляет 2-3%. Трубки также используются для идентификации частиц (PID) по ионизованным потерям, то позволяет разделять пионы и каоны до импульсов 0,5 ГэВ/с с разрешением около 8-14%.

4) Система ранжирования (RANGE) требует наличия адронного калориметра магнитного ярма и мюонной идентификации. Состоит из 20 слоев железа толщиной по 3 см с промежутками 3 см, в которые мини- дрейфовые трубки вставлены для двумерного измерения координат. Общий вес системы составляет примерно 1000 тонн, что соответствует четырем ядерным длинам взаимодействия.

5) Электромагнитный калориметр используется для обнаружения фотонов и электронов, главным образом в реакциях с высокими энергиями (4-6 ГэВ). Из свинцово-сцинтилляционных слоев состоит 200 слоев, 18 радиационных длин общей массой 68 тонн. Разрешение по энергии для электронов составляет около 5% при 1 ГэВ, а временное разрешение - порядка наносекунд.

6) Система времени пролета T0F - обеспечивает идентификацию частиц (пионов, каонов) с импульсами до 1,3-1,5 ГэВ/с с временным разрешением порядка 60пс. Система необходима для реакций с открытым чармом, где нужно разделять частицы по массе.

7) Для идентификации частиц с импульсами до 6 ГэВ/с предусматривается установка аэрогельных Черенковских детекторов с фокусирующей оптикой FAR. Эти детекторы используют аэрогель с увеличивающимся показателем преломления для фокусировки Черенковского излучения. Тесты показали возможность разделения протонов, каонов, пионов, и электронов с разрешением около 3 сигм при 6 ГэВ.

8) Для контроля светимости и поляризации пуска используется:

1. BBC (Beam-Beam Counter): Два кольца из пластиковых сцинтилляторов, расположенные на расстоянии 1,5 метров от точки взаимодействия.

2. MCP-детекторы: Детекторы устанавливаются сразу за детектором для регистрации частиц малых углов.

3. Zero-Degree калориметр: Расположен на расстоянии 13 метров от точки взаимодействия для регистрации нейтральных частиц (фотонов и нейтронов). Состоит из пластиковых сцинтилляторов с электромагнитной и адронной системами.

Система сбора данных (DAQ) SPD разработана для обработки высокой триггерной частоты (до 1 МГц) без использования аппаратного триггера. Вместо этого применяется программный триггер (онлайн-фильтр), который выполняет предварительную реконструкцию треков и отбор событий. Данные с детекторов собираются через платы FE (Front-End) и концентраторы первого уровня, расположенные рядом с детектором. Далее данные передаются по оптоволоконной связи в DAQ-комнату, где компьютеры-билдеры собирают события по временным меткам. Объем данных достигает 20 ГБ/с, что требует значительного дискового пространства (до 120 ПБ для второй фазы).

Проект SPD разделен на две фазы:

1) Первая фаза: Она предполагает использование ограниченного набора подсистем (магнит, Range-система, строу трубки, BBC, MCP) при пониженной энергии пучка (до 3 ГэВ) и светимости. Основное внимание уделяется упругому и квазиупругому рассеянию, а также тестированию детекторов на прямолинейных участках пучка.

2) Вторая фаза: Полная конфигурация детектора с установкой вершинного детектора, ToF, Черенковских детекторов и Zero-Degree калориметра. Энергия в центре масс достигает 27 ГэВ. Эта фаза позволит проводить высокоточные измерения спиновых асимметрий и реакций с чармом.

Эксперимент SPD вносит огромный вклад в изучение спиновой структуры нуклона, уточняя вклад орбитальных моментов кварков и глюонов. Так же он дополняет эксперимент MPD, фокусируясь на процессах с поляризованными пучками и высокой триггерной частотой. Результаты SPD позволят закрыть пробелы в данных, полученных в предыдущих экспериментах (в Протвино и CERN), особенно в диапазоне энергий 10–27 ГэВ. Возможность работы с асимметричными модами (протон-дейтрон) и ядрами расширяет физическую программу.

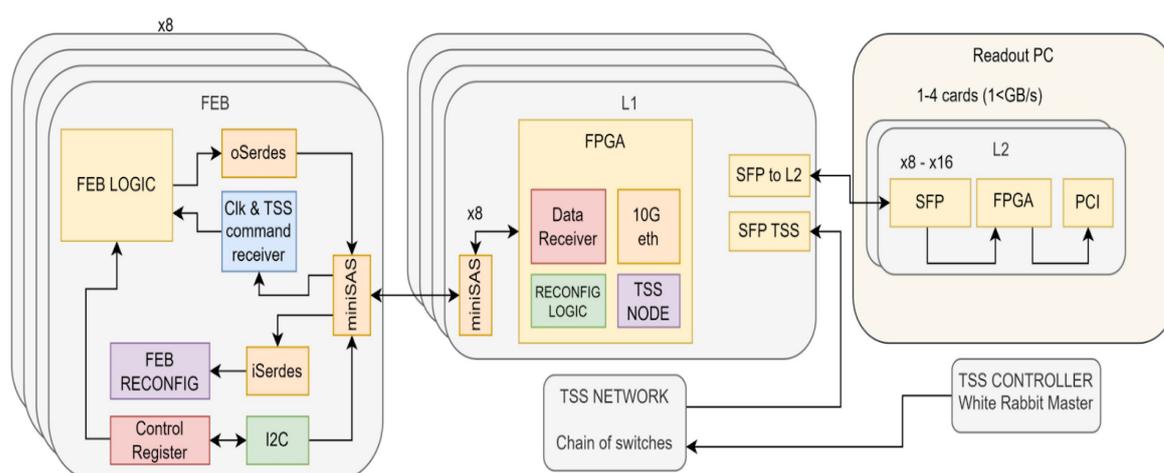


Рисунок 3. Схема цепи считывания данных

3.2. Информационные системы эксперимента SPD

Различные базы данных и информационные системы, разрабатываются для проведения эксперимента СПД:

- 1) База данных метаданных на уровне событий (Event Index);
- 2) Базы данных условий сбора данных и калибровок;
- 3) Системы управления распределенными вычислениями и хранением данных;
- 4) База данных физических метаданных;
- 5) Системы мониторинга;
- 6) Системы протоколирования и учета;

Эти информационно-вычислительные системы находятся на различных стадиях разработки, в основном в самом начале. Проект системы сбора данных находится на стадии планирования, выбор компонентов еще продолжается, поэтому из-за отсутствия реальных данных картографическая база данных разрабатывается на основе тестовых компонентов. Ее можно рассматривать как часть базы данных аппаратуры - каталога компонентов детекторной системы SPD.

База данных аппаратуры SPD разрабатывается как интегрированная информационная система, которая должна обеспечить:

- 1) Получение информации о параметрах аппаратных компонентов и их взаимосвязи с другими компонентами;
- 2) Передачу этой информации и внесение в базы данных;
- 3) Доступ к информации для программ обработки и анализа данных через API и приложения;
- 4) Доступ к информации для пользователей через интерактивные и асинхронные интерфейсы.
- 5) Авторизация в системе с помощью JINR SSO и предоставление прав доступа;

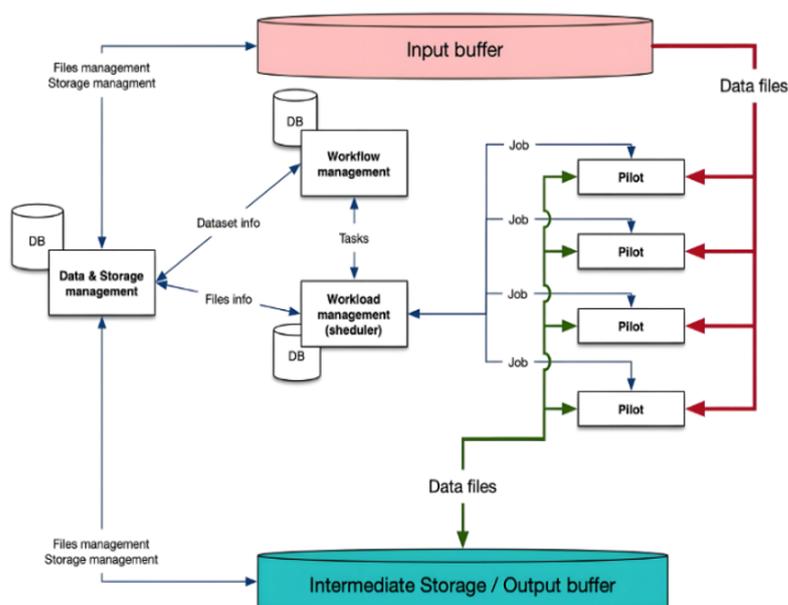


Рисунок 4. Архитектура системы SPD Online Filter

Информационные системы эксперимента SPD:

1. Система сбора данных (Рис. 5) DAQ (Data Acquisition System) – обеспечивает сбор, структурирование и запись данных с субнаносекундной синхронизацией [4]. Технология White Rabbit используется для достижения высокой точности временной маркировки событий. Ожидаемый объем данных, более 10 петабайт в год, что требует создания распределенных хранилищ и конвейеров обработки.

2. SPD Online Filter (Рис. 6) - система для предварительной обработки данных [3], она включает алгоритмы фильтрации шумов и первичного анализа событий. Прототип системы проходит тестирование на стендах, имитирующих реальные экспериментальные условия. Методы машинного обучения - используются для реконструкции треков частиц, для получения наиболее точных результатов применяются методы на основе нейронных сетей Хопфилда. Эти алгоритмы позволяют эффективно обрабатывать сложные события, возникающие при столкновениях поляризованных пучков.

3.3. Информационная система Hardware Database

Информационная система Hardware Database создается для обеспечения стабильной работы, технического обслуживания SPD, а также оптимизации процессов управления данными. Основной целью является формирование единого информационного хранилища, где будут систематизированы параметры, спецификации и конфигурации компонентов установки.

Система предоставит возможность:

- 1) Хранить информацию о типах и экземплярах аппаратных компонентов.
- 2) Отслеживать расположение и перемещение устройств
- 3) Формировать параметры оборудования
- 4) Облегчить техническое обслуживание и передачу информации.

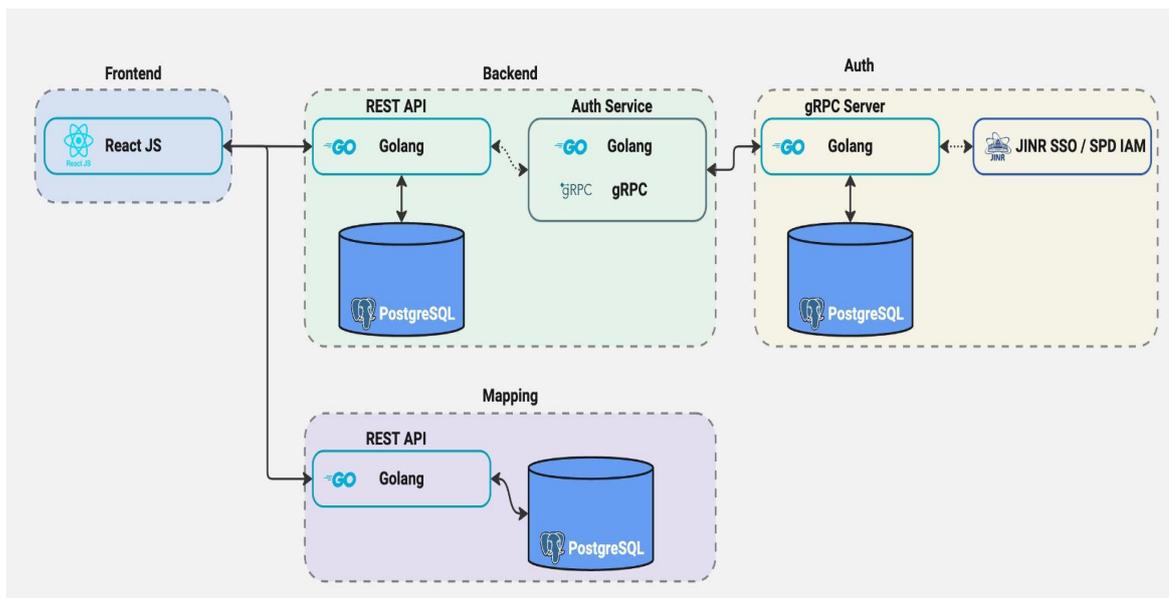


Рисунок 5. Архитектура системы SPD HWDB

HWDB будет иметь вид многокомпонентной системы, которая включает в себя Frontend, Backend, Mapping сервис, а также систему аутентификации пользователей. Взаимодействие между серверными компонентами выполняется при помощи протокола HTTP, посредством REST API.

Каждое устройство имеет уникальный идентификатор (HWID) в системе и относится к определенному типу. Тип устройства задает параметры и их допустимые значения, а также может иметь иерархические связи с другими типами устройств. При помощи этого формализуется конфигурация оборудования и применяются общие правила валидации.

Так как SPD имеет сотни тысяч каналов передачи данных, ручное построение карты подключения невозможно. Поэтому необходима система сопоставления сигналов с компонентами (DAQ Mapping), в которой предусмотрено автоматическое построение схемы на основе иерархии устройств и уникальных идентификаторов, которые передаются по каналу связи.

Вывод по главе 3

В третьей главе рассмотрены ключевые аспекты эксперимента SPD, включая его физическую программу, конструкцию детектора, а также информационные системы, обеспечивающие функционирование. Эксперимент SPD, реализуемый на коллайдере NICA, направлен на изучение спиновой структуры нуклонов.

Конструкция детектора SPD включает в себя сверхпроводящий магнит, вершинный и трековый детекторы, электромагнитный и адронный калориметры, а также системы ToF и Черенковских детекторов. Оптимизирована для регистрации частиц с высокой триггерной частотой и точной идентификацией.

Информационные системы эксперимента, включая базы данных метаданных, условий, калибровок и аппаратуры, находятся на стадии активной разработки. Основной информационной системой является Hardware Database которая станет центральным хранилищем данных о компонентах установки.

Глава 4. SPD: Сервис для хранения и обработки данных о взаимосвязи компонентов установки SPD

4.1. Информационная система Hardware Database

4.1.1. Frontend-часть HWDB

Веб приложение создано при помощи фреймворка React. Основной целью Веб интерфейса является работа с большими массивами данных при помощи интегрированных табличных компонентов. Этот подход облегчает взаимодействие пользователя с информацией, обеспечив удобство навигации, сократив время отклика интерфейса и повысив общую эффективность системы. При помощи TypeScript была минимизирована вероятность ошибок и упрощен процесс сопровождения кода. Использование этих технологий также гарантировало стабильную работу приложения и эффективную систему управления состоянием компонентов.

Система была усилена за счет интеграции механизмов авторизации и аутентификации пользователя, что дало возможность контролировать доступ к данным на основе роли и прав каждого пользователя. Данный подход гарантирует безопасность и строгое соблюдение принципов разграничения доступа. В результате пользователи видят только ту информацию и возможности функционала, которые доступны для их роли.

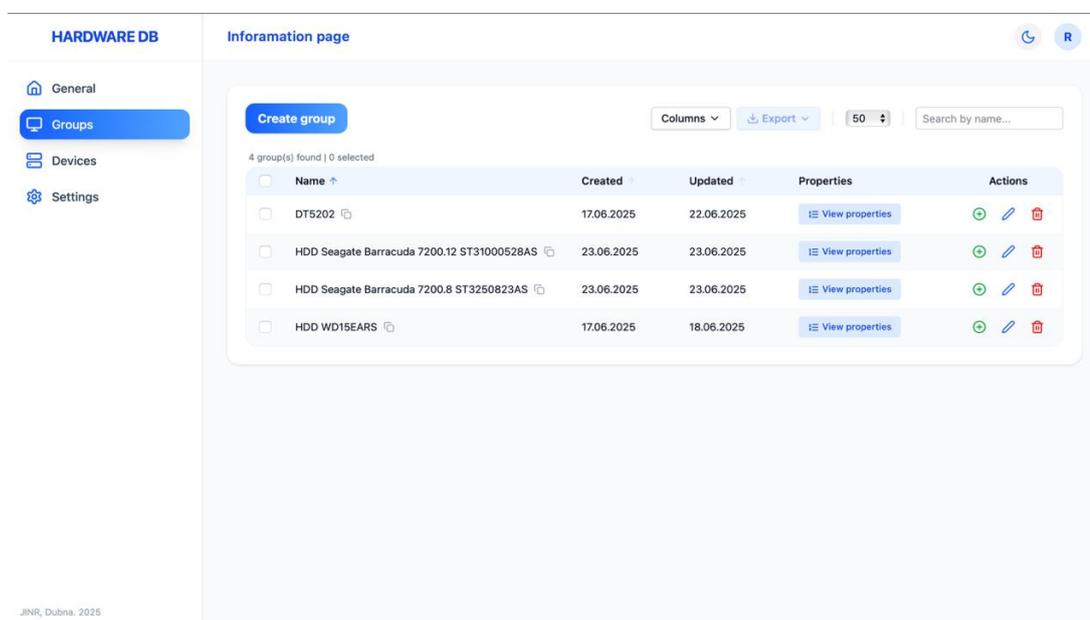


Рисунок 6. Страница Group-properties

Управление состояниями приложения реализовано посредством современных библиотек таких как Redux, React Query, что позволяет добиться высокой согласованности данных и эффективного взаимодействия с серверной частью. Использование таких способов сильно повысило удобство и скорость работы с приложением.

Данное решение является эффективным так как имеет высокую производительность, гибкость и масштабируемость. Структура приложения, основанная на модульной архитектуре, облегчает добавление новых функций и компонентов без существенной перестройки всего проекта. TypeScript предоставила строгую типизацию, позволяющую избежать типичных ошибок при разработке, сокращая время на отладку и тестирование. Реализация продуманного управления состояниями и авторизацией позволила создать надежную и отказоустойчивую систему.

Кроме того, внедрение данных решений улучшило взаимодействие с конечными пользователями, предоставляя им удобные инструменты для повседневной работы. Грамотно спроектированный пользовательский интерфейс минимизирует необходимость обучения пользователя, позволяя быстро освоиться и начать рабочий процесс.

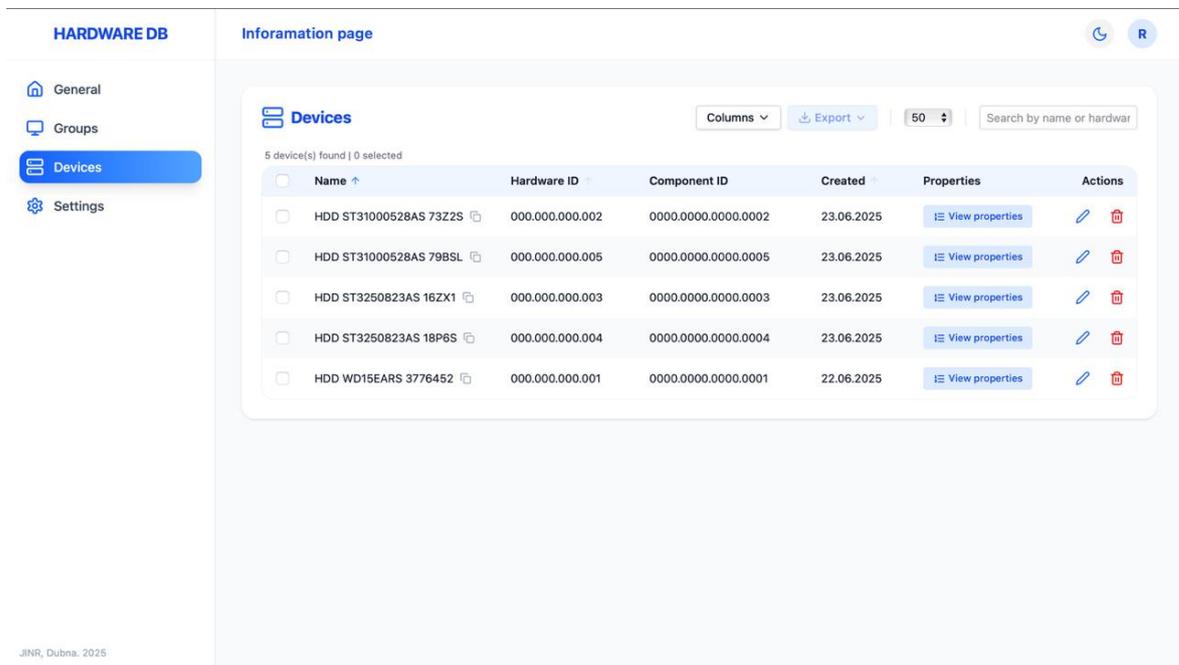


Рисунок 19. Страница Devices

4.1.2. Backend: сервис для хранения информации о компонентах установки SPD

Серверная часть информационной системы имеет модульную архитектуру, состоящую из трех ключевых уровней: API Gateway, Business Logic и Database Requests. API Gateway служит единой точкой входа для всех внешних запросов и обеспечивает их маршрутизацию к соответствующим сервисам. Уровень Business Logic реализует основную предметно-ориентированную логику, включая преобразование данных, выполнение сложных операций и оркестрацию процессов. Этот уровень организован в виде набора независимых модулей, что обеспечивает гибкость и адаптивность системы. Уровень Database Requests абстрагирует работу с хранилищами данных, используя библиотеку pgx для эффективного взаимодействия с PostgreSQL.

Для обеспечения надежности и согласованности данных применяется транзакционная модель, где каждая операция выполняется в рамках атомарной транзакции, что гарантирует целостность данных даже в случае ошибок. Архитектура системы построена с использованием паттернов проектирования, таких как Абстракция и Фасад, что позволяет легко заменять компоненты и масштабировать систему. В качестве языка программирования выбран Golang, который обеспечивает высокую производительность и простоту синтаксиса, что идеально подходит для распределенных систем.

Система также включает в себя микросервис авторизации и аутентификации, который обеспечивает безопасность и контроль доступа. Аутентификация осуществляется через интеграцию с JINR SSO и протоколом OpenID, что позволяет пользователям использовать единую учетную запись для доступа к различным приложениям. Процесс аутентификации включает в себя генерацию одноразовых кодов и JWT-токенов, которые обеспечивают безопасную передачу данных и управление сессиями без необходимости хранения состояния на сервере.

Авторизация реализована на основе ролевых моделей, где права пользователей определяются в зависимости от их роли и групповой принадлежности. Это позволяет гибко управлять доступом и минимизировать риски несанкционированного доступа. Микросервис также поддерживает механизм запроса и модерации прав, что обеспечивает дополнительный уровень контроля.

Технологии gRPC и Protocol Buffers используются для обеспечения эффективного межсервисного взаимодействия, обеспечивая высокую производительность и строгую типизацию контрактов. Это позволяет создавать надежные и масштабируемые распределенные системы, способные адаптироваться к изменяющимся требованиям.

4.2. Запросы к базе данных

В проекте запросы к базе данных реализованы через отдельный слой хранилища, который обеспечивает изоляцию работы с данными от бизнес-логики и обработчиков HTTP-запросов. Для устройств, групп и портов созданы свои репозитории, в которых определены методы для выполнения основных операций.

Для получения информации из базы данных используются SQL-запросы SELECT. К примеру, чтобы получить список всех устройств или найти устройство по идентификатору, выполняется соответствующий SELECT запрос с нужными условиями. Для поиска по имени или фильтрации по другим параметрам в запросах используются WHERE, LIMIT и OFFSET, что позволяет реализовать постраничную навигацию и гибкую фильтрацию данных.

Добавление новых записей выполняется с помощью SQL-запросов INSERT. При создании нового устройства, группы или порта в базу данных передаются все необходимые поля, а в ответ возвращается идентификатор созданной записи.

Обновление данных выполнено через запросы UPDATE. К примеру, при изменении информации об устройстве или порте обновляются только те поля, которые были изменены пользователем. Для этого в запросе указывается идентификатор записи и новые значения полей.

Удаление записей выполняется с помощью SQL-запросов DELETE. При удалении устройства, группы или порта из базы данных по их идентификатору соответствующая запись полностью удаляется.

Все запросы к базе данных выполняются с учетом обработки ошибок и проверки целостности данных. Использование отдельного слоя для работы с базой данных позволяет изменять структуру запросов, оптимизировать их или переходить на другую СУБД без необходимости переписывать бизнес-логику.

4.3. Уровни устройства сервиса

Для обеспечения гибкости, поддержки масштабируемости системы сервис был разработан в виде модульной архитектуры (см рисунок) состоящая из нескольких основных уровней.

1) API Gateway: уровень который выступает в роли единичной точки входа для всех внешних запросов. Помимо маршрутизации запросов к соответствующим сервисам.

2) Business Logic: это ядро всей системы, в которой реализована основная предметно-ориентированная логика. На этом уровне происходят преобразования сырых данных из базы в доменные модели, которые соответствуют бизнес-процессам, выполнение сложных операций, таких как анализ зависимостей, расчеты или применение правил валидации, оркестрация процессов. Также вызов внешних сервисов и обработку транзакций. Вся логика организована в виде набора независимых модулей, что позволяет адаптировать функционал без переработки всей системы.

3) Database Requests: слой, абстрагирующий работу с хранилищами данных.

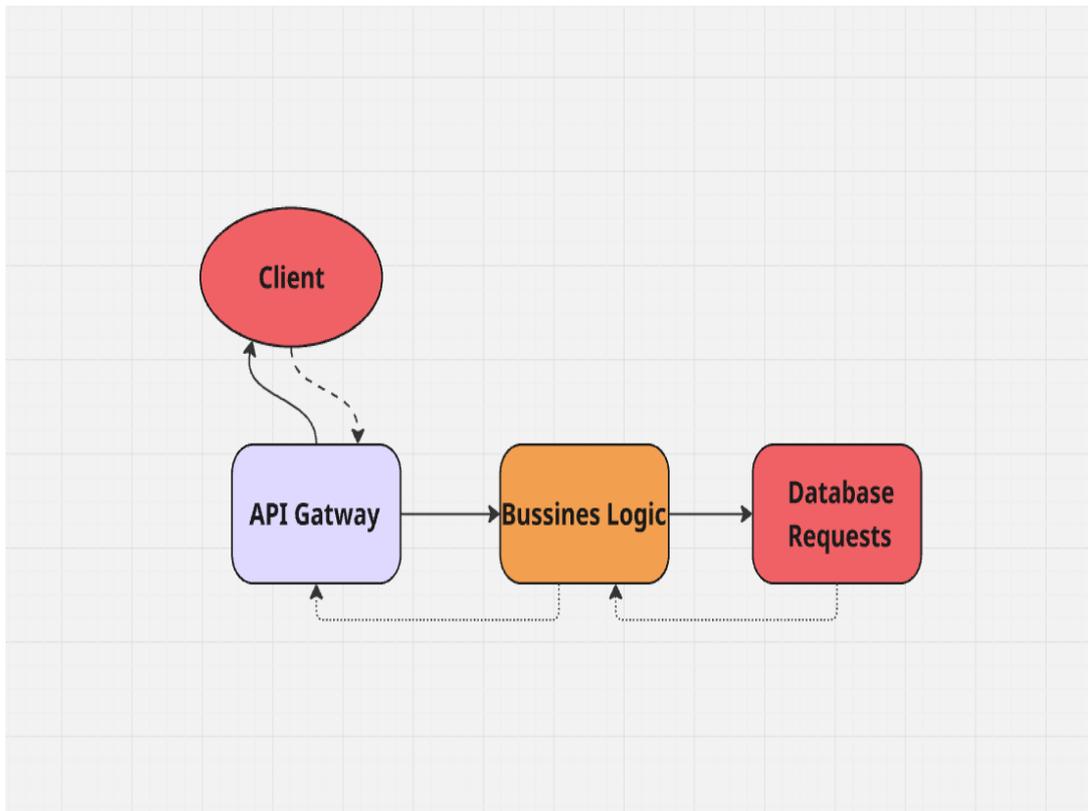


Рисунок 6. Схема уровней сервиса

В эволюции взаимодействий между данными блоками активно применялись паттерны проектирования и абстракции:

Абстракция: каждый уровень взаимодействует с другими через интерфейсы, а не через конкретные реализации. К примеру, бизнес-логика обращается к хранилищу через абстракцию, при этом можно заменять источники данных без модификаций кода приложения.

Абстрактная фабрика – это паттерн, который позволяет создавать объекты без упоминания их конкретных классов, вместо этого используются интерфейсы, которые описывают методы, которыми должна обладать структура.

Фасад – это паттерн, который позволяет создать упрощенное взаимодействие со сложной системой

Благодаря данным подходам удалось достичь такой архитектуры, где все компоненты являются взаимозаменяемыми и масштабируемыми. Такая

архитектура отвечает текущим требованиям и закладывает основу адаптации к будущим изменениям сервиса.

В качестве языка программирования был выбран Golang - современный компилируемый язык, который сочетает в себе высокую производительность С-подобных языков с простотой синтаксиса и развитой экосистемой. GO соответствует требованиям распределенной системы:

1) Эффективность для API Getway - использование стандартной библиотеки net/http и фреймворка Chi прошло мимо легковесного и высокоскоростного шлюза, обрабатывающего тысячи параллельных подключений.

2) Бизнес-логика с приоритетом читаемости - статистическая типизация и легкий синтаксис уменьшают риск ошибок в сложных операциях.

4.4. Структура базы данных

База данных состоит из трех основных таблиц: group, device и port. Таблица Group содержит следующие поля:

1) id - уникальный идентификатор группы (целое число, первичный ключ)

2) name - название группы

Таблица Device содержит:

1) id - уникальный идентификатор устройства (целое число, первичный ключ)

2) name - название устройства

3) hardware_id - аппаратный идентификатор устройства

4) parent_id - идентификатор родительского устройства (целое число, внешний ключ, может быть null)

5) group_id - идентификатор группы, к которой принадлежит устройство

Таблица Port содержит:

- 1) id - уникальный идентификатор порта (целое число, первичный ключ)
- 2) name - название порта
- 3) owner_id - принадлежность порта
- 4) device_id идентификатор устройства к которому относится порт

Связи между таблицами реализованы через внешние ключи:

- 1) device.group_id ссылается на group.id
- 2) device.parent_id ссылается на device.id (иерархия устройств)
- 3) port.device_id ссылается на device.id
- 4) Данная структура позволяет хранить устройства, объединять их в группы, строить иерархии устройств и привязать к ним порты.

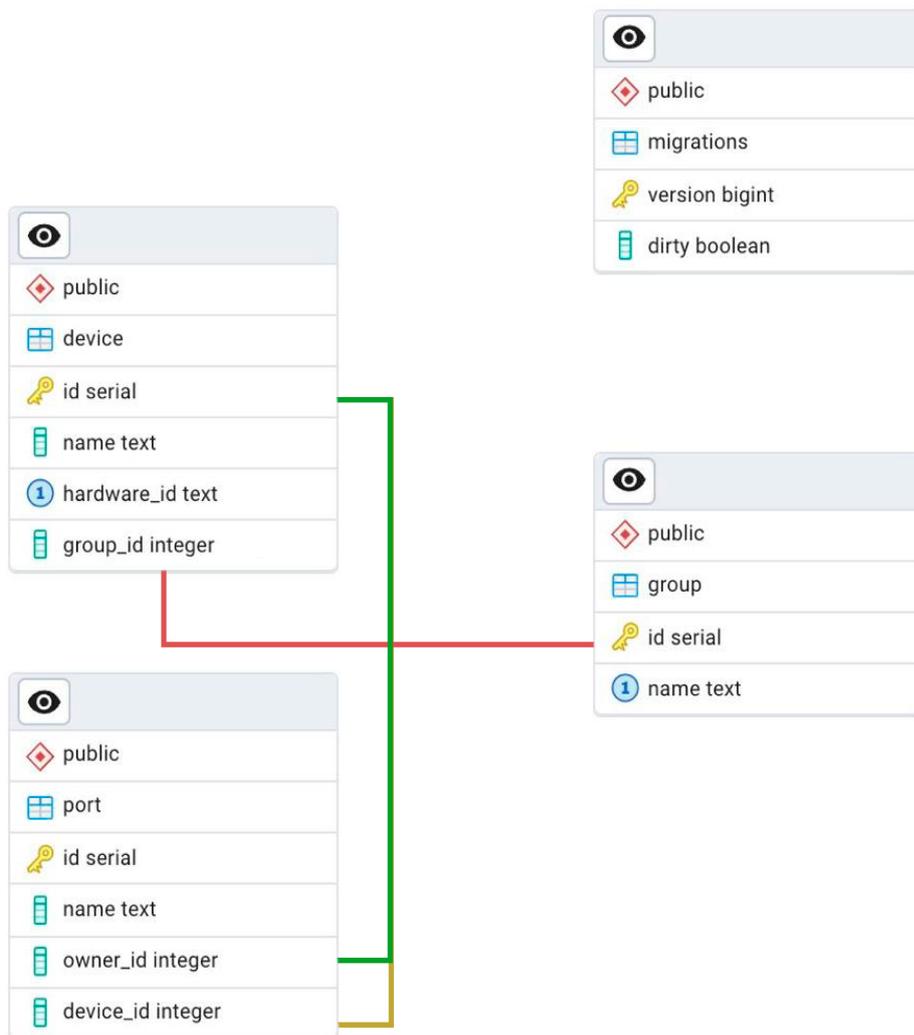


Рисунок 7. Схема структуры базы данных

4.5. Бизнес логика

Бизнес-логика на уровне сервиса распределена по модульной архитектуре, где функционал распределен на отдельные сервисы, взаимодействующие друг с другом посредством хорошо определенных интерфейсов. Есть два интерфейса:

- 1) Group: контролирует операции с группами объектов
- 2) Device: контролирует обрабатываемые данные, связанные с компонентами

Каждый сервис ориентирован на решение задач в своей области, что обеспечивает высокую степень связанности кода и снижает зависимость компонентов друг от друга.

Для обработки данных каждый сервис использует абстракцию уровня доступа к базе данных, что дает возможность спокойно разрабатывать логику запросов без изменения бизнес-процессов. Это разделение гарантирует, что масштабирование и оптимизация механизмов хранения данных не будут включать переписывание кода сервисов, а собственная бизнес-логика останется независимой от специфики реализации хранилища. Сервисная инфраструктура создана вокруг пула соединений с базой данных, эффективно распределяющего ресурсы и минимизирующая задержки при выполнении операций.

Критически важным в архитектуре является использование метода атомарных транзакций. Началом работы каждой сервисной операции становится начало транзакции, в рамках которой выполняются все запросы на базу данных и информационные преобразования. Если на одном из этапов возникает ошибка, будь то некорректные входные данные или нарушение целостности информации, транзакция автоматически откатывается, что сохраняет согласованность данных. Успешное завершение всех процессов приводит к фиксации изменений, обеспечивающий атомарность и надежность процессов. Данная модель повышает отказоустойчивость системы и упрощает

тестирование, так как каждый сервис можно проверить изолированно. Кроме того, разделение ответственности между компонентами позволяет распределять нагрузку между разработчиками, ускоряя внедрение новых функций.

В целях обеспечения правильности функционирования сервисов (Group, Device) каждый метод поддерживается набором автоматизированных тестов, включающей как позитивные сценарии, так и обработку исключительных ситуаций.

Тесты не только проверяют основную функциональность, но и соответствие бизнес-логики уровню взаимодействия с базой данных, включая проверку транзакций и целостность данных при параллельной обработке.

4.6. Обработка HTTP запросов

Реализация обработки HTTP-запросов выполнена при помощи современных подходов к построению серверных приложений на языке GO с упором на надежность и масштабируемость. Вся архитектура взаимодействия с клиентами построена вокруг REST API, где используются стандартные HTTP методы (GET, POST, PUT, DELETE) для работы с устройствами, группами и портами.

Архитектура запроса:

На начальном этапе запрос проходит цепочку middleware, которая выполняет:

- 1) логирование всех входящих запросов
- 2) обработку CORS для обеспечения безопасного взаимодействия с фронтендом
- 3) генерацию уникального идентификатора запроса для отслеживания и отладки
- 4) восстановление после критических ошибок без остановки сервера

После middleware запрос передается маршрутизатору, реализованному с использованием фреймворка chi. Для каждой бизнес-сущности определены отдельные группы маршрутов:

- 1) /device - для устройства
- 2) /group - для групп
- 3) /port - для портов

Это обеспечивает логическую структуру и упрощает понимание API.

Каждый маршрут связан с обработчиком, который:

- 1) Декодирует выходные данные (из JSON)
- 2) Валидирует их на полноту, типы и соответствия бизнес-правилам

В случае ошибок возвращает клиенту подробное сообщение с соответствующим HTTP-статусом.

При успешной валидации данные передаются в сервисный слой, содержащий основную бизнес-логику и взаимодействие с хранилищем. Ответ клиента формируется с помощью вспомогательных функций, которые обеспечивают единообразие структуры и коды статусов (200, 201, 400, 500)

REST - методы и маршруты

GET - получение информации:

- 1) /device/get_by_id/{id} - устройство по ID
- 2) /device/get_all - все устройства
- 3) /group/get_by_name/{name} - группа по имени
- 4) /port/get_by_id{id} - порт по ID.

PUT- обновление данных:

- 1) /device/update - обновление устройства
- 2) /group/update - изменение данных групп
- 3) /port/update - изменение информации порта

POST - создание:

- 1) /device/create - создание устройства с портами
- 2) /group/create - новая группа
- 3) /port/create - массовое создание портов

DELETE:

- 1) /device/delete/{id} - удалить устройство
- 2) /group/delete/{group_id} - удалить группу
- 3) /port/delete/{id} - удалить порт

Каждый метод сопровождается строгой валидацией и обработкой ошибок. Все ответы стандартизированы, что упрощает интеграцию с внешними системами.

Обработчики имеют аннотации для автоматической генерации Swagger-документации. Это обеспечивает актуальность, снижает вероятность ошибок и упрощает интеграцию сторонних клиентов.

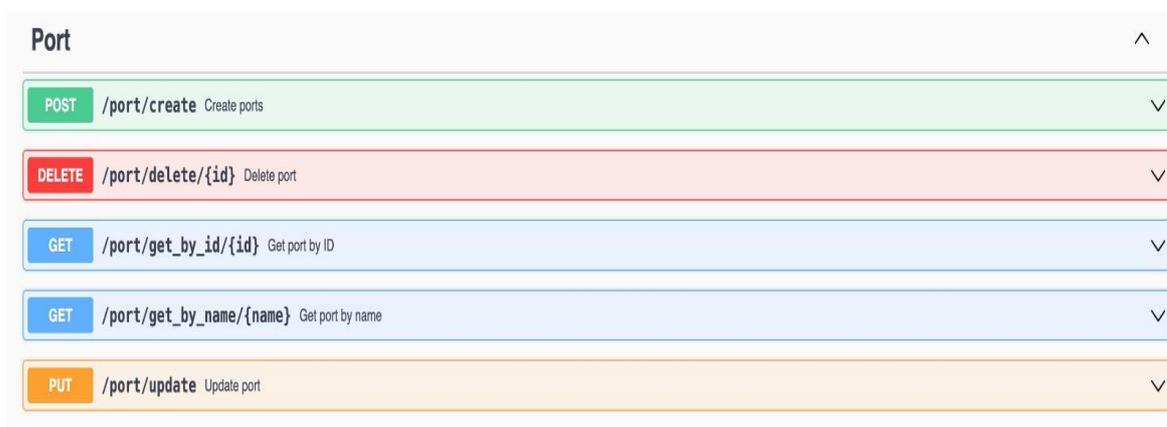


Рисунок 8. HTTP запросы для портов



Рисунок 9. HTTP запросы для групп устройств

Device	
GET	/device/children/{id} Get children devices
POST	/device/create Create device
DELETE	/device/delete/{id} Delete device
GET	/device/get_all Get all devices
GET	/device/get_by_hardware_id/{hardware_id} Get device by hardware ID
GET	/device/get_by_id/{device_id} Get device by ID
GET	/device/get_by_name/{name} Get device by name
GET	/device/parent/{id}/{level} Get parent devices
PUT	/device/update Update device

Рисунок 10. HTTP запросы для портов

4.7. Frontend: Визуализация взаимосвязей между компонентами

В рамках данного проекта реализована визуализация взаимодействия компонентов, написанная с использованием Python и фреймворка Django. При помощи данного веб интерфейса можно удобно взаимодействовать с группами, устройствами и портами, выполнять поиск, фильтрацию.

Общий интерфейс:

The screenshot shows the HWDB web interface. On the left is a sidebar with a tree view containing 'Groups', 'Devices', and 'Ports'. The main content area is titled 'Update Port' and includes a search bar, a 'Search' button, and filter buttons for 'Online' and 'HWID'. Below the search bar is a table of search results. The table has columns for ID, NAME, HWID, PORTS, STATUS, and ACTIONS. One result is shown: ID 1, NAME GigabitEthernet0/1, HWID Router-01, PORTS (with an eye icon), STATUS Up (in a green circle), and ACTIONS (with download, edit, and delete icons). At the bottom, it says 'Showing 1 to 10 of 20 results' and has pagination controls for 1 and 2 pages.

Рисунок 11. Веб интерфейс для визуализации взаимосвязей компонентов и управления устройствами, группами и портами

В левой части экрана находится навигационное меню, позволяющее быстро переключаться между разделами:

- 1) Groups
- 2) Devices
- 3) Ports

Также выбирать типы HTTP запросов (PUT, POST, GET)

В центральной части находится поле для поиска нужного устройства, группы или порта по определенным данным, а также воспользоваться фильтром. (Рис.10)

В таблице на каждом устройстве можно открыть таблицу портов, которая показывает входящие и исходящие порты устройства, а также статус активности портов (Рис. 11)

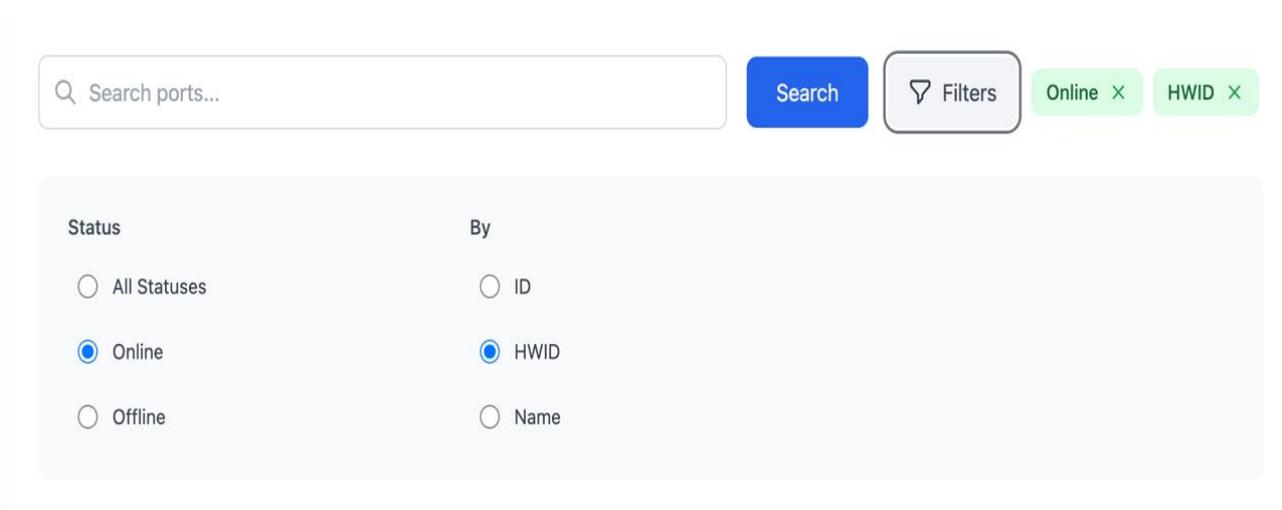


Рисунок 12. Фильтр для поиска групп, портов и устройств

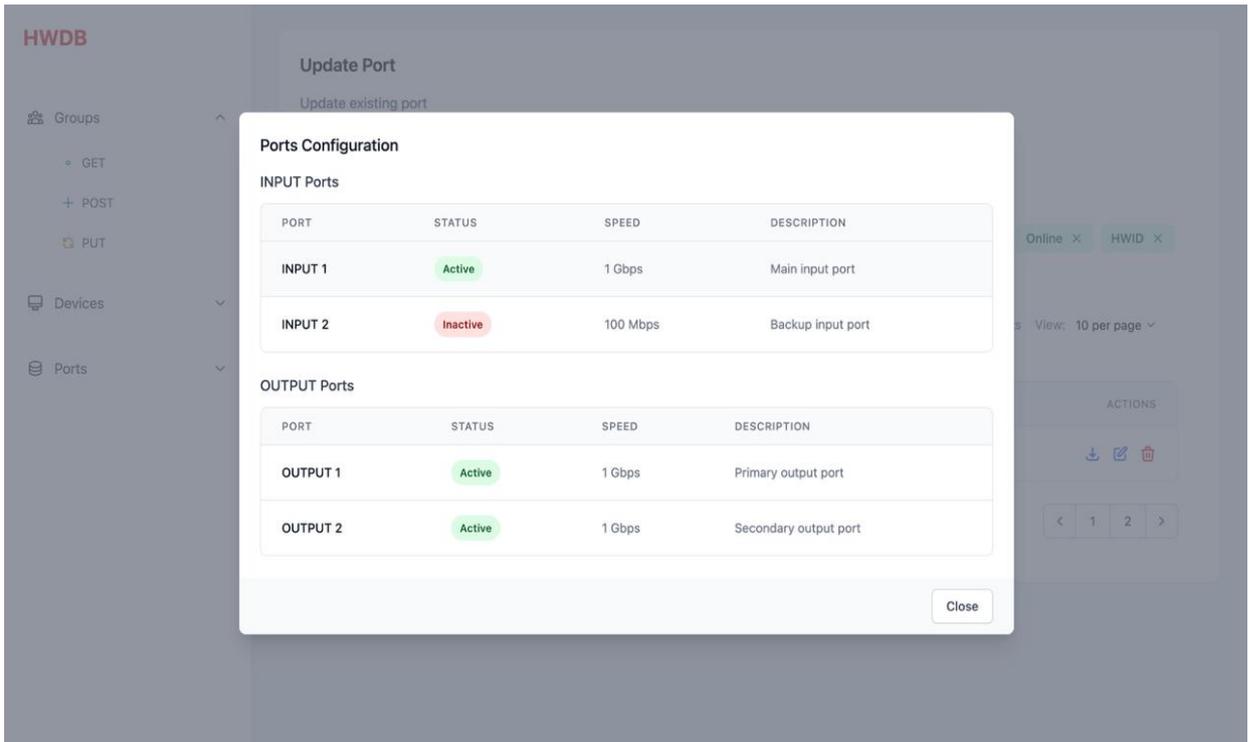


Рисунок 13. Список входящих и выходящих портов устройств

Вывод по главе 4

Работа с базой данных организована через отдельный слой репозитория для изоляции бизнес-логики от SQL запросов. Поддерживаются добавления, обновления и удаления данных с использованием SELECT, INSERT, UPDATE, и DELETE. Структура базы, которая состоит из таблиц group, device, и port, позволяет хранить и связывать информацию об устройствах, группах и портах. Атомарные транзакции и обработка ошибок обеспечивают надежность операций.

Обработка HTTP- запросов реализована через REST API при помощи фреймворка Chi. Middleware выполняет логирование, обработку CORS и восстановление после ошибок, а маршруты для групп и портов логично структурированы. Внедрена Swagger- документация для упрощения интеграции с внешними сервисами.

ЗАКЛЮЧЕНИЕ

В работе была рассмотрена разработка сервиса для хранения и обработки данных о взаимосвязях компонентов установки Spin Physics Detector (SPD) на ускорительном комплексе NICA. Основной проектом работы являлось создание системы, способной обеспечить хранение, быстрый доступ и визуализацию информации о взаимосвязи компонентов установки SPD.

В ходе работы были выполнены следующие задачи:

1) Проведен анализ экспериментальных установок в физике элементарных частиц, изучены особенности детектора SPD.

2) Спроектирована структура базы данных на основе PostgreSQL, включающая таблицы для хранения информации об устройствах, группах и портах.

3) Разработан Backend на языке Golang для обработки HTTP запросов и взаимодействие с базой данных. Сервис содержит в себе CRUD-операции, валидацию данных и транзакционную модель для обеспечения целостности данных.

4) Создан Frontend с использованием React и TypeScript, предоставляющий возможность удобно проводить поиск и фильтрацию данных.

Разработанная система имеет модульную архитектуру, что обеспечивает гибкость, масштабируемость и возможность расширения. Интеграция с JINR SSO гарантирует безопасность и контроль доступа.

СПИСОК ЛИТЕРАТУРЫ

1. Ускорительный комплекс NICA. - Режим доступа: <https://nica.jinr.ru/>
2. NICA Spin Physics Detector (SPD). - Режим доступа: <https://nica.jinr.ru/projects/spd.php>
3. Олейник Д.А. / SPD Online Filter. Первичная обработка экспериментальных данных эксперимента SPD. – Режим доступа: https://indico-hlit.jinr.ru/event/394/contributions/2339/attachments/681/1147/19-10_05_JINR_IT_school_SPD_Online_Filter_2023_.pdf
4. Прокошин, Ф.В. / SPD Information system - Режим доступ: <https://indico.jinr.ru/event/4419/contributions/25174/attachments/18267/31227/SPD%20Information%20systems.big.pdf>
5. СУБД PostgreSQL. - Режим доступа: <https://www.postgresql.org/docs/>
6. SPD Hardware IDs and types. - Режим доступа: https://wiki.jinr.ru/index.php?title=SPD_Hardware_IDS_and_types
7. Шаблон проектирования. - Режим доступа: https://ru.wikipedia.org/wiki/Шаблон_проектирования
8. Язык программирования Golang. - Режим доступа: <https://go.dev/ref/spec#Introduction>
9. Библиотека Chi. - Режим доступа: <https://pkg.go.dev/github.com/go-chi/chi/chi>
10. Open ID - Режим доступа: <https://openid.net/foundation/>
11. Язык программирования Python – Режим доступа <https://docs.python.org/3/>
12. Фреймворк Django – Режим доступа <https://docs.djangoproject.com/en/5.2/>
13. The JINR distributed computing environment – Режим доступа <https://www.researchgate.net/publication/335860362>
14. SPD JINR – Режим доступа <https://spd.jinr.ru/>

15. NICA JINR – Режим доступа <https://nica.jinr.ru/>