

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования
«Тольяттинский государственный университет»

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Разработка программного обеспечения

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка системы автоматизации производства»

Обучающийся

М.С.Ретин

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд. пед. наук, доцент, О.В. Оськина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Аннотация

Тема бакалаврской работы – «Разработка системы автоматизации производства».

Данная бакалаврская работа решает задачу улучшения работы отдела систем передачи данных (СПД) в компании «ИнфоЛада» — местного провайдера интернета, телевидения и телефонии в Тольятти. Существующая проблема в том, что текущий мониторинг сети в компании построен на независимых инструментах: биллинг показывает статус оборудования, а техсайт — схему узлов, но они не связаны. Это тормозит диагностику и заставляет сотрудников тратить время на ручную работу.

Необходимо было создать веб-приложение, которое объединяет эти функции. Оно показывает интерактивную карту сети, где видны все коммутаторы, маршрутизаторы и их состояние, а данные берутся из Prometheus в реальном времени. Основной объект исследования — процессы мониторинга и учета оборудования в СПД, а предмет — как их автоматизировать, чтобы всё стало быстрее и проще.

Актуальность работы. С ростом числа клиентов и оборудования в «ИнфоЛаде» старые методы не справляются. Задержки в ремонте сети вызывает недовольство абонентов, а ошибки из-за несвязанных систем увеличивают нагрузку на инженеров.

Работа состоит из введения, четырех разделов основной части, заключения и списка литературы.

Во введении приводится обоснование актуальности выбранной темы, описываются цели и задачи работы, предмет и объект исследования.

В первом разделе выполняется анализ предметной области с подробным описанием объекта исследования, рассматривается деятельность объекта автоматизации и существующие практики в данной области.

Во втором разделе выполнено концептуальное моделирование проекта, в частности выделены классы и пользователи, составлены функциональные требования, а также описаны бизнес-цели и требования к проекту.

В третьем разделе приведено описание системной архитектуры проекта, составлена информационная модель, описано технологическое обеспечение задачи, а также приведен контрольный пример работы системы.

В четвертом разделе осуществлен расчет экономической эффективности проекта.

В заключении приведены выводы по проделанной работе.

Объем: 83 страницы, 19 таблиц, 30 рисунков.

Оглавление

Аннотация	2
Введение.....	6
Глава 1 Анализ предметной области.....	9
1.1 Описание организации, являющейся объектом исследования.....	9
1.2 Функциональная модель и/или процессная модель организации «как есть».....	11
1.3 Анализ лучших практик в предметной области и обоснование выбора решения по оптимизации/реинжинирингу	15
1.4 Анализ существующих разработок для решения обозначенной задачи .	19
Глава 2 Концептуальное моделирование проекта	25
2.1 Классы и формализация пользователей программного проекта.....	25
2.2 Описание функциональных требований проекта	29
2.3 Формирование бизнес-цели и требований ИТ-проекта для составления календарного плана.....	30
Глава 3 Архитектура проекта и особенности реализации	34
3.1 Системная архитектура проекта	34
3.2 Информационная модель и ее описание.....	38
3.3 Технологическое обеспечение задачи.....	41
3.4 Контрольный пример реализации проекта.....	45
Глава 4 Оценка экономической эффективности проекта	55
4.1 Планирование и организация процесса разработки	55
4.2 Расчет затрат на разработку программного обеспечения	56
4.3 Расчет общих капитальных вложений по вариантам	61

4.5 Расчет годового экономического эффекта и прочих показателей экономической эффективности	73
4.6 Экономический эффект	75
Заключение	78
Список используемой литературы	81

Введение

Тема бакалаврской работы: «Разработка системы автоматизации производства».

Актуальность темы выпускной квалификационной работы обусловлена необходимостью в улучшении взаимодействия сотрудников ИнфоЛады с их сетью, где предоставляют интернет, телевидение и телефонию. Отдел, который следит за сетью — СПД, — тратит много времени на поиск неисправностей. У них есть биллинг, который показывает, что с оборудованием, и техсайт со схемой сети, но они не связаны с друг другом.

Новизна работы заключается в создании простого и дешёвого ПО, которое бы рисовало карту сети «ИнфоЛады» и сразу показывало, где сбой, используя данные из Prometheus. Готовые системы либо для гигантов, либо не подходят под специфику ИнфоЛада: кольцевые магистрали в центре города, звездообразные подключения в новостройках, старые коммутаторы без поддержки новых протоколов. Эта программа создана под конкретные.

Целью выпускной квалификационной работы является разработка системы автоматизации производства.

Объектом исследования является сеть «ИнфоЛады» и то, как отдел СПД ищет и чинит неисправности.

Предмет исследования – программное обеспечение, которое соединяет данные Prometheus с картой сети и делает работу СПД удобнее.

Для создания данного приложения необходимо реализовать следующие задачи:

- разобраться, как отдел СПД сейчас ищет сбои, и что не оптимизировано в биллинге и техсайте;
- проверить, как Prometheus собирает данные с коммутаторов по типу Eltex, Huawei;
- проанализировать что должно быть в программе: карта, фильтры, уведомления, роли пользователей;

- спроектировать программные модули: интерфейс, сервер, база данных;
- написать код на React.js для интерактивной карты сети, Node.js для обработки запросов клиента и сервера и PostgreSQL для хранения данных.
- протестировать ПО на актуальной сети ИнфоЛада и провести опрос по улучшению взаимодействия с приложением;
- составить руководство по установке и настройке комплекса в «ИнфоЛада».

Практическая значимость: приложение сокращает время обнаружения сбоев на 20–25%, позволяя инженерам видеть проблемный коммутатор на карте без перехода между системами. Это снижает простои сети, повышая удовлетворенность клиентов и репутацию «ИнфоЛада». Программное средство можно адаптировать для других провайдеров с похожей инфраструктурой, сохраняя низкую стоимость внедрения. Тесты показали, что операторы тратят меньше времени на обработку заявок, а инженеры быстрее обновляют топологию сети.

Работа состоит из введения, трех глав, заключения и списка используемых источников.

В первой главе проводится анализ как всё устроено в «ИнфоЛада», а именно рассматривается как сейчас устроена работа инженеров, описание базовых понятий с чем работает инженер, оператор и остальные структуры предприятия. Дается общее понимание как устроена топология сети. Описана структура: 200+ коммутаторов, кольцевая топология в центральных узлах, звездообразная на периферии. Выявлены проблемы: биллинг не указывает расположение сбоя, техсайт устаревает из-за ручного обновления. Рассмотрены аналоги («Cisco Prime», «Zabbix», российский «NetPing»), но они либо дороги, либо не подходят для локальных задач.

Во второй главе описано: проектирование программного средства, сформированы требования: карта с фильтрацией по районам, уведомления о

сбоях через SNMP, доступ для админов (настройка), инженеров (диагностика) и операторов (мониторинг). Выбраны инструменты: «React.js» для интерактивной карты, «Node.js» для обработки данных «Prometheus», «PostgreSQL» для хранения топологии и логов [15,16,22].

В третьей главе проводится реализация и тестирование. Разработан комплекс, интегрированный с «Prometheus» через REST API. Карта отображает коммутаторы с цветовой индикацией статуса. Тестирование на 50 коммутаторах показало ускорение диагностики на 20%. Подготовлены инструкции по эксплуатации.

В четвертой главе проводится расчет экономической эффективности программного обеспечения.

Результатом работы является программный комплекс обеспечивающий визуализацию сети «ИнфоЛады» с данными «Prometheus», ускоряет поиск неисправностей и упрощает управление. Решение демонстрирует, как открытые технологии решают задачи малых провайдеров, предлагая альтернативу дорогим системам.

Глава 1 Анализ предметной области

1.1 Описание организации, являющейся объектом исследования

Компания «ИнфоЛада» работает в сфере телекоммуникаций и предоставляет услуги доступа в интернет, телевидение и телефонию для пользователей в Тольятти и Самарской области. Компания занимается проектированием информационных систем для банков и предприятий города, поставками техники и компьютерными программами.

Помимо банков и предприятий «ИнфоЛада» предоставляет доступ в интернет и телевидения для частных лиц.

Компания предлагает новейшие информационные технологии и информационные системы для различных видов деятельности. В отличие от ставшей «традиционной» системы компьютеризации, когда закупают технику, а потом начинают думать, как ее использовать, предлагается сделать вначале системный анализ текущего состояния производства, формализовав существующий документооборот и бизнес — процессы.

Миссия и стратегические цели.

Создавать автоматизированные информационные системы для средних и крупных российских предприятий, организаций, банков и тем самым содействовать укреплению и развитию экономики регионов.

Также стоит цель в обеспечении жителей Тольятти и области высококачественными и доступными услугами связи, интернет и телевидения.

Компания предлагает новейшие информационные технологии и информационные системы для различных видов деятельности, «основанные на системном анализе текущего состояния производства и формализации документооборота и бизнес-процессов» [4].

В компании есть несколько ключевых отделов для работы предприятия, представленные на рисунке 1.

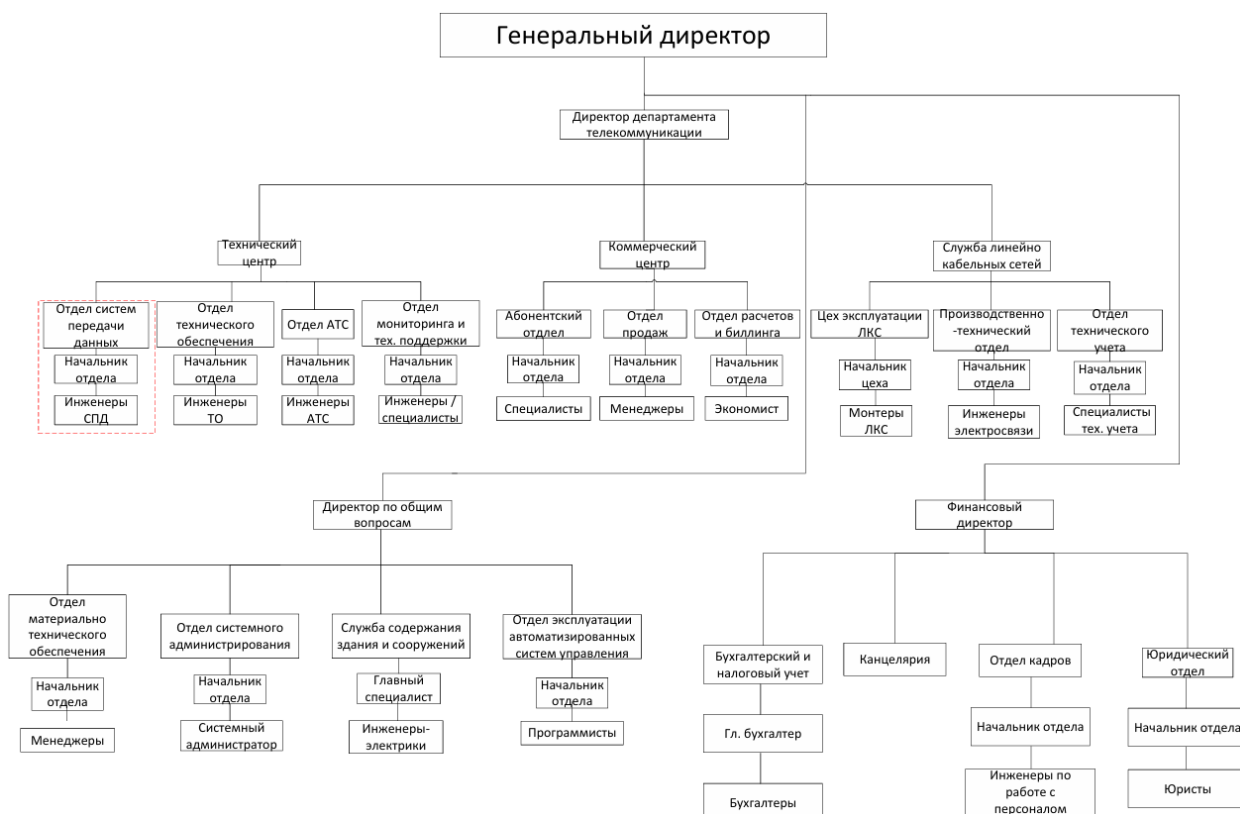


Рисунок 1 – Структура предприятия

Подробнее про ключевые отделы, выполняющие основные функции предприятия.

Отдел технического обеспечения (ОТО) – выполняет функции электромонтажа, установка шкафов с оборудованием, проведение линий питания, установка резервных источников питания.

Отдел службы линейно-кабельных систем (СЛКС) – необходим для прокладки линий связи как до абонентов, так и прокладки и разварки волокон для связи оборудования сети между собой.

«Отдел систем передачи данных (СПД) – связующий отдел, который анализирует текущую ситуацию на сети, передаёт отделам информацию по необходимым работам, где размещать оборудование и какое оборудование между собой соединять» [4].

1.2 Функциональная модель и/или процессная модель организации «как есть»

Функции подразделения отдела СПД:

- внедрение и модернизация сетевого оборудования;
- поддержка и администрирование коммутаторов L2 и L3 уровня модели TSP/IP;
- анализ данных о неисправностях и отказах;
- обеспечение защиты оборудования;
- проведение регламентных и восстановительных работ на сетевом оборудовании;
- обработка тикетов и заявок на технические работы;
- ведение и поддержание технического сайта компании.

Документы, регулирующие работу отдела:

- регламенты по эксплуатации оборудования;
- должностные инструкции;
- заявки на запасные части и комплектующие.

Взаимодействие с другими отделами:

С отделом технического обеспечения (ОТО) для передачи настроенного оборудования на монтаж.

С отделом линейно-кабельных систем для технической поддержки выездных специалистов и помощи в обработке заявок от клиентов.

Основные документы:

Входящие документы: тикеты на обслуживание, заявки на ремонт, списки оборудования.

Исходящие документы: Отчеты о проделанной работе, технические инструкции, заявки на материалы.

“Диаграмма потоков данных” на рисунке 2 показывает общее взаимодействие между отделом систем передачи данных (СПД) и другими отделами [20]:

- абоненты передают тикеты на обслуживание в отдел СПД, а также получают отчеты о выполнении работ;
- отдел технического обеспечения получает настроенное оборудование, а отдел линейно-кабельных систем — заявки на поддержку.

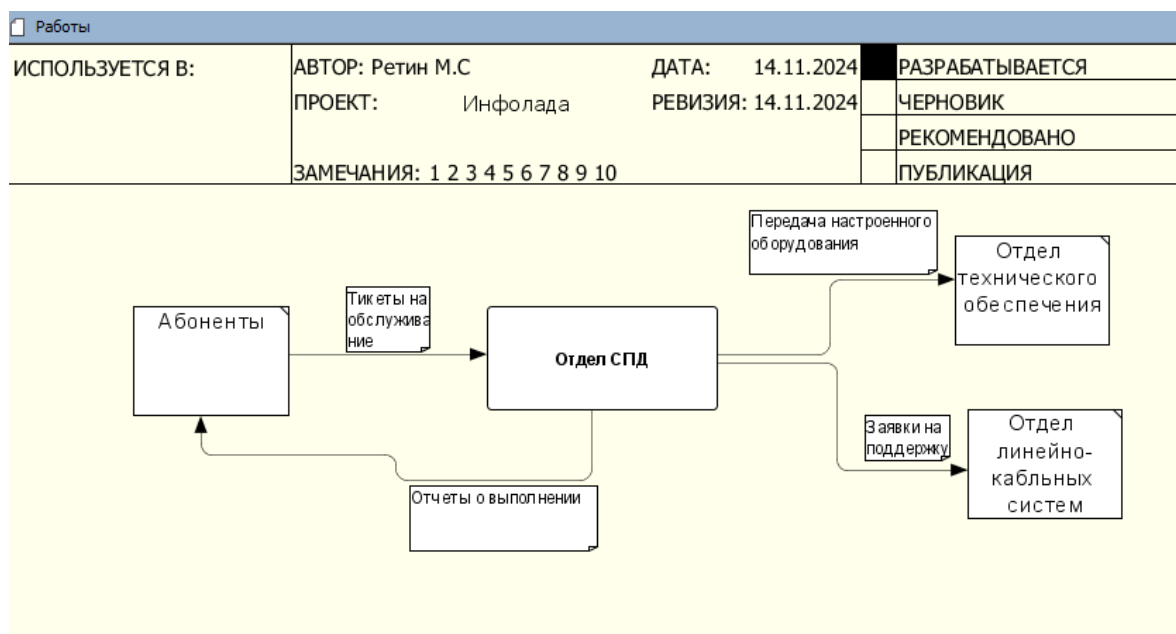


Рисунок 2 – Диаграмма потоков данных.

На данный момент процесс мониторинга и учета оборудования осуществляется с помощью нескольких несвязанных инструментов, что усложняет быструю диагностику и требует дополнительных программ для добавления и отслеживания узлов. Взаимодействие между отделами ограничено, так-как доступ к логической схеме сети есть только у отдела СПД, а использование текущего сайта невозможно из-за ограничений по доступу. Это приводит к задержкам в работе и увеличивает количество ошибок [7]. Это видно на рисунке 3.

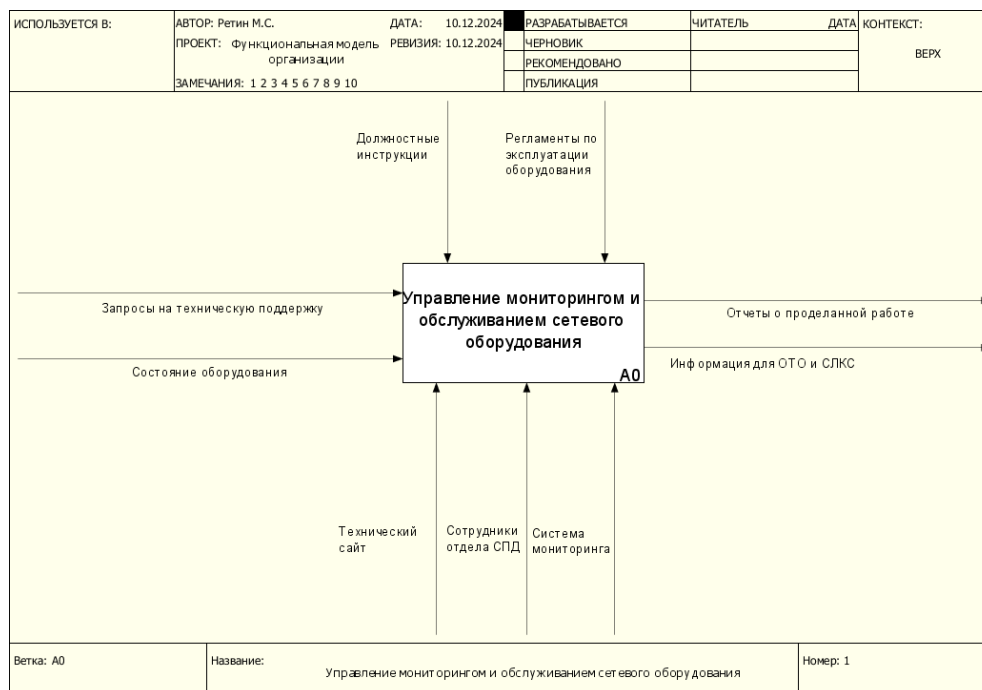


Рисунок 3 – Функциональная модель AS-IS

В текущей модели учета и мониторинга сетевого оборудования используются несколько отдельных инструментов, которые не связаны между собой. Это создает сложности в процессе диагностики и требует использования сторонних программ для добавления и отслеживания устройств.

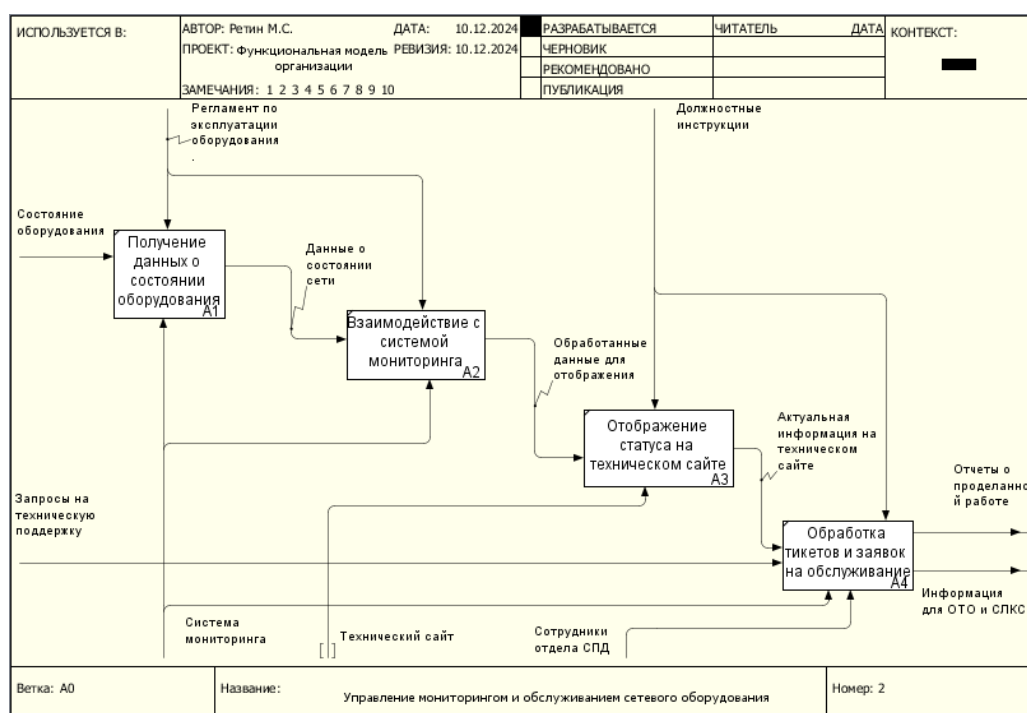


Рисунок 4 – Декомпозиция функциональной модели AS-IS

После внедрения нового решения все узлы сети будут отображаться на интерактивной карте, и сотрудники компании смогут в любое время получать актуальную информацию о состоянии оборудования и вносить изменения без использования сторонних программ. Система обеспечит быстрый поиск и анализ данных о сети в реальном времени, а также позволит прогнозировать и устранять неисправности оперативно.

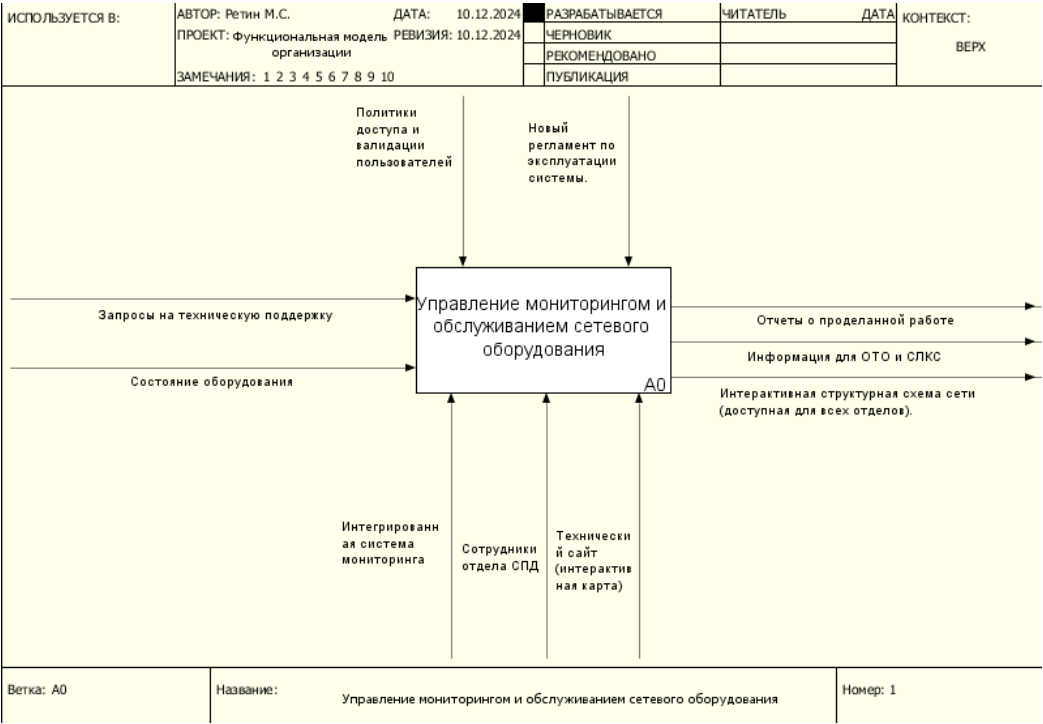


Рисунок 5 – Функциональная модель ТО-ВЕ

Теперь, для создания “IDEF0 модели «как должно быть»”, выделим основные шаги и процессы на рисунке 6, которые нужно будет интегрировать и улучшить, учитывая автоматизацию и улучшения на основе анализа текущей ситуации [20].

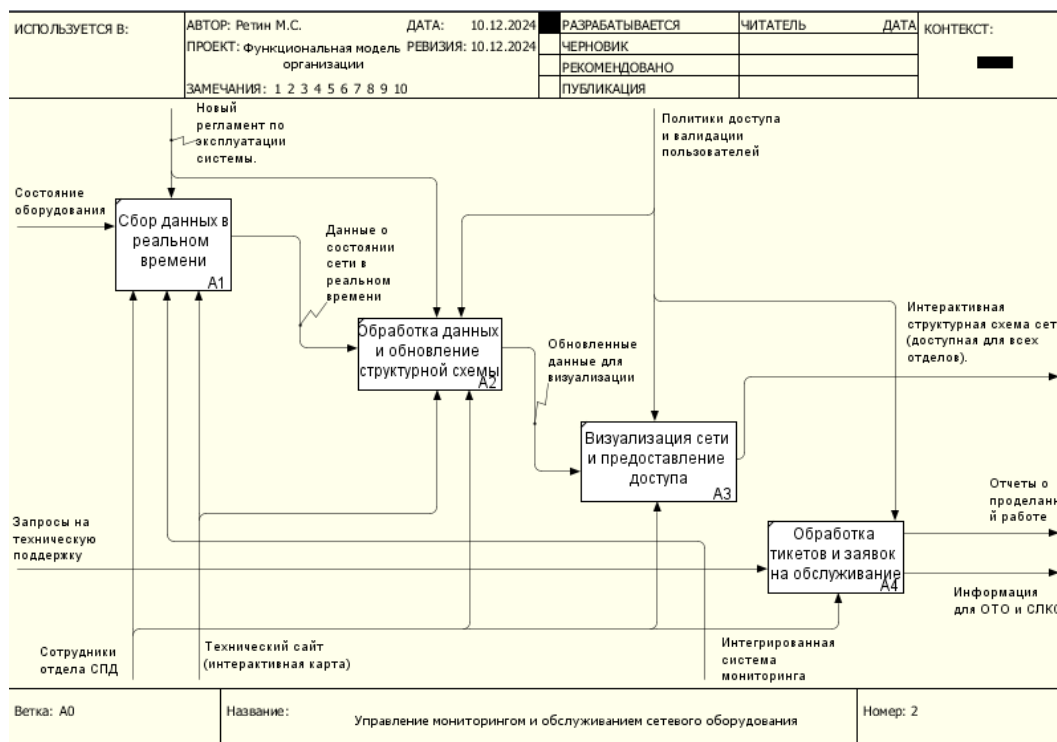


Рисунок 6 – Декомпозиция функциональной модели ТО-ВЕ

1.2 Анализ лучших практик в предметной области и обоснование выбора решения по оптимизации/реинжинирингу

Для оптимизации предприятия в сфере телекоммуникаций рассмотрим несколько подходов, которые полезны для компании «ИнфоЛада».

Автоматизация учета и мониторинга оборудования

В современных компаниях по предоставлению услуг связи используются системы мониторинга, для отслеживания состояния оборудования на сети, просмотр состояний линий абонентов, время авторизации и показ графиков о передаваемом трафике на портах [12].

Системы мониторинга в реальном времени выполняют следующие задачи:

Отслеживает состояние оборудования.

Осуществлять прогнозирование возможных сбоев на основе данных о производительности.

Автоматически отправляет сообщения о сбоях в работе.

Такой подход позволит отделу СПД значительно повысить эффективность работы, ускорив диагностику и минимизируя время на восстановление работы сетевых узлов.

Визуализация сети и мониторинг в реальном времени.

“Для диагностики проблем в сети и быстрого реагирования на неисправности используется визуальное представление в виде логической схемы подключения оборудования” [7]. Это позволяет техническим специалистам оперативно выявлять проблемные узлы, отслеживать их состояние и немедленно принимать меры по ремонту или замене оборудования. Визуализация включает в себя:

Географические карты расположения оборудования.

Статусы отдельных узлов и соединений.

Историческую информацию о состоянии оборудования.

Пути включения узлов, какой порт с чем связан и ширину пропускания трафика линков.

Такая система также может быть интегрирована с инструментами мониторинга для автоматической диагностики неисправностей [17].

На данный момент крупные провайдеры используют готовые программные решения, такие как:

Система мониторинга и управления Cisco Prime Infrastructure.

Программное обеспечение «Cisco Prime Infrastructure» позволяет управлять топологией сети предприятия как проводной сети, так и беспроводной. Она необходима для поддержания локальной сети, для которой используется Cisco Prime LAN Management Solution (LMS) и систему мониторинга сетей Cisco Prime Network Control System (NCS), которая объединена в одну общую систему управления. Главными задачами данного ПО является:

- выполнение текущих обязанностей администратора (модуль управления полным циклом): контроль состояния технических средств, обеспечение сохранности данных, возврат систем в

- рабочее состояние, фиксация ключевых показателей, документирование состава оборудования, регулировка параметров устройств (с применением стандартных форм, заранее прописанных алгоритмов и прочих инструментов), диагностика сетевых ресурсов;
- проверка бесперебойности функционирования программ (модуль Assurance);
 - мгновенная подготовка оборудования к эксплуатации без использования прямого консольного взаимодействия (модуль Plug and Play).

«Система фиксирует сигналы от устройств через SNMP-уведомления и журналы Syslog» [18], а также самостоятельно извлекает данные с аппаратуры с определённой периодичностью. При нарушении установленных пределов значений она отмечает возникновение нештатных ситуаций. Доступны заранее подготовленные модели наблюдения с фиксированными границами, которые можно переработать под собственные нужды.

Полученные сообщения можно просматривать, отклонять или помечать как разрешённые, добавляя к ним поясняющие записи. Из панели управления открывается доступ к инструментам проверки оборудования: тестирование связи, определение пути следования пакетов и выполнение запросов для анализа текущего состояния системы, а также другие сходные действия.

Данные сетевого оборудования хранятся в базе данного программного продукта, к которым есть удобный доступ из пользовательского интерфейса. Также есть возможность управления версиями ПО и доступ к резервным копиям как самого PI так и оборудования сети, которые удобно структурированы в базе [24].

«Cisco DNA» (Digital Network Architecture) - предназначено для тонкой настройки и управления сетевой структурой предприятия или производства. Оно позволяет легко менять конфигурацию сети, изменение политик доступа

и протоколов доступа сети. Преимущество в том, что она на лету строит связи между оборудованием и позволяет сразу начать управлять структурированной корпоративной сетью.

Архитектура Cisco DNA строится на основе проверенных практик и новейших разработок, ранее не находивших применения в корпоративных сетях.

Для обеспечения слаженной работы всех элементов и решений в рамках этой системы применяются программные интерфейсы API с открытым доступом, которые определяют оптимальные способы их интеграции.

Архитектура Cisco DNA выстраивается на сочетании устоявшихся методов и новейших разработок, которые до этого не применялись в корпоративных сетевых решениях.

Взаимодействие всех её частей регулируется через программные интерфейсы API с открытым доступом, задающие предпочтительные пути их совместной работы. ИТ-функции предприятия объединены в крупные категории:

- упрощение и механизация процессов в информационных технологиях;
- ускоренное освоение передовых идей;
- извлечение сведений о сети, программах и пользователях для анализа;
- сокращение уязвимостей, трат и текущих расходов;
- организация продуктивного и автоматизированного сотрудничества с основными бизнес-направлениями.

«Cisco DNA» продолжает развитие концепции SDN (сетей с программным управлением). Изменения затрагивают управление сетевыми ресурсами, наблюдение за их состоянием, аналитические инструменты (включая обработку с применением машинного обучения) и связи между

этими элементами, что позволяет оперативно реагировать на сбои, ошибки и угрозы в сети.

Позволяет в реальном времени получать отчеты от оборудования о возможных поломках или нестабильности работы, благодаря машинному обучению. Через данное средство легко взаимодействовать с оборудованием на прямую, при этом без использования встроенного CLI данного оборудования. Можно менять конфигурацию, планировать обновления и выгружать резервные копии прямо из системы.

Данное программное средство также сигнализирует о проблеме на сети в реальном времени, также предоставляет информацию о включенных приложениях и сервисах опрашивая подключенное оборудование.

Эти программные решения действительно покрывают весь перечень задач, который необходим на сетях провайдера. Но основной минус — это цена. Для использования данного ПО необходимо собирать сеть только на продукции Cisco, которое весьма дорогостоящее, а также покупка и обслуживание программного обеспечения.

В связи с этим и было принято решение разработки своего ПО, под свои нужды.

1.3 Анализ существующих разработок для решения обозначенной задачи

В «ИнфоЛада» уже используются системы учета и мониторинга сетевого оборудования, с автодиагностикой сети и отправке уведомлений о проблемах называемое биллинг. В ней заносятся оборудование в узлы доступа, налаживается связь с коммутаторами по протоколу SNMP и ICMP для отслеживания следующих состояний:

- доступность хоста.
- достояние порта.
- трафик на порту.

- загрузка процессора.
- состояние питания AC/DC.
- температура.

Связь биллинга и оборудования производится по средствам популярного API «Prometheus» [12].

«Prometheus» — это бесплатное программное обеспечение, используемое для мониторинга событий и оповещения. Он записывает показатели в базу данных временных рядов (с учетом высокой размерности), построенную с использованием HTTP pull model, с гибкими запросами и оповещением в режиме реального времени. Проект написан на Go и лицензируется по лицензии Apache 2, с исходным кодом, доступным на GitHub, и является дипломным проектом «Cloud Native Computing Foundation», наряду с Kubernetes и Envoy [12].

«Prometheus» сохраняет информацию в виде показателей, каждому из которых присвоено название для обращения и запросов.

Эти показатели можно уточнять через произвольное число пар "ключ-значение" (меток), отражающих источник данных (например, устройство, их предоставляющее) или детали приложения, такие как код ответа HTTP, тип запроса (GET/POST) или точка назначения. Возможность задавать гибкий набор меток и обрабатывать их в текущем времени определяет многоплановость модели данных «Prometheus»[12].

Сведения собираются в виде временных последовательностей. Процесс основан на активном извлечении: сервер «Prometheus» периодически обращается к перечню источников (именуемых экспортёрами). Каждый источник выдаёт актуальные данные через точку доступа, после чего сервер сводит их воедино. Для автоматического выявления доступных ресурсов, как показано на рисунке 7, используются встроенные механизмы поиска.

The screenshot displays the BILINFOLADA billing system interface. The top navigation bar includes links for 'Интернет каналы', 'BOCC 2022.03.24', 'Tech Site - Главн...', 'Web CUI 1.47.37', 'Система управл...', 'Определение гр...', 'Топологии (Топол)', 'ASBna', 'Docs', 'Главная стран...', 'Index of /', 'Статистика Билл...', and 'адреса Билл...'. The main content area is divided into several sections: 'Мониторинг' (Monitoring), 'Управление маршрутизаторами' (Router Management), 'Сл.Разм...', '01079', 'Замена оборудования', 'Список VLAN', 'IP роут', and 'Новые заявки: 1, 5'. The 'Мониторинг' section shows a list of network devices with columns for 'Выход по IP', 'Выход по сообщению', and 'Выход по статусу'. The 'Управление маршрутизаторами' section shows a list of routers with columns for 'Базовые, Кю, Физ, VIP', 'Платформа', 'Состояние', and 'Скорость'. The 'Сл.Разм...' section shows a list of services with columns for 'Служба', 'Состояние', and 'Скорость'. The '01079' section shows a list of devices with columns for 'Имя', 'IP', 'Состояние', and 'Скорость'. The 'Замена оборудования' section shows a list of devices with columns for 'Имя', 'IP', 'Состояние', and 'Скорость'. The 'Список VLAN' section shows a list of VLANs with columns for 'Имя', 'IP', 'Состояние', and 'Скорость'. The 'IP роут' section shows a list of IP addresses with columns for 'Имя', 'IP', 'Состояние', and 'Скорость'. The 'Новые заявки: 1, 5' section shows a list of new requests with columns for 'Имя', 'IP', 'Состояние', and 'Скорость'.

Рисунок 7 – Биллинг ИнфоЛада

Помимо биллинга используется графическое представление узлов доступа, однако текущая структурная схема сети не обладает всем функционалом, необходимым для эффективного управления и оптимизации работы с оборудованием. Рассмотрим существующее решение, которое внедрено в компанию.

Для визуализации узлов доступа компании используется технический сайт.

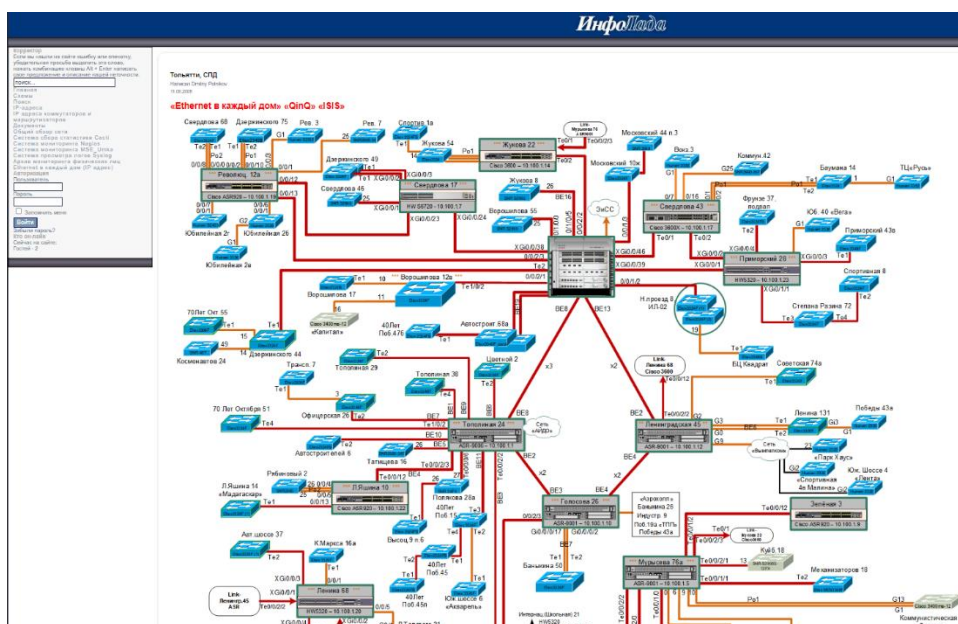


Рисунок 8 – Схема узлов доступа

Номер порта	Имя порта	Направление
100	TenGigE0/0/0/14	
101	TenGigE0/0/0/15	
102	TenGigE0/0/0/16	
103	TenGigE0/0/0/17	
105	TenGigE0/0/0/19	
106	TenGigE0/0/0/20	
107	TenGigE0/0/0/21	

Рисунок 9 – Вид схемы узлов доступа

Данный сайт предоставляет возможности такие как:

- просмотр структуры сети;
- описание портов оборудования;
- сайт отображает пути включения оборудования;
- добавление новых схем узлов связи.

На данный момент сайт реализован недостаточно продуманно, для добавления нового узла его необходимо отрисовать заранее схему в специализированной программе, например «MS Visual Studio» и далее загрузить эту схему на сайт, а для того что бы происходил редирект на отдельный узел и более подробного изучения структуры сети необходимо вручную прописывать координаты конкретного блока, по которому будет происходить дальнейшая навигация, что существенно затрудняет использование и поддержания в актуальном виде технический сайт.

Пример кода страницы для перехода на узел ASR_9006:

```
<area alt="ASR_9006" coords="297,39,450,104" shape="rect" href="/minisite/index.php?ip=10.100.1.10" /> </map>
```

И так необходимо прописывать каждый отдельный узел.

Также сайт не выводит информацию о состоянии оборудования из-за чего и возникает потребность использования двух средств, таких как биллинг и техсайт.

Поэтому было решено реализовать новое приложение.

На основании анализа текущих процессов можно сформулировать следующие функциональные требования для проекта по визуализации сети.

Основные функциональные требования:

- возможность добавлять и обновлять узлы сети, такие как коммутаторы, маршрутизаторы, серверы и другие устройства;
- соединения между узлами, отображающие логическую связь между оборудованием;
- возможность масштабировать карту для подробного просмотра топологии, с возможностью получения подробной информации о каждом узле.

Не функциональные требования

- возможность просмотра истории изменений состояния сети;
- поиск устройств по ключевым параметрам, таким как тип устройства, IP-адрес или местоположение;
- разграничение прав доступа.
- система должна быть масштабируемой, что позволит легко добавлять новые устройства и расширять сеть без необходимости глобальных изменений.

Проект направлен на разработку современного решения для улучшения мониторинга и диагностики телекоммуникационной инфраструктуры компании. Основная цель — повысить качество обслуживания, улучшить взаимодействие сотрудников с сетью и сократить время на устранение неполадок.

В итоге первой главы можно сделать следующие выводы:

- актуальность автоматизации: автоматизация производственных процессов является необходимым условием для повышения эффективности и конкурентоспособности предприятий, современные рыночные условия требуют оптимизации ресурсов и снижения затрат, что можно достичь только через внедрение автоматизированных систем;

- выявление ключевых процессов, анализ предметной области позволил идентифицировать основные процессы, требующие автоматизации, включая управление производственными потоками, учет материалов и комплектующих, планирование производства и контроль качества;
- были выявлены специфические потребности целевой аудитории, включая требования к удобству интерфейса, скорости обработки данных и интеграции с существующими системами, учет этих потребностей критически важен для успешного внедрения системы;
- проведен обзор существующих на рынке решений, что позволило определить их сильные и слабые стороны, выяснилось, что многие из них страдают от недостатка гибкости, сложности в использовании и высокой стоимости, что создает возможности для создания более эффективного продукта.

Глава 2 Концептуальное моделирование проекта

2.1 Классы и формализация пользователей программного проекта

Работа системы во многом строится вокруг того, как распределяются роли среди пользователей — в телекоммуникациях их обычно называют «бизнес-ролями», и они напрямую влияют на то, что каждый может сделать в автоматизированной среде [1,11]. «Эти роли чётко разграничивают доступ к разным частям системы, чтобы никто не путался и не лез туда, где ему не положено, что упрощает ориентироваться в интерфейсе. Скажем, если оператор вдруг получит доступ к настройкам администратора — это сразу риск для безопасности, особенно когда речь идёт о сети вроде «ИнфоЛада», где каждый узел на карте критически важен» [12]. На этом этапе выделены основные роли, прописано, что каждая из них делает, и продуманы дополнительные функции, которые могли бы помочь, учитывая, как в СПД приходилось разбираться с тикетами и данными с «Prometheus».

Для демонстрации будущих классов была составлена диаграмма классов представленная на рисунке 10.

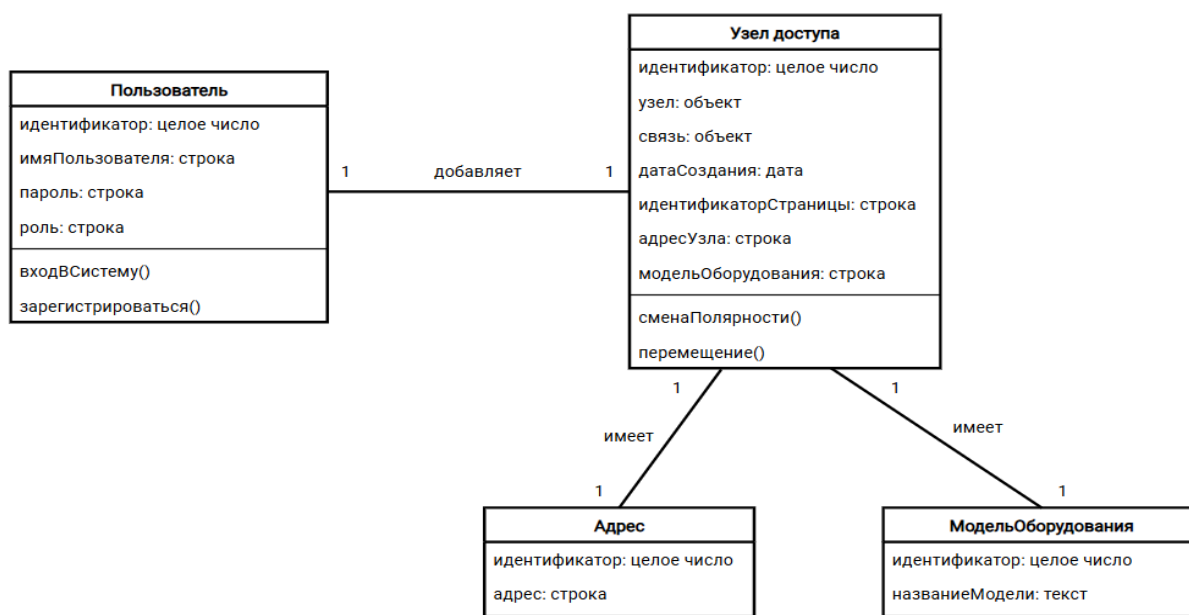


Рисунок 10 – Диаграмма Классов

Определение бизнес-ролей

Администраторы системы:

- функции: Возможность управления системой в админ панели, добавление узлов и ролей с возможностью менять привилегии каждому пользователю. Также добавление и удаление узлов доступа на карте сети. Добавление новых адресов и оборудования в базу данных. Также есть возможность просмотра истории изменения страницы и возврат её состояние до изменений;
- особенности: Требуется высокий уровень технической компетенции для администрирования серверной части «Node.js» и базы данных «PostgreSQL» [15,16];
- дополнительные возможности: Возможность настройки уведомлений о критических сбоях и экспорта логов для аналитики.

Технические специалисты (инженеры СПД):

- функции: Мониторинг состояния сети через интерактивную карту, добавление новых узлов и соединений (например, коммутаторов L2/L3), анализ данных о трафике и неисправностях, обработка тикетов от абонентов;
- особенности: Необходимость оперативного доступа к данным в реальном времени и интеграции с существующим биллингом через API «Prometheus» [12];
- дополнительные возможности: Возможность создания пользовательских фильтров по статусам узлов (например, "вне работы", "перегрев").

Операторы мониторинга:

- функции: Просмотр текущего состояния сети на интерактивной карте, получение уведомлений о сбоях, фиксирование инцидентов в системе без права внесения изменений;

- особенности: Ограниченный доступ к данным, что минимизирует риск случайных ошибок, но требует чёткой визуальной индикации (например, цветовой код состояния узлов);
- дополнительные возможности: Доступ к историческим данным о состоянии сети для подготовки отчётов.

Процесс формализации включает определение данных, необходимых для выполнения задач, последовательность операций и критерии их завершения. Сценарии разработаны на основе анализа потребностей сотрудников «ИнфоЛады» и опыта работы с аналогичными системами.

Сценарий для технического специалиста: Мониторинг и устранение неисправности узла сети

Данные: IP-адрес узла (например, 10.100.1.10 для ASR_9006), статус портов, уровень трафика, температура оборудования (получаемые через Prometheus).

Последовательность:

- авторизация в системе с ролью "Технический специалист";
- переход к интерактивной карте сети через меню "Показать схемы";
- выбор узла ASR_9006 с помощью фильтра по IP или местоположению;
- анализ индикаторов состояния (например, красный цвет сигнализирует о сбое);
- получение уведомления о перегреве через WebSocket и проверка данных в реальном времени;
- обновление записи в базе данных через POST-запрос /api/flow с новыми параметрами узла.

Сценарий для администратора: настройка нового узла и управление доступом.

Данные: Модель оборудования (например, Cisco Catalyst 2950), адрес из справочника access_nodes, параметры доступа (логин/пароль).

Последовательность:

- вход в систему с ролью "Администратор";
- переход в админ-панель для добавления нового узла;
- использование Drag-and-Drop для размещения узла на карте;
- заполнение полей в таблице equipment_model (например, "Cisco Catalyst 3200, 24 порта");
- настройка прав доступа для инженеров через таблицу users;
- сохранение изменений через POST-запрос с ограничением до 5 исторических записей на page_id.

Сценарий для оператора: мониторинг и регистрация инцидента.

Данные: Список активных узлов, история изменений (доступная через /api/flow/history/page/:page_id).

Последовательность:

- авторизация с ролью "Оператор";
- открытие интерактивной карты через главное меню;
- выбор узла с критическим статусом (например, сбой питания);
- регистрация инцидента в системе с указанием времени и описания;
- просмотр уведомления и ожидание реакции инженера.

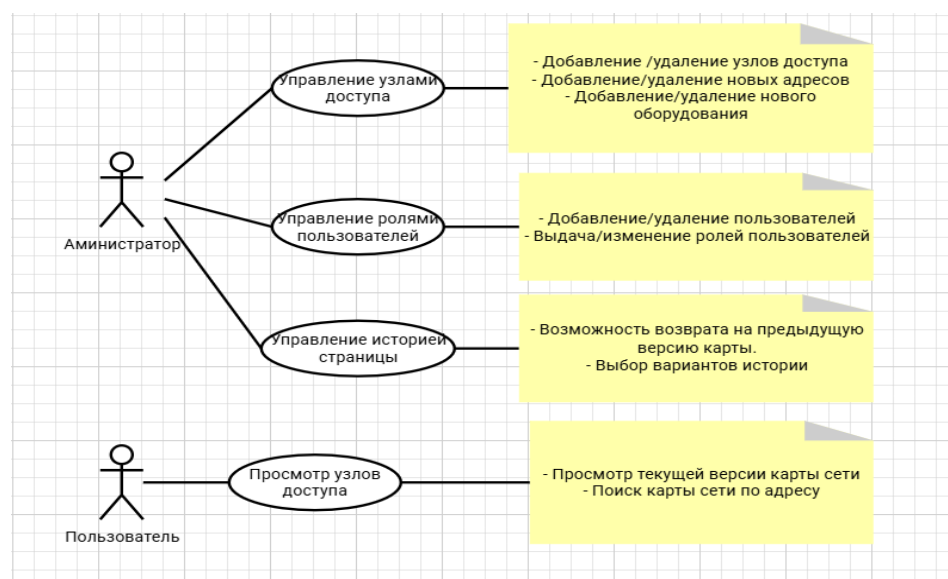


Рисунок 11 - Диаграмма вариантов использования

2.2 Описание функциональных требований проекта

Основные функциональные требования:

Интерактивная карта сети должна включать:

- узлы сети, такие как коммутаторы, маршрутизаторы, серверы и другие устройства, с наглядными индикаторами их состояния;
- соединения между узлами, отображающие текущее состояние (например, нагрузка, ошибки, пропускная способность);
- возможность масштабировать карту для подробного просмотра топологии, с возможностью получения подробной информации о каждом узле (состояние портов, пути подключения, текущий трафик, температура и другие параметры).

Интеграция с существующими системами мониторинга, такими как «Prometheus», для получения актуальных данных о состоянии оборудования.

Обновление данных в реальном времени с минимальной задержкой, используя технологии вроде «WebSocket» [21].

Возможность просмотра истории изменений состояния сети, включая информацию о неисправностях и времени устранения проблем.

Логирование изменений топологии сети, чтобы отслеживать изменения и улучшения инфраструктуры.

Уведомления о критических неисправностях, которые должны отображаться на карте, выделяя проблемные узлы.

Поиск устройств по ключевым параметрам, таким как тип устройства, IP-адрес или местоположение.

Функция фильтрации данных по состоянию устройств, чтобы можно было увидеть только те узлы, которые требуют внимания.

Роли пользователей с различными правами доступа:

Администраторы — полный доступ ко всем функциям.

Технические специалисты — возможность мониторинга состояния сети и добавления новых устройств.

Операторы — доступ только для просмотра и мониторинга.

«Защищенные зоны доступа для предотвращения несанкционированного доступа к важной информации» [26].

Система должна быть масштабируемой, что позволит легко добавлять новые устройства и расширять сеть без необходимости глобальных изменений.

На рисунке 12 представлена последовательность действий роли администратор для добавления нового узла доступа.

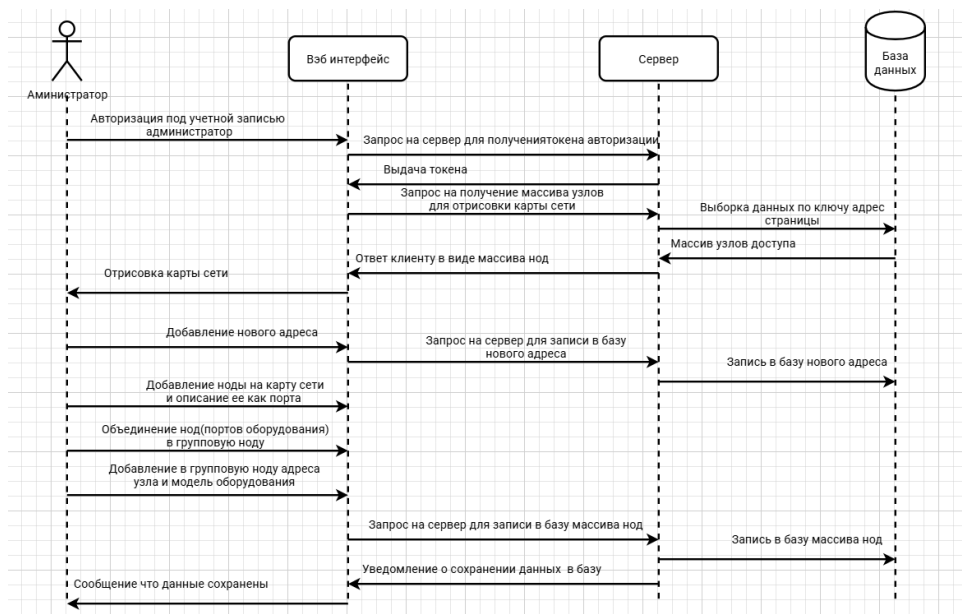


Рисунок 12 - Диаграмма последовательности сценария «Добавление нового узла доступа»

2.3 Формирование бизнес-цели и требований ИТ-проекта для составления календарного плана

Проект по созданию сайта для визуализации сети решает несколько важных задач, которые помогут улучшить мониторинг и управление телекоммуникационной инфраструктурой компании «ИнфоЛада». Важно понимать, как проект будет способствовать достижению бизнес-целей компании.

Повышение эффективности — упрощение мониторинга и диагностики позволит значительно ускорить процессы обслуживания. Система, которая автоматически показывает состояние оборудования и сети, уменьшит необходимость в ручной проверке и ускорит решение проблем.

Снижение нагрузки на сотрудников — автоматизация мониторинга и уведомлений поможет техническим специалистам сосредоточиться на решении реальных проблем, а не на отслеживании состояния оборудования.

Снижение времени простоя — уведомления и исторический анализ помогут быстрее реагировать на неисправности, что снизит время простоя сети и уменьшит финансовые потери.

Оптимизация использования ресурсов — наглядная визуализация состояния сети поможет обнаружить узкие места и избыточно перегруженные узлы, что позволит эффективнее использовать ресурсы без постоянного мониторинга.

Улучшение качества обслуживания клиентов — более быстрое реагирование на проблемы сети благодаря мониторингу в реальном времени снизит количество негативных отзывов и повысит удовлетворенность клиентов.

Повышение надежности сети — возможность быстро диагностировать проблемы и предотвращать сбои улучшит стабильность работы сети.

Увеличение конкурентоспособности — внедрение передового решения для мониторинга и визуализации поможет компании занять лидирующие позиции в области стабильности и качества обслуживания.

Помощь в принятии бизнес-решений — сбор исторических данных о состоянии сети позволит лучше прогнозировать нужды компании и принимать более обоснованные решения.

Поддержка роста компании — система должна поддерживать увеличение масштаба сети и адаптироваться под новые технологии и устройства.

Снижение рисков и повышение безопасности — внедрение системы мониторинга поможет быстро выявлять угрозы, а разграничение прав доступа обеспечит защиту данных.

Обучение пользователей — интуитивно понятный интерфейс и простота освоения помогут быстро обучить сотрудников и улучшить работу всей команды.

Календарный план проекта

Проект будет разбит на несколько этапов. Ожидаемое время на выполнение каждого из них представлено в таблице 1.

Таблица 1 - Календарный план проекта

Этап	Описание	Сроки
Этап 1	Анализ текущих процессов и требований	1 месяц
Этап 2	Проектирование системы и выбор решений	1.5 месяца
Этап 3	Разработка и тестирование системы	2 месяца
Этап 4	Внедрение и обучение персонала	1 месяц

Во второй главе работы, посвященной концептуальному моделированию проекта, были сделаны следующие ключевые выводы.

1. Определение архитектуры системы: В результате концептуального моделирования была разработана модульная архитектура системы, что позволяет обеспечить гибкость и масштабируемость. Основные компоненты системы были четко структурированы, что упрощает как дальнейшую разработку, так и внедрение.

2. Модели бизнес-процессов: Были созданы модели ключевых бизнес-процессов, которые подлежат автоматизации. Это позволило

визуализировать текущие и будущие состояния процессов, а также выявить узкие места, требующие оптимизации.

3. Функциональные требования: На основе концептуального моделирования были сформулированы функциональные требования к системе. Эти требования определяют ожидания от системы и ее основные функции, такие как управление производственными потоками, учет материалов, мониторинг производительности и контроль качества.

4. Интерфейсы и взаимодействие: Важное внимание было уделено проектированию пользовательских интерфейсов и взаимодействию компонентов системы. Это критически важно для обеспечения удобства работы пользователей и повышения эффективности их взаимодействия с системой.

5. Технологические выборы: Анализ потребностей и возможностей системы привел к выбору конкретных технологий для разработки. Это включает языки программирования, базы данных и инструменты для интеграции, что позволит создать надежное и современное решение.

Глава 3 Архитектура проекта и особенности реализации

3.1 Системная архитектура проекта

Система мониторинга и управления сетью для компании «ИнфоЛада» была разработана как удобная и надёжная платформа. Её основная задача — соответствовать текущим требованиям компании и быть готовой к возможному расширению в будущем. Для достижения этих целей выбрана клиент-серверная архитектуру, которая хорошо подходит для подобных проектов благодаря своей гибкости и стабильной работе.

Клиент-серверная архитектура позволяет разделить функционал системы на две основные части — клиентскую и серверную, что упрощает управление и поддержку всех компонентов. Серверная часть отвечает за обработку запросов и управление данными, в то время как клиентская часть предоставляет интерфейс для пользователей, который позволяет им взаимодействовать с системой. Такой подход делает систему масштабируемой: ее можно расширять и модифицировать без значительных затрат времени и ресурсов.

Кроме того, проектировка системы ориентировалась на использование современных технологий, что дает возможность интеграции с различными сторонними сервисами, такими как облачные решения и другие инструменты для аналитики и отчетности. Это значительно повысит функциональность системы, позволяя компании «ИнфоЛада» оптимизировать управление сетевыми ресурсами и повысить уровень сервиса для своих клиентов.

В процессе разработки были учтены требования к безопасности и защите данных, что критически важно в свете растущих киберугроз. Реализованы специальные механизмы аутентификации и шифрования данных, которые гарантируют защиту информации на всех уровнях системы. Ограничение доступа к определенным функциям и данным основано на

ролях пользователей, что дополнительно защищает корпоративную информацию.

Система также включает в себя возможности автоматизированного мониторинга состояния сети. Это позволяет в режиме реального времени отслеживать производительность сетевых устройств, выявлять потенциальные проблемы и уведомлять администраторов о любых неполадках. Благодаря этому подходу, «ИнфоЛада» сможет минимизировать время простоя сетевых сервисов и повышать общую эффективность бизнес-процессов.

В дополнение к этому, интерфейс платформы был разработан с учетом принципов удобства и простоты использования. Это позволяет пользователям быстро обучаться работе с системой без необходимости длительного обучения или специальных навыков. Отзывы пользователей о прототипе в значительной степени повлияли на окончательное оформление интерфейса, что сделало его более интуитивно понятным.

Таким образом, созданная система мониторинга и управления сетью для компании «ИнфоЛада» станет важным инструментом для оптимизации бизнес-процессов. Она не только отвечает текущим требованиям компании, но и создаёт гибкую основу для дальнейшего развития и улучшения услуг, которые предоставляет компания своим клиентам. В будущем планируется внедрение дополнительных модулей, которые расширят функционал системы и позволят эффективнее управлять ресурсами.

Система разделена на несколько компонентов, каждый из которых выполняет свои задачи. Клиентская часть — это веб-приложение, которое работает в браузере. Оно обеспечивает доступ к данным из любой точки, что важно для распределённой инфраструктуры. На карте сети отображаются узлы, соединения и их текущее состояние. Если оборудование выходит из строя, система сразу выделяет это цветом или уведомлением.

Клиентская часть строится на основе «React.js — библиотеки с открытым кодом на JavaScript для создания интерфейсов взаимодействия с пользователем» [22].

«React подходит для разработки приложений с одной страницей или мобильных решений, обеспечивая быстроту создания, удобство и гибкость роста» [20]. В задачах построения интерфейсов он часто комбинируется с инструментами вроде MobX, Redux или GraphQL. Разработка сводится к описанию желаемого отображения, а не к детальным указаниям для браузера, что заметно уменьшает объём типового кода.

«На серверной части работает Node.js, который обеспечивает обработку запросов от клиента и взаимодействует с базой данных и внешними системами» [16]. Сервер — это связующее звено между всеми компонентами системы. Он не только получает данные от мониторинговой системы Prometheus, и анализирует их. Такой подход позволяет минимизировать нагрузку на клиентские устройства, перенеся всю сложную обработку на серверную часть.

«Node.js — это кроссплатформенная платформа с открытым кодом для выполнения JavaScript на серверной стороне» [16].

С момента появления в 2009 году она завоевала значительную популярность и сегодня играет ключевую роль в веб-разработке. Основу составляет движок V8 от Google, знакомый по Chrome. Хотя платформа чаще используется для веб-серверов, её возможности этим не ограничиваются. Основные черты:

Скорость выполнения: «JavaScript» в «Node.js» может работать вдвое быстрее, чем в языках вроде C или Java, и значительно опережать Python или Ruby благодаря неблокирующей структуре (точные результаты зависят от тестов);

Лёгкость освоения: платформа проще в изучении и применении по сравнению с рядом серверных решений;

Асинхронный подход: однопоточная модель с функциями обратного вызова и событийной логикой облегчает написание неблокирующего кода. Тяжёлые операции передаются в обработку с указанием действия по их завершении, исключая ожидание.

Для работы с большими объемами информации, включая историю событий и состояния оборудования, выбрана «PostgreSQL».

«PostgreSQL — это производительная система управления базами данных объектно-реляционного типа с открытым кодом, за более чем 35 лет доказавшая свою надёжность и функциональность» [15].

Её преимущества:

- мощные инструменты транзакций и дублирования данных;
- гибкость языков программирования: базовый набор включает PL/pgSQL, PL/Perl, PL/Python, PL/Tcl, с возможностью добавления PL/Java, PL/R и других, а также поддержка модулей на C;
- механизм наследования;
- работа с геометрическими объектами (включая географию) через PostGIS;
- обработка JSON-данных с их индексацией;
- возможность расширения через создание новых типов данных, индексов и подключение внешних источников.

Для проекта выбраны технологии, которые подходят под задачи системы и её развитие. Интерфейс создан на React.js — он удобный и легко дорабатывается. Node.js отвечает за обработку данных, а PostgreSQL — за их хранение и анализ. Prometheus интегрировали для получения актуальных данных.

3.2 Информационная модель и ее описание

Основная таблица системы — `node_data`. В ней хранятся данные о сети: узлы и их соединения.

- `nodes` — список узлов, где указаны их ID, тип оборудования, IP-адреса и другие параметры.
- `edges` — связи между узлами с информацией о пропускной способности и состоянии соединения.

В таблице также есть поле `created_at` (дата добавления данных) и `page_id`, которое помогает разделять карты сети.

Дополнительные таблицы:

- `access_nodes` — справочник адресов. Пользователи выбирают адрес из списка, чтобы избежать ошибок;
- `equipment_model` — список моделей оборудования для настройки и замены устройств.

Для управления доступом используется таблица `users`. Она содержит логины, пароли и роли:

- администратор управляет системой.
- инженер добавляет и меняет данные.
- оператор только просматривает информацию.

Эта структура помогает организовать работу с данными и защитить систему.

Все таблицы системы связаны и работают вместе:

- Основная информация об узлах и их связях хранится в `node_data`.
- При добавлении узлов и связей используются справочники:
 - `access_nodes` — для выбора адреса.
 - `equipment_model` — для указания модели оборудования.
- Действия пользователей фиксируются через таблицу `users`, которая также регулирует доступ.

Концептуальная модель включает основные сущности:

- Users — пользователи системы.
- Node Data — данные о топологии сети.
- Access Nodes — справочник адресов.
- Equipment Model — справочник оборудования.

Связи между ними:

- Users добавляют и редактируют данные в node_data.
- Node Data использует справочники access_nodes и equipment_model для работы с узлами и связями, представлены на рисунке 13.

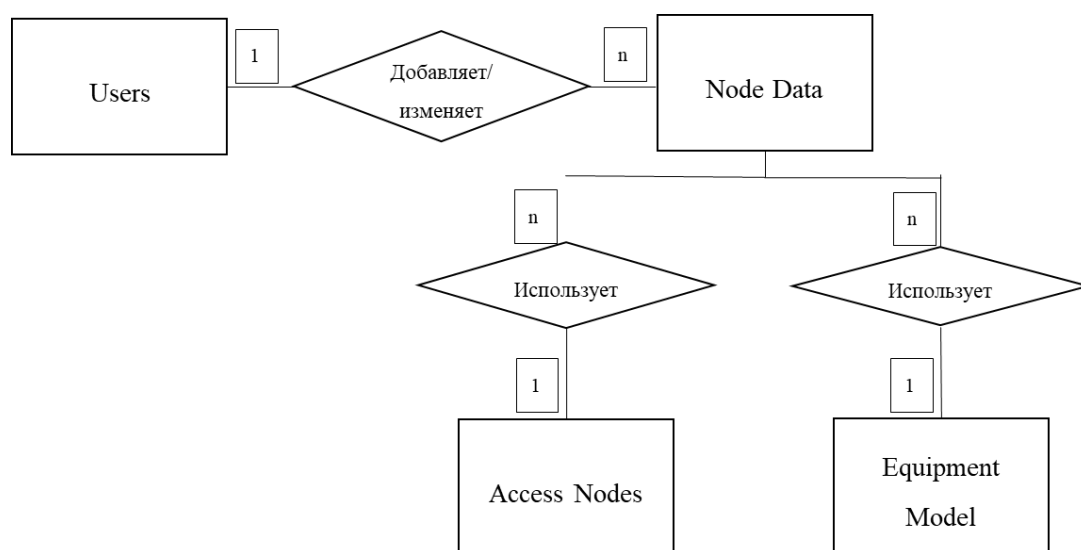


Рисунок 13 – Схема концептуальной модели предметной области

Users взаимодействуют с таблицей Node Data, выполняя добавление узлов и связей.

Node Data использует справочники (Access Nodes для адресов и Equipment Model для моделей оборудования).

Подсхемы приложения представляют собой разделение данных и функций, реализуемых в разных модулях системы. Для проекта ключевые подсхемы могут быть сформированы на основе следующих функциональных модулей: подсхема управление узлами сети. Представлена на рисунке 14.

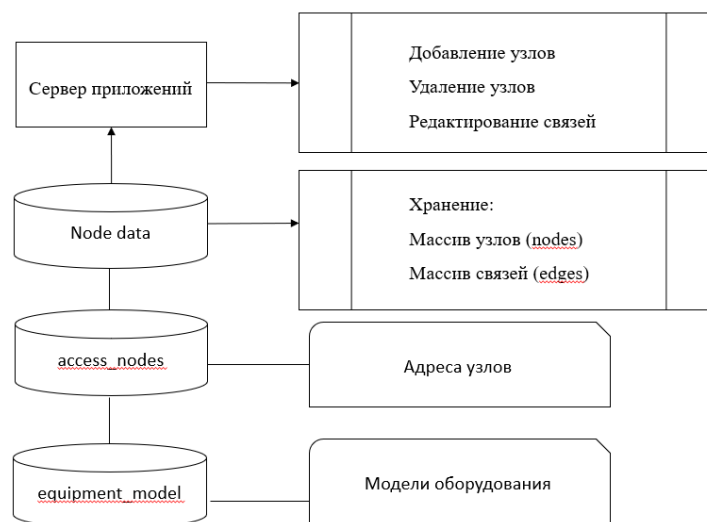


Рисунок 14 – Управление узлами сети

Подсхема на рисунке 14 охватывает функционал работы с узлами и связями. Она использует данные из node_data, а также справочники access_nodes и equipment_model. Подсхема на рисунке 15 отвечает за авторизацию, разграничение прав доступа и управление пользователями.

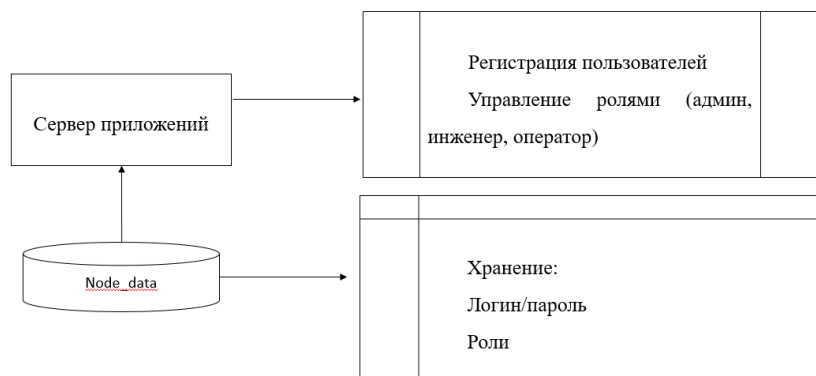


Рисунок 15 – Управление пользователями и доступами

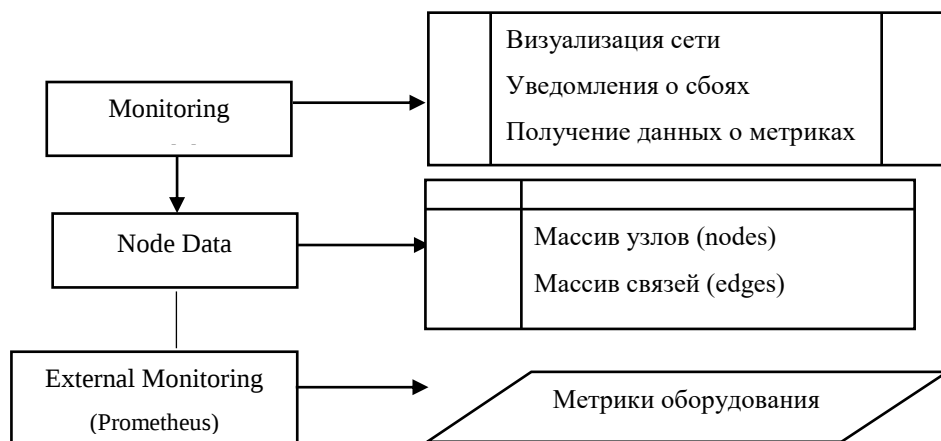


Рисунок 16 – Мониторинг и визуализация сети

Подсхема на рисунке 16 позволяет отображать текущую топологию сети на интерактивной карте, получать актуальные данные и уведомления о сбоях.

3.3 Технологическое обеспечение задачи

Для работы системы создаются модули, которые отвечают за добавление, удаление и обновление данных. Эти модули также позволяют работать со справочниками и событиями.

«Сервер занимается обработкой данных, клиентская часть — их отображением, а база данных — хранением» [3].

База данных представляет собой 4 таблицы где хранятся данные о пользователях, массивах нод, названия модели оборудования, и адреса узлов доступа [27].

Диаграмма, на которой представлено как компоненты взаимодействуют между собой представлена на рисунке 17.

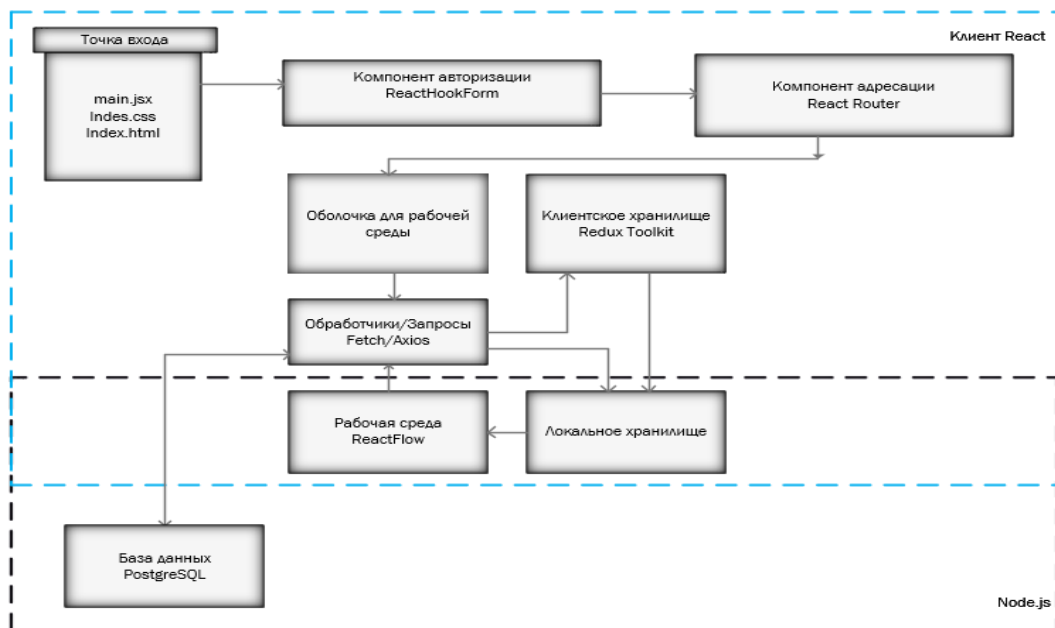


Рисунок 17 – Диаграмма компонентов приложения

Физическая модель базы данных представлена на рисунке 18.

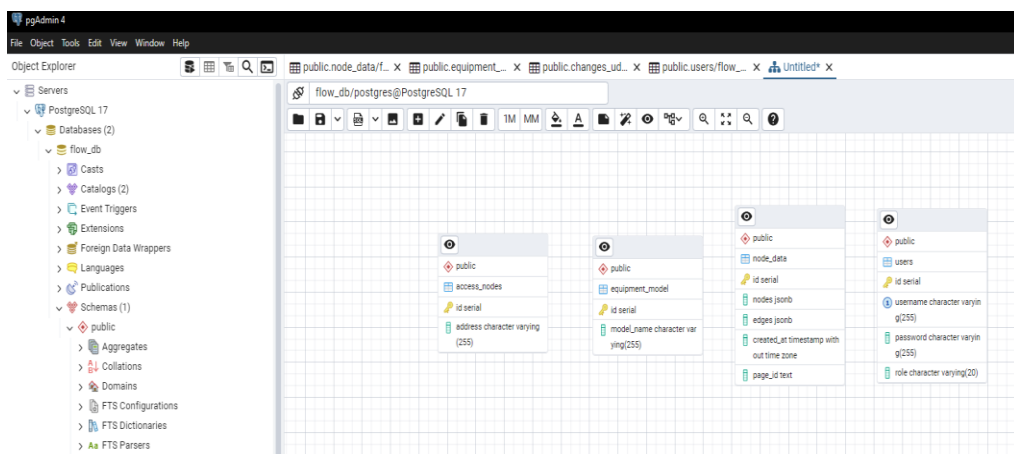


Рисунок 18 – ERD диаграмма базы данных.

Основным функционалом на сервере является добавление и получение массивов нод из базы данных., которыми и являются узлы на нашей интерактивной карте.

1. Получение всех данных нод и линий

```

app.get('/api/flow', async (req, res) => {
  try {
    const result = await pool.query('SELECT * FROM node_data ORDER BY created_at DESC');
    // Преобразуем JSONB-поля в строки JSON для клиента
    const serializedData = result.rows.map((row) => ({

```

```

        ...row,
        nodes: JSON.stringify(row.nodes),
        edges: JSON.stringify(row.edges),
    }));

    res.json(serializedData);
  } catch (error) {
    console.error('Ошибка при получении всех данных:', error);
    res.status(500).json({ error: 'Ошибка сервера при получении данных' });
  }
});

```

В этом блоке кода происходит запрос от клиента на получение массива нод из базы данных и передачу на клиент.

2. Сохранение данных нод и линий

```

app.post('/api/flow', authenticateToken, async (req, res) => {
  const { page_id, nodes, edges } = req.body; // Теперь извлекаем page_id

  if (!page_id || !Array.isArray(nodes) || !Array.isArray(edges)) {
    return res.status(400).json({ error: 'Некорректный формат данных или отсутствует page_id' });
  }

  try {
    // Проверяем количество записей для данного page_id
    const countResult = await pool.query(
      'SELECT COUNT(*) FROM node_data WHERE page_id = $1',
      [page_id]
    );
    const count = parseInt(countResult.rows[0].count, 10);

    // Если записей больше 5, удаляем самую старую
    if (count >= 5) {
      await pool.query(
        'DELETE FROM node_data WHERE page_id = $1 AND created_at = (SELECT MIN(created_at) FROM node_data WHERE page_id = $1)',
        [page_id]
      );
    }

    // Вставляем новые данные
    const result = await pool.query(
      'INSERT INTO node_data (page_id, nodes, edges, created_at) VALUES ($1, $2::jsonb, $3::jsonb, CURRENT_TIMESTAMP) RETURNING *',
      [page_id, JSON.stringify(nodes), JSON.stringify(edges)]
    );

    res.json({ message: 'Данные успешно сохранены!', data: result.rows[0] });
  } catch (error) {
    console.error('Ошибка при сохранении данных:', error);
  }
});

```

```

    res.status(500).json({ error: 'Ошибка сервера при сохранении данных' });
  }
});

```

Этот блок кода принимает от клиента массив нод с данными о текущей странице и записывает их в базу данных, каждый массив записывается под своим уникальным ID, по которому можно удалять конкретный массив, а также выводить историю. Также мы делаем проверку на превышение количества массивов на одной странице, чтобы избежать большого количества повторяющейся информации. Необходимо, чтобы в базе хранилось до пяти изменённых состояний массивов, чтобы отображать историю изменений на конкретной странице.

3. Получение истории сохранений для конкретной страницы.

```

app.get('/api/flow/history/page/:page_id', async (req, res) => {
  const { page_id } = req.params; // Получаем page_id из параметра маршрута
  console.log(req.params)
  try {
    // Запрос для получения истории только для данного page_id
    const result = await pool.query(
      'SELECT page_id, id, created_at FROM node_data WHERE page_id = $1 ORDER BY created_at DESC',
      [page_id] // Фильтруем по page_id
    );
    res.json(result.rows); // Возвращаем ID, page_id и дату создания для выбранного page_id
  } catch (error) {
    console.error('Ошибка при получении истории нод:', error);
    res.status(500).json({ error: 'Ошибка сервера при получении истории нод.' });
  }
});

```

Здесь от клиента в теле запроса получаем page_ID по которому производим сортировку данных и выдаем клиенту только те массивы нод которые принадлежат текущей странице.

4. Поиск по узлам доступа.

Так как интерактивная карта представляет собой ноды как узлы доступа, то по этим узлам можно также перейти, все узлы доступа не рационально выводить на одной странице, так как у одного узла может быть много зависимых узлов [9]. Для этого и необходим поиск, так мы сможем легко переходить внутрь конкретного адреса.

```

app.get('/api/flow/search/:query', async (req, res) => {
  const { query } = req.params;

  try {
    const result = await pool.query(
      `
      SELECT DISTINCT ON (page_id) page_id, created_at
      FROM node_data
      WHERE page_id ILIKE $1
      ORDER BY page_id, created_at DESC
      `,
      [`${query}%` ]
    );

    console.log('Результаты поиска:', result.rows); // Печатаем результат в
    консоль

    res.json(result.rows); // Отправляем результат
  } catch (error) {
    console.error('Ошибка при поиске:', error);
    res.status(500).json({ error: 'Ошибка сервера при поиске страниц' });
  }
});

```

В этом блоке кода от клиента приходит запрос от клиента в теле которого приходит набор символов которые он вводит в поле поиск. По этому набору происходит выборка из базы и выдает самое последнее сохранение текущей страницы. Выбор же узла уже происходит по ID массива.

3.4 Контрольный пример реализации проекта

Разработка интерфейса конечного пользователя

Интерфейс конечного пользователя — это формы и окна, с помощью которых конечный пользователь может взаимодействовать с системой. Также возможно управлять данными или другими пользователи. Можно скачать что, это посредник между человеком и разработанной системой.

1. Форма авторизации

Изначально при входе на сайт нас встречает окно авторизации.

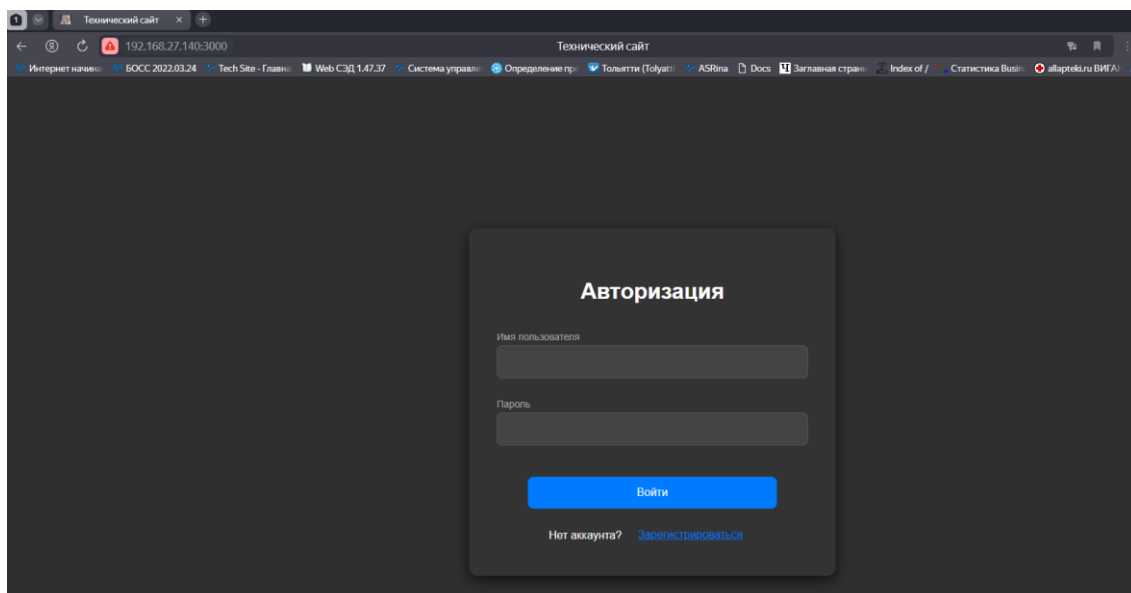


Рисунок 19 – Авторизация

2. Форма регистрации.

Если у пользователя нет учетной записи ему необходимо пройти регистрацию.

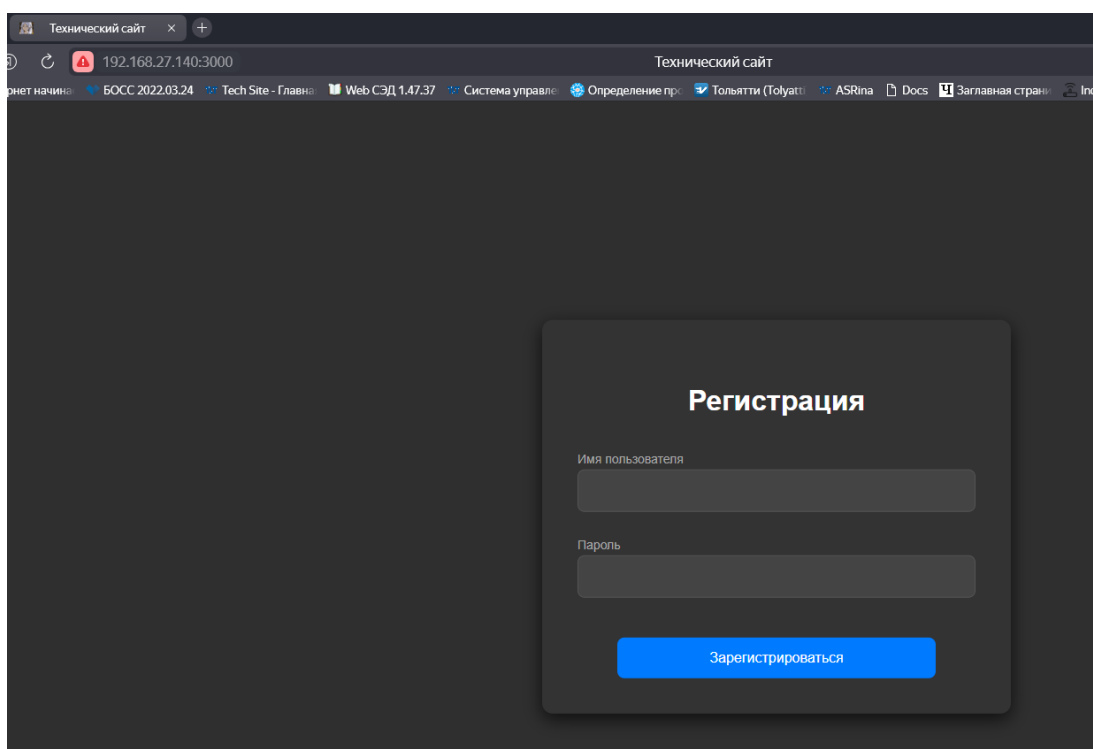


Рисунок 20 – Регистрация

3. Форма админ-панели.

В админ-панели админ может изменить привилегии пользователя или удалить учетную запись, при этом учетные записи админов скрываются.

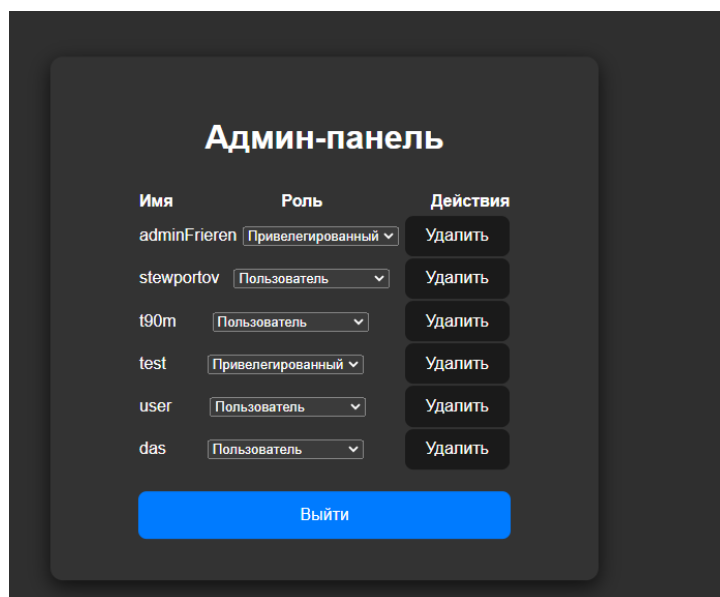


Рисунок 21 – Админ-панель.

4. Меню

После авторизации пользователь попадает в главное меню.

На данном этапе главное меню предоставляет доступ к основным разделам системы:

- показать схемы
- перейти в админ панель

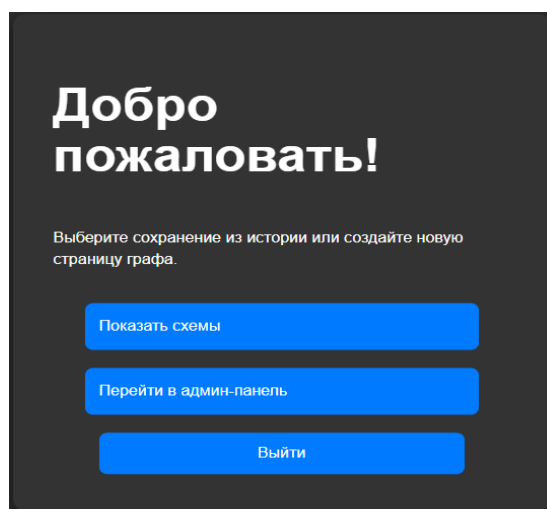


Рисунок 22 – Главное меню

5. Главная схема.

При нажатии на кнопку пользователь попадает на главную схему.

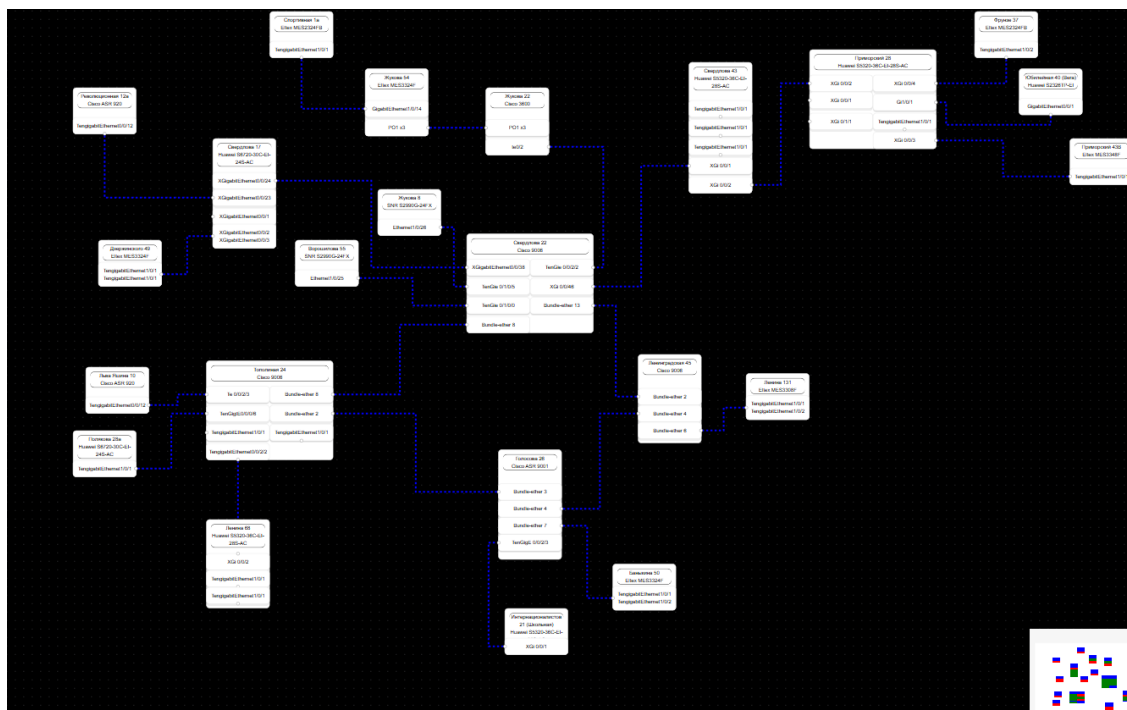


Рисунок 23 – Главная схема

6. Вспомогательная панель.

При авторизации пользователь как админ или привилегированный пользователь, появляется меню с функциональными кнопками.

Вы можете перетаскивать эти узлы на панель справа.

Input Node

Custom Node

Output Node

Выберите сохранение

Страница от: (24.12.2024, 09:40:49) ▾

Объединение объектов

Удаление выбранных объектов

Сохранения данных текущей страницы

Добавить новый адрес

Удалить адрес

Добавить новое оборудование

Удалить оборудование

Рисунок 24 – Toolbar

Поподробнее про каждую кнопку.

Три верхних блока предоставляют возможность перетаскивать на страницу новый порт оборудования.

Далее идет история страницы до пяти сохранений.

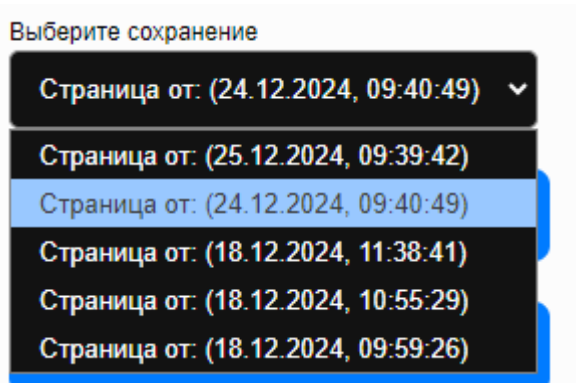


Рисунок 25 – Форма вывода истории.

Кнопка «Объединение объектов». Так как узел доступа представляет собой некий агрегационное оборудование, в которое сходится другие узлы, а оборудование представляет собой совокупность портов, объединённых в группу, то эта кнопка позволяет собрать из независимых портов в одно целое оборудование.

Кнопка «Удаление выбранных объектов» - удаляет выделенный объект.

Кнопка «Сохранение данных текущей страницы» - сохраняет все изменения, произошедшие на странице.

Кнопка «Добавить новый адрес» - позволяет добавить в базу адресов новый адрес.

Кнопка «Удалить адрес» - удаляет из базы адресов выбранный адрес.

Кнопка «Добавить новое оборудование» - позволяет добавить новую модель оборудования в базу.

Кнопка «Удалить оборудование» - удаляет из базы оборудования выбранный оборудование.

Разработка тестов нацеленное на проверку правильной работы запросов и отрисовкой данных на странице. В таблице 2 представлен план тестирования.

Таблица 2 - План тестирования

Наименование теста	Роль	Ожидаемый результат
Просмотр главной схемы	Пользователь	Отрисовка всех данных нод из базы данных по ключу главная схема.
Добавление нового узла доступа	Администратор	Данные успешно сохранены в базу и от сервера пришел ответ.
Добавление нового адреса	Администратор	Данные успешно сохранены в базу и от сервера пришел ответ.
Добавление новой модели оборудования	Администратор	Данные успешно сохранены в базу и от сервера пришел ответ.
Соединение нод друг с другом	Администратор	Добавление двух нод. Изменение полярности у одной и ноды. Соединение двух точек друг с другом.
Изменение положения соединения нод	Администратор	Выбор ноды. По щелчку ЛКМ меняются направление точки.
Выделение родителя и потомка нод	Пользователь	Выбор двух нод или выбор связи этих узлов. Ноды и линия подсвечены.
Объединение в групповую ноду	Администратор	Выделение нескольких нод. После нажатия на кнопку объединение объектов, ноды объединились в одну групповую ноду.
Добавление групповой ноды адреса и модель оборудования	Администратор	После нажатия на верхний блок групповой ноды появляется UI для выбора из списка существующих адресов и моделей оборудования.
Регистрация пользователя	Пользователь	Переход на форму авторизации. Ввод логина и пароля. После регистрации переход на форму авторизации. После авторизации возможность просмотра главной схемы.
Изменение роли пользователя	Администратор	Авторизация под четной записью администратора. Вход в админ панель. Изменение роли пользователя.
Переход по адресу	Пользователь	Нажатие ЛКМ на пустое поле групповой ноды с адресом. Переход на страницу с адресом который на ней указан.

1. Добавление массива нод в базу данных.

При нажатии на кнопку «Сохранение данных текущей страницы» происходит POST запрос на сервер который отправляет JSON объект в котором хранится массив нод и связей

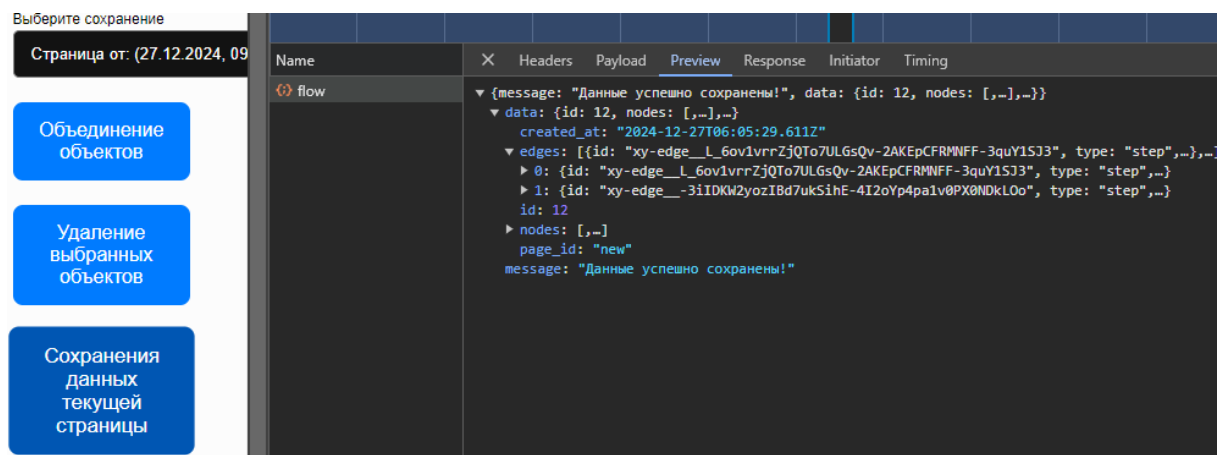


Рисунок 26 – POST запрос на сервер.

Сервер в свою очередь видит что от клиента пришел POST запрос.

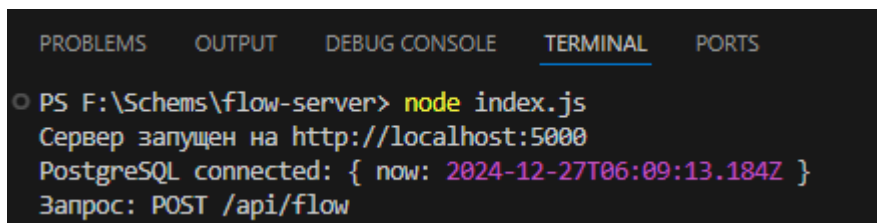


Рисунок 27 – POST запрос на сервере.

В базе происходит сохранение полученного массива нод от клиента.

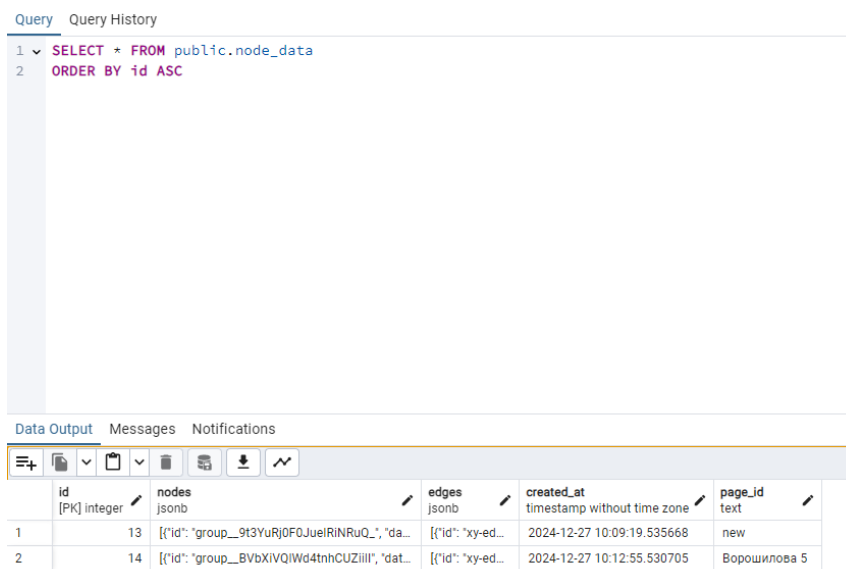


Рисунок 28 – Хранение данных в базе

2. Получение массива нод.

При загрузке страницы конкретного адреса происходит запрос на сервер с ключем page_id по которому сервер находит данные и отправляет новейший массив нод по этому адресу.

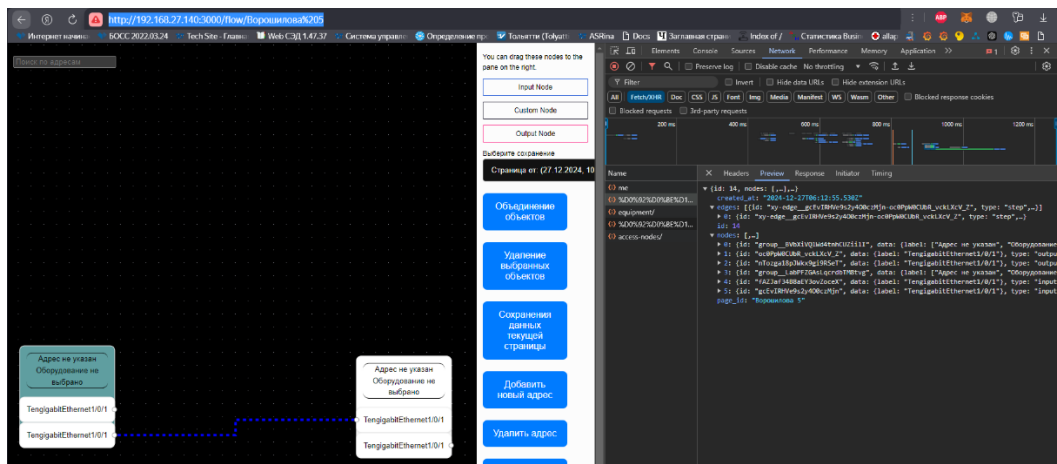


Рисунок 29 – GET запрос на сервер.

Сервер в свою очередь видит, что от клиента пришел GET запрос.

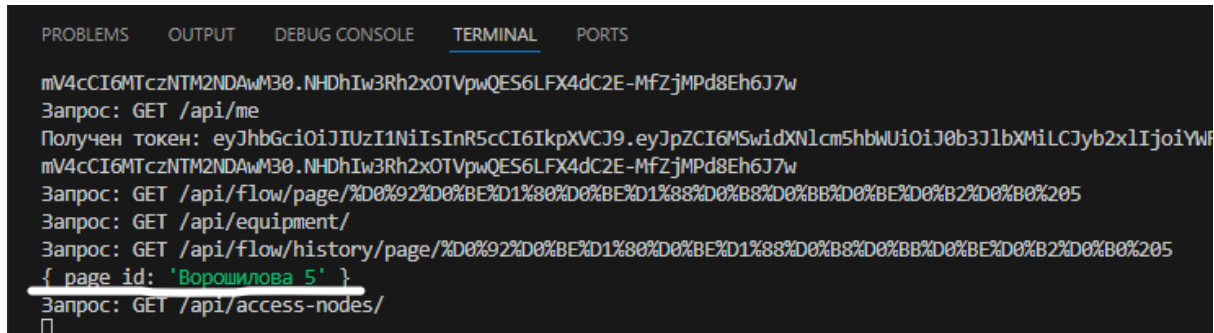


Рисунок 30 – GET запрос на сервере.

Таблица 3 - Результат тестирования

Наименование теста	Результат
Просмотр главной схемы	Тест прошел успешно. Ошибок не обнаружено.
Добавление нового узла доступа	Тест прошел успешно. Ошибок не обнаружено.
Добавление нового адреса	Тест прошел успешно. Ошибок не обнаружено.
Добавление новой модели оборудования	Тест прошел успешно. Ошибок не обнаружено.
Соединение нод друг с другом	Тест прошел успешно. Ошибок не обнаружено.
Изменение положения соединения нод	Тест прошел успешно. Ошибок не обнаружено.

Продолжение таблицы 3

Наименование теста	Результат
Выделение родителя и потомка нод	Тест прошел успешно. Ошибок не обнаружено.
Объединение в групповую ноду	Тест прошел успешно. Ошибок не обнаружено.
Добавление групповой ноде адреса и модель оборудования	Тест прошел успешно. Ошибок не обнаружено.
Регистрация пользователя	Тест прошел успешно. Ошибок не обнаружено.
Изменение роли пользователя	Тест прошел успешно. Ошибок не обнаружено.
Переход по адресу	Тест прошел успешно. Ошибок не обнаружено.

По третьей главе, посвященной архитектуре проекта и особенностям реализации, были сделаны следующие ключевые выводы.

1. Архитектурный подход: Определена трехуровневая архитектура системы, состоящая из уровня представления, уровня логики приложения и уровня данных. Такой подход обеспечивает четкое разделение функций и упрощает дальнейшую поддержку и модификацию системы.

2. Компоненты системы: Подробно описаны основные компоненты системы, такие как модуль управления производственными процессами, модуль учета ресурсов и модуль отчетности. Каждый компонент был спроектирован с учетом функциональных требований, что позволяет эффективно решать задачи автоматизации.

3. Выбор технологий: Выбор технологий для реализации системы был обоснован с точки зрения производительности, удобства использования и возможностей интеграции. Использование фреймворков и библиотек, таких как React для фронтенда и FastAPI для бэкенда, обеспечивает высокую скорость разработки и гибкость.

4. Интеграция и взаимодействие: Предусмотрены механизмы для интеграции с существующими системами и внешними сервисами. Это позволит использовать уже имеющиеся данные и минимизировать дублирование информации, что сократит затраты и время на внедрение.

5. Механизмы безопасности: Уделено внимание вопросам безопасности системы. Реализованы такие механизмы, как аутентификация и авторизация пользователей, шифрование данных и защита от несанкционированного доступа. Это критически важно для защиты корпоративной информации и обеспечения доверия пользователей.

6. Удобство и пользовательский опыт: В дизайне пользовательского интерфейса уделено внимание удобству и интуитивности. Это подразумевает использование адаптивного дизайна, что позволяет системе корректно отображаться на различных устройствах, включая ПК и мобильные платформы.

7. Тестирование и отладка: Описаны планы по тестированию системы, включая юнит-тестирование, интеграционное тестирование и тестирование пользовательского интерфейса. Это гарантирует высокое качество разработки и минимизирует количество ошибок на этапе внедрения.

8. План реализации: Приведен поэтапный план реализации проекта, который включает в себя временные рамки, задачи каждой фазы разработки и ответственных исполнителей. Четкая структура плана позволяет контролировать прогресс и вносить изменения при необходимости.

Глава 4 Оценка экономической эффективности проекта

4.1 Планирование и организация процесса разработки

Эффективность программного средства — это экономическая эффективность ее использования.

Предполагается, что внедрение интерактивной карты позволит оптимизировать рабочий процесс, сэкономить время, облегчить работу, а также избавить от ошибок при использовании ручного ввода.

Для экономического обоснования применения разработанного ПО сравним два варианта работы организации:

- базовый вариант — действия с сетью проводятся без автоматизированных систем работниками;
- проектный вариант — использование автоматизированной системы с интерактивной картой сети;
- анализ экономической эффективности включает:
- планирование и организация разработки ПО;
- оценку затрат на создание системы;
- сравнение капитальных вложений для обоих вариантов;
- расчет изменяющихся затрат;
- определение годового экономического эффекта и других показателей эффективности;
- оценку экономического и социального эффекта.

Планирование и организация процесса разработки.

Для оценки стоимости программного продукта требуется составить план действий.

Необходимо выделить фазы создания и определить их сроки. Перечень задач по разработке информационной системы указан в таблице 4.

Таблица 4 – Перечень работ по созданию ПО

Наименование этапа	Продолжительность, час
Формирование требований	3
Разработка концепции	25
Техническое задание	12
Разработка модели базы данных	17
Разработка проектных решений по системе и ее частям	15
Разработка межпрограммного взаимодействия	15
Разработка интерфейса пользователя	12
Тестирование и отладка ИС	14
Рабочая документация для обучения пользования ИС	13
Итого	126

4.2 Расчет затрат на разработку программного обеспечения

«Затраты на покупные изделия рассчитываются по формуле» [18]:

$$I_{\Pi} = \sum_{i=1}^N n_i \cdot C_i \quad (1)$$

где n_i – количество покупных изделий i -го вида на создание ИС, шт.;

C_i – цена покупных изделий i -го вида, р./шт;

N – общее количество покупных изделий.

Расчет затрат на покупные изделия сведем в таблицу 5

Таблица 5 – Затраты на покупные изделия

Наименование изделий	Количество, шт.	Цена за единицу, руб./шт	Сумма затрат, руб.	Примечание
Бумага «Svetocopy»	1	225	225	500 листов

Продолжение таблицы 5

Заправка картриджа для принтера Ricoh SP 100	1	500	500	Ресурс 800 листов
Итого			775	

Затраты на электроэнергию рассчитываются по формуле:

$$\mathcal{E}_T = \sum_{i=1}^N H_{\mathcal{E}i} \cdot t_i \cdot C_{\mathcal{E}} \quad (2)$$

где $H_{\mathcal{E}i}$ – норма расхода энергии в единицу времени i -го потребителя, кВт/ч;

t_i – время работы i -го оборудования при создании АРМ, ч.;

$C_{\mathcal{E}}$ – цена 1 кВт/ч энергии, руб составляет 2,57;

N – общее количество потребителей энергии.

$$\mathcal{E}_T = \sum_{i=1}^1 0,08 \cdot 126 \cdot 2,57 = 25,9 \text{ руб.}$$

$$\mathcal{E}_T = \sum_{i=1}^1 0,25 \cdot 2 \cdot 2,57 = 1,3 \text{ руб.}$$

Расчет затрат на электроэнергию сведем в таблицу 6.

Таблица 6 – Затраты на электроэнергию

Потребитель	Норма расхода энергии, кВт/ч	Время работы, ч	Цена 1 кВт/ч, руб	Сумма затрат, руб.
Ноутбук ASUS X553SA-XX007D	0.08	126	2,57	25,9
Принтер Ricoh SP 100	0,25	2	2,57	1,3
Итого:				27,2

Основная заработная плата разработчика системы рассчитывается по формуле:

$$З_o = \sum_{i=1}^N C_{\text{ч}} \cdot t_{\text{шт}i} \quad (3)$$

где $C_{\text{ч}}$ – часовая тарифная ставка с учетом доплат, руб./ч;

$t_{\text{шт}i}$ – трудоемкость выполнения i -го вида работ при создании ИС, ч.

$$З_o = \sum_{i=1}^1 80 \cdot 126 = 10080 \text{ руб.}$$

Дополнительная заработная плата определяется по формуле:

$$З_d = \frac{З_o \cdot Н_{\text{зд}}}{100\%} \quad (4)$$

где $Н_{\text{зд}}$ – норматив дополнительной заработной платы, % (9%);

$З_o$ – основная заработная плата работника.

$$З_d = \frac{10080 \cdot 9\%}{100\%} = 907,2 \text{ руб.}$$

Расчет отчислений на социальное страхование ведется по формуле:

$$О_{\text{сс}} = \frac{(З_o + З_d) \cdot Н_{\text{сс}}}{100\%} \quad (5)$$

Где $Н_{\text{сс}}$ – норматив отчислений на социальное страхование, %(26%);

$(З_o + З_d)$ – фонд заработной платы, руб.

$$О_{\text{сс}} = \frac{(10080 + 907,2) \cdot 26\%}{100\%} = 2856,7 \text{ руб.}$$

Для расчета амортизационных отчислений оборудования применяется формула:

$$А_{\text{об}} = \sum \frac{K_{\text{п}} \cdot Н_{\text{А}} \cdot n}{100\% \cdot 12} \quad (6)$$

где $K_{\text{п}}$ – первоначальная стоимость оборудования, руб.;

H_A – норматив амортизационных отчислений на полное восстановление, % (20%);

n – количество месяцев работы оборудования при создании ИС.

$$n = 0,7$$

$$O_{cc} = \sum \frac{27000 \cdot 20\% \cdot 0,7}{100\% \cdot 12} = 315 \text{ руб.}$$

В процессе разработки использовалось следующее оборудование, которое представлено в таблице 7.

Таблица 7 - Перечень используемого оборудования

Наименование	Количество (шт.)	Цена за единицу, руб./шт.	Сумма затрат, руб.	Примечание
Ноутбук ASUS X553SA- XX007D	1	17999	18000	15.6-дюймовый экран, с разрешением 1366x768. ASUS Intel Celeron N3050 2x1.6 ГГц. RAM объемом 4 ГБ. Intel HD.
Принтер Ricoh SP 100	1	1899	1900	Лазерный принтер. Цветность печати черно-белая
Итого:	1	3698	27000	

Затраты на текущий ремонт и обслуживание оборудования рассчитываются по формуле:

$$З_{po} = \frac{K_{п} \cdot H_{po} \cdot n}{100\% \cdot 12} \quad (7)$$

Где $K_{п}$ – первоначальная стоимость оборудования, руб.;

H_{po} – норматив отчислений на ремонт и обслуживание оборудования, % (5%);

n – количество месяцев работы оборудования при создании ИС:

$$З_{ро} = \frac{27000 \cdot 5\% \cdot 0,7}{100\% \cdot 12} = 78,8 \text{ руб.}$$

Накладные расходы рассчитываются по формуле:

$$P_H = \frac{З_о \cdot N_{НР}}{100\%} \quad (8)$$

где $N_{НР}$ – норматив накладных расходов, %(50%).

$$P_H = \frac{10080 \cdot 50\%}{100\%} = 5040 \text{ руб.}$$

Результаты расчетов по всем статьям затрат сведем в таблицу 8.

Таблица 8 – Себестоимость разработки

Наименование статей затрат	Обозначение	Сумма, руб.
Покупные изделия	I_n	775
Затраты на энергию	$Э_r$	27,2
Основная заработная плата	$З_о$	10080
Дополнительная заработная плата	$З_d$	907,2
Отчисления на социальное страхование	O_{cc}	2856,7
Амортизация	$A_{об}$	315
Затраты на текущий ремонт и обслуживание оборудования	$З_{ро}$	78,8
Накладные расходы	P_H	5040
Итого:		20079,9

Далее следует рассчитать прибыль, которую организация получит в результате внедрения автоматизированной системы.

Прибыль рассчитывается по формуле:

$$\Pi = \frac{Y_P \cdot C}{100\%} \quad (9)$$

где Π – прибыль от внедрения автоматизированной системы;

Y_P – уровень рентабельности – (условно 10%);

C – себестоимость разработки автоматизированной системы.

$$\Pi = \frac{10\% \cdot 20079,9}{100\%} = 2007,99 \text{ руб.}$$

Цена информационной системы будет соответствовать сумме себестоимости разработки и прибыли:

$$\mathcal{C} = C + \Pi \quad (10)$$

$$\mathcal{C} = 20079,9 + 2007,99 = 22087,89 \text{ руб.}$$

4.3 Расчет общих капитальных вложений по вариантам

Сравнение проводится по двум вариантам:

Базовый вариант – это ручное управление без затрат на программное обеспечение. В этом случае все процессы, связанные с управлением, осуществляются вручную, что требует значительных временных затрат и человеческих ресурсов. Работники должны выполнять все операции по контролю и учету данных с использованием только бумажных носителей или традиционных методов. Это может привести к ошибкам, связанным с человеческим фактором, и увеличению времени, необходимого для обработки информации. Однако такой подход имеет свои преимущества, такие как отсутствие первоначальных вложений в программное обеспечение и возможность избежать технических сбоев в работе системы.

Проектный вариант – это программное обеспечение, разработанное в процессе выполнения выпускной квалификационной работы. Использование разработанного ПО автоматически упрощает процессы управления, позволяя сократить время на выполнение задач. Эта система обеспечивает более точный учет и контроль данных, так как все операции выполняются с использованием автоматизированных алгоритмов. Кроме того, программное обеспечение предлагает функционал для создания отчетов, анализа данных и прогнозирования, что значительно улучшает принятие решений.

Сравнение между двумя вариантами показывает, что проектный вариант предлагает множество преимуществ. Во-первых, он способствует снижению затрат времени на выполнение рутинных задач. Автоматизация процессов позволяет работникам сосредоточиться на более важных аспектах своей работы, повышая общую продуктивность. Во-вторых, использование ПО снижает вероятность ошибок, связанных с ручным вводом данных, что также может положительно сказаться на качестве итоговых результатов.

Важно отметить, что программное обеспечение может быть адаптировано под специфические нужды компании, что делает его универсальным инструментом для различных бизнес-процессов. Это также позволяет интегрировать новое ПО с существующими системами, что обеспечивает плавный переход и минимизирует риски, связанные с обучением сотрудников. Кроме того, возможность обновления и доработки программного обеспечения открывает перспективы для дальнейшего улучшения функционала.

Сравнение также стоит провести с точки зрения затрат. Несмотря на то, что проектный вариант может потребовать начальных вложений на стадии разработки, в долгосрочной перспективе он может значительно сократить общие затраты. А именно, благодаря увеличению скорости обработки данных и сокращению необходимости в большом количестве сотрудников для выполнения рутинных задач. Более того, внедрение автоматизированной системы позволяет добиться лучшей управляемости проектами и более эффективного использования ресурсов компании.

Безусловно, внедрение нового ПО требует времени на обучение сотрудников, однако современные системы интуитивно понятны и просты в использовании. Это позволяет быстро интегрировать программное обеспечение в рабочие процессы без значительных временных затрат.

В заключение, сравнение базового варианта и проектного варианта показывает, что хотя ручной метод управления может быть экономически выгоднее на первом этапе, внедрение программного обеспечения

представляет собой более рациональное решение в долгосрочной перспективе. Автоматизация процессов не только оптимизирует работу, но также способствует повышению конкурентоспособности компании на рынке.

Формула для расчета капитальных вложений в проектом варианте:

$$K = \sum_i K_i, \quad (\Pi) \quad (11)$$

Капитальные вложения на оборудование рассчитываются по следующей формуле:

$$K = \sum_i n_{\text{ОБ}} \cdot K_{\text{П}} \cdot \eta_i \quad (12)$$

где $n_{\text{ОБ}}$ – количество оборудования, шт.;

$K_{\text{П}}$ – первоначальная стоимость оборудования, руб.;

η_i – коэффициент использования оборудования.

$$K = \sum_i 7 \cdot 19290 \cdot 1 = 135030 \text{ руб.} \quad \text{для базового варианта;}$$

$$K = \sum_i 6 \cdot 19290 \cdot 1 = 115740 \text{ руб.} \quad \text{для проектного варианта;}$$

$$K = \sum_i 1 \cdot 11190 \cdot 1 = 11190 \text{ руб.} \quad \text{для базового варианта;}$$

$$K = \sum_i 1 \cdot 11190 \cdot 1 = 11190 \text{ руб.} \quad \text{для проектного варианта;}$$

Расчет капитальных вложений на оборудование сведем в таблицу 9.

Таблица 9 – Капитальные вложения на оборудование

Наименование	Необходимое количество, шт.		Цена, руб./шт.	Сумма, руб.	
	Базовый вариант	Проектный вариант		Базовый вариант	Проектный вариант
19.5" Моноблок Lenovo C20-00	7	6	19290	135030	115740
МФУ XEROX WorkCentre 3025	1	1	11 190	11 190	11 190
Итого:				146220	126930

Базовый вариант не предполагает капитальных вложений на программное обеспечение – $K'_{по} = 0$.

Капитальные вложения для проектного варианта рассчитываются по формуле:

$$K_{по} = C_{ас} + Z_{обуч} \quad (13)$$

где $C_{ас}$ – стоимость программного продукта, руб.;

$Z_{обуч}$ – затраты на обучение персонала работе с программным продуктом, руб.

Стоимость разработки была определена в предыдущем подразделе.

Затраты на обучение отсутствуют, так как разработано руководство пользователя, включенное в стоимость разработки. Разработка руководства пользователя проводилась в рамках создания, поэтому затраты на разработку руководства пользователя включены в затраты, $Z_{обуч} = 0$

$$K_{по} = 22087,89 \text{ руб.}$$

При подсчёте вложений в площадь учитывается неизменность размеров рабочих мест при любом варианте внедрения.

Затраты на площадь определяются по формуле:

$$K_s = \sum_i S_i \cdot n_i \cdot C_{м^2} \cdot \eta_i \quad (14)$$

где S_i – площадь i -го рабочего места $м^2$;

n_i – количество рабочих мест для сотрудников, шт.;

$C_{м^2}$ – цена площади производственных помещений, 450 руб./ $м^2$;

η_i – коэффициент использования площади i -м оборудованием. (1)

$$K_{s2} = \sum_i 6 \cdot 7 \cdot 450 \cdot 1 = 14175 \text{ руб.}$$

$$K_{s1} = \sum_i 6 \cdot 6 \cdot 450 \cdot 1 = 12150 \text{ руб.}$$

Все элементы капитальных вложений сведем в таблицу 10.

Таблица 10 – Общие капитальные вложения по вариантам

Элементы капитальных вложений	Величина затрат, руб.	
	Базовый вариант	Проектный вариант
Оборудование	146220	126930
Программное обеспечение	0	22087,89
Площадь	14175	12150
Итог:	160395	161167,89

Расчет затрат будет вестись относительно базового варианта (ручной ввод) и проектного варианта с помощью ПО.

При расчете затрат использовались исходные данные, приведенные в таблице 11 и 12.

Таблица 11 – Исходные данные

Наименование	Единица измерения	Значение	Примечание
1 Стоимость 1 кВт/ч электроэнергии	руб.	2,57	По данным предприятия
2 Часовая тарифная ставка программиста	руб./ч.	80	По исходным данным
3 Годовой действительный фонд времени работы оборудования	ч.	1600	По исходным данным
4 Количество работников	шт.	6	По исходным данным
5 Режим работы	ч.	6	По исходным данным

Таблица 12 – Трудоемкость по базовому и проектному вариантам

Наименование этапа	Продолжительность этапа за месяц, мин.	
	Базовый вариант	Проектный вариант
Подготовка помещения	50	20
Получение и обработка данных	70	10
Сбор информации, запрос в архив	120	40
Внесения данных в базу данных	45	45
Поиск информации в базе данных	150	105
Сортировка информации	140	40
Вывод данных	210	150
Утверждение готовой отчетности	400	200
Сверка данных	40	15
Итог мин/ч	1225/53,26	625/27,17

Расчет затрат на покупные изделия проводится по формуле:

$$I_{\Pi} = \sum_{i=1}^N n_i \cdot C_{\Pi i} \quad (15)$$

где n_i – количество покупных изделий i -го вида, шт.;

$C_{\Pi i}$ – цена покупных изделий i -го вида, р./шт.;

N – общее количество покупных изделий.

Расчет затрат на покупные изделия для базового варианта:

$$I_{\Pi} = \sum_{i=1}^2 2 \cdot 500 = 1000 \text{ руб.}$$

$$I_{\Pi} = \sum_{i=1}^2 2 \cdot 225 = 450 \text{ руб.}$$

Расчет затрат на покупные изделия для проектного варианта:

$$I_{\Pi} = \sum_{i=1}^1 1 \cdot 225 = 225 \text{ руб.}$$

$$И_{\Pi} = \sum_{i=1}^1 1 \cdot 500 = 500 \text{ руб.}$$

Расчет затрат на покупные изделия сведем в таблицу 13.

Таблица 13 – Затраты на покупные изделия

Наименование изделия	Цена, руб./ шт.	Количество, шт.		Сумма затрат, руб.	
		Базовый вариант	Проектный вариант	Базовый вариант	Проектный вариант
Картридж (заправка)	500	2	1	1000	500
Бумага	225	2	1	450	225
Итого:				1450	775

Затраты на энергию рассчитываются по следующей формуле:

$$\mathcal{E}_{\Gamma} = \sum_{i=1}^N H_{\mathcal{E}i} \cdot t_i \cdot \mathcal{C}_{\mathcal{E}}, \quad (16)$$

где $H_{\mathcal{E}i}$ – норма расхода энергии в единицу времени i -го потребителя, кВт/ч (по данным таблицы 5);

t_i – время работы i -го оборудования, ч.;

$\mathcal{C}_{\mathcal{E}}$ – цена 1 кВт/ч энергии, руб.;

N – общее количество потребителей энергии.

Расчет затрат на электроэнергию для базового варианта:

$$\mathcal{E}_{\Gamma} = \sum_{i=1}^7 0,15 \cdot 6 \cdot 2,57 = 16,191 \text{ руб.}$$

$$\mathcal{E}_{\Gamma} = \sum_{i=1}^1 0,25 \cdot 6 \cdot 2,57 = 3,855 \text{ руб.}$$

$$\mathcal{E}_{\Gamma} = 20,046 \text{ руб.}$$

Расчет затрат на электроэнергию для проектного варианта:

$$\Xi_{\tau} = \sum_{i=1}^6 0,15 \cdot 6 \cdot 2,57 = 13,878 \text{ руб.}$$

$$\Xi_{\tau} = \sum_{i=1}^1 0,25 \cdot 6 \cdot 2,57 = 3,855 \text{ руб.}$$

$$\Xi_{\tau} = 17,733 \text{ руб.}$$

Основная заработная плата рассчитывается по формуле:

$$З_o = \sum_{i=1}^N C_{\text{ч}} \cdot t_{\text{шт}i} \quad (17)$$

где C – часовая тарифная ставка с учетом доплат, руб./ч. (по данным таблицы 8);

$t_{\text{шт}i}$ – трудоемкость выполнения i -го вида работ (по данным таблицы 10).

Расчёты основной заработной платы будет проводиться в базовом варианте для 7 человек, как в исходных данных с тарифной ставкой в 80 руб./ч, а в проектном варианте число людей сокращается на одного, но тарифная ставка остаётся та же.

Базовый вариант:

$$З_o = \sum_{i=1}^7 80 \cdot 53,26 = 29825,6 \text{ руб.}$$

Проектный вариант:

$$З_o = \sum_{i=1}^6 80 \cdot 27,17 = 13041,6 \text{ руб.}$$

Расчет основной заработной платы сведем в таблицу 14.

Таблица 14 – Затраты на основную заработную плату

Варианты	Часовая тарифная ставка, руб./ч	Трудоемкость, ч.	Сумма затрат, руб.
Базовый вариант	80 руб.	53,26	29825,6
Проектный вариант	80 руб.	27,17	13041,6

Дополнительная заработная плата определяется по формуле:

$$З_д = \frac{З_о \cdot Н_{зд}}{100\%} \quad (18)$$

где $H_{зд}$ – норматив дополнительной заработной платы,

$З_о$ – основная заработная плата работника.

Расчет затрат на дополнительную заработную плату:

Базовый вариант:

$$З_д = \frac{29825,6 \cdot 9\%}{100\%} = 2684,304 \text{ руб.}$$

Проектный вариант:

$$З_д = \frac{13041,6 \cdot 9\%}{100\%} = 1173,744 \text{ руб.}$$

Расчет отчислений на социальное страхование ведется по формуле:

$$О_{сс} = \frac{(З_о + З_д) \cdot Н_{сс}}{100\%} \quad (19)$$

где $H_{сс}$ – норматив отчислений на социальное страхование

$(З_о + З_д)$ – фонд заработной платы, руб.

Расчет размера отчислений на социальное страхование:

Базовый вариант:

$$О_{сс} = \frac{(29825,6 + 2684,304) \cdot 26\%}{100\%} = 8452,575 \text{ руб.}$$

Проектный вариант:

$$О_{сс} = \frac{(13041,6 + 1173,744) \cdot 26\%}{100\%} = 3695,989 \text{ руб.}$$

Амортизационные отчисления рассчитываются по формуле:

$$А_{об} = \sum \frac{K_{п} \cdot H_A \cdot t}{100\% \cdot F_d} \quad (20)$$

где $K_{п}$ – первоначальная стоимость используемого оборудования, руб.;

H_A – норматив амортизационных отчислений на полное восстановление;

tt – суммарное время использования оборудования, ч.

Базовый вариант:

$$A_{об} = \frac{146220 \cdot 20\% \cdot 53,26}{100\% \cdot 1600} = 973,46 \text{ руб.}$$

Проектный вариант:

$$A_{об} = \frac{126930 \cdot 20\% \cdot 27,17}{100\% \cdot 1600} = 431,1 \text{ руб.}$$

Расчет затрат на амортизационные отчисления сведем в таблицу 15.

Таблица 15 – Амортизационные отчисления

Варианты	Кп, руб.	Н _А , %	t , ч.	F_d , ч.	Амортизационные отчисления, руб.
Базовый вариант	146220	20%	53,26	1600	973,46
Проектный вариант	126930		27,17		431,03

Отчисления на текущий ремонт рассчитываются по формуле:

$$A_{тр} = \frac{K_p \cdot H_{тр} \cdot t}{100\% \cdot F_d} \quad (21)$$

где K_p – первоначальная стоимость оборудования руб.;

$H_{тр}$ – норма отчислений на текущий ремонт, (по данным таблицы 10);

tt – суммарное время использования оборудования, ч.;

F_d – действительный годовой фонд времени работы оборудования, ч.

Базовый вариант:

$$A_{тр} = \frac{146220 \cdot 5\% \cdot 53,26}{100\% \cdot 1600} = 243,36 \text{ руб.}$$

Проектный вариант:

$$A_{тр} = \frac{126930 \cdot 5\% \cdot 27,17}{100\% \cdot 1600} = 107,77 \text{ руб.}$$

Таблица 16 – Отчисления на текущий ремонт

Вариант	Кп, руб.	Нтр, %	t , ч.	F_{∂} , ч.	Амортизационные отчисления, руб.
Базовый вариант	146220	5%	53,26	1600	243,36
Проектный вариант	126930		27,17		107,77

Ежемесячные затраты на программное обеспечение вычисляются по следующей формуле:

$$P_{\text{по}} = \frac{C_{\text{по}} \cdot k_{\text{по}}}{12} \quad (22)$$

где $C_{\text{по}}$ – стоимость программного обеспечения, руб.;

$k_{\text{по}}$ – коэффициент учета стоимости автоматизированной системы;

Коэффициент стоимости обратно пропорционален сроку службы: для АРМ это 4 года (с учётом устаревания), что даёт значение 0,25.

Расчет затрат на программное обеспечение:

$$P_{\text{по}} = \frac{22087,89 \cdot 0,25}{12} = 460,16 \text{ руб.}$$

4.4.8 Амортизация площади рабочего места

Амортизационные отчисления рассчитываются по формуле:

$$A_s = \frac{S \cdot C_{\text{м}^2} \cdot H_s \cdot t}{100\% \cdot F_{\partial}} \quad (23)$$

где S – площадь рабочего места, м^2 ;

$C_{\text{м}^2}$ – цена площади производственных помещений, руб./ м^2 (цена площади составляет 450 руб./ м^2);

H_s – норма амортизации площади, 1%;

t – суммарное время использования оборудования ч.;

F_{∂} – действительный годовой фонд времени работы оборудования, ч.

$$A_s = \frac{36 \cdot 450 \cdot 0,01 \cdot 53,26}{100 \cdot 1600} = 4,5 \text{ руб}$$

$$A_s = \frac{36 \cdot 450 \cdot 0,01 \cdot 27,17}{100 \cdot 1600} = 2,3 \text{ руб.}$$

Расчет затрат на амортизацию рабочего места сведем в таблицу 17.

Таблица 17 – Амортизация площади рабочего места

Варианты	$S, \text{ м}^2$	$H_s, \%$	$t, \text{ ч.}$	$F_\partial, \text{ ч.}$	Амортизационные отчисления, руб.
Базовый вариант	36	1%	53,26	1600	4,5
Проектный вариант	36		27,17		2,3

Результаты всех расчетов сведем в таблицу 18.

Таблица 18 – Ежемесячные затраты по изменяющимся элементам

Наименование элемента затрат	Затраты по изменяющимся элементам, руб.	
	Базовый вариант	Проектный вариант
Покупные изделия	1450	775
Электроэнергия	20,046	17,733
Основная заработная плата	29825,6	13041,6
Дополнительная заработная плата	2684,304	1173,744
Отчисления на социальное страхование	8452,575	3695,989
Амортизация оборудования	973,46	431,1
Текущий ремонт	243,36	107,77
Затраты на программное обеспечение	0	460,16
Амортизация площади рабочего места	4,5	2,3
Итого:	43653,845	32655,738

Остальные элементы затрат не изменяются.

Процент снижения затрат от применения интерактивной карты рассчитывается по формуле:

$$\Delta C = \frac{C' - C''}{C'} \cdot 100\% \quad (24)$$

где C' – текущие затраты по базовому варианту, руб. (таблица 18);

C'' – текущие затраты по проектному варианту, (таблица 18);

В результате использования информационной системы снижения затрат составит:

$$\Delta C = \frac{43653,845 - 32655,738}{43653,845} \cdot 100\% = 25,2\%$$

4.5 Расчет годового экономического эффекта и прочих показателей экономической эффективности

Для принятия решения о внедрении используется показатель годовой выгоды, определяемый сравнением текущего и предлагаемого подходов.

Ожидаемый годовой экономический эффект от применения информационной системы рассчитывается по формуле:

$$\mathcal{E}_Г = [(C' + E_H \cdot K'_y) - (C'' + E_H \cdot K''_y)] \cdot 12 \quad (25)$$

Где C – ежемесячные затраты, руб.;

E_H – нормативный коэффициент экономической эффективности капитальных вложений. $E_H = 0,33$;

K_y – удельные капитальные вложения, руб.,

$$K_y = \frac{K_0}{12} \quad (26)$$

где K_0 – общие капитальные вложения, руб.

Базовый вариант:

$$K'_y = \frac{160395}{12} = 13366,25 \text{ руб.}$$

Проектный вариант:

$$K''_y = \frac{161167,89}{12} = 13430,65 \text{ руб.}$$

Таким образом, ожидаемый годовой экономический эффект составит:

$$\mathcal{E}_Г = \left[\frac{(43653,845 + 0,33 \cdot 13366,25) - (32655,738 + 0,33 \cdot 13430,65)}{12} \right] \cdot 12 = 131722,2$$

Условно-годовой экономический эффект от применения ПО рассчитываем по формуле:

$$\mathcal{E}_{\text{уг}} = (C' - C'') \cdot 12 \quad (27)$$

где C'_1 – себестоимость на единицу работы по базовому варианту, руб. (таблица 16);

C''_1 – затраты на единицу работы по проектному варианту, руб.;

Условно-годовой экономический эффект от применения ПО равен:

$$\mathcal{E}_{\text{уг}} = (43653,845 - 32655,738) \cdot 12 = 131976,78 \text{ руб.}$$

Срок окупаемости дополнительных капитальных вложений вычисляется по формуле:

$$T = \frac{\Delta K}{\mathcal{E}_{\text{уг}}} \quad (28)$$

где ΔK – дополнительные капитальные вложения на разработку и внедрение системы, руб. ;

$\mathcal{E}_{\text{уг}}$ – условно-годовой экономический эффект (руб.).

$$T = \frac{22087,89}{131976,78} = 0,167 \text{ года.}$$

Итак, срок окупаемости дополнительных капитальных вложений на разработку и внедрение составит – 2 месяца.

Коэффициент экономической эффективности капитальных вложений на разработку:

$$E = \frac{\mathcal{E}_{\text{уг}}}{\Delta K} \quad (29)$$

$$E = \frac{131976,78}{22087,89} = 5,98\%$$

Процент снижения трудоемкости рассчитывается по формуле:

$$\Delta t = \frac{t' - t''}{t'} \cdot 100\% \quad (30)$$

где t' – трудоемкость по базовому варианту, ч.;

t'' – трудоемкость по проектному варианту, ч.;

$$\Delta t = \frac{53,26 - 27,17}{53,26} \cdot 100\% = 48,98\%$$

Рост производительности труда определяется по формуле:

$$\Delta П = \frac{\Delta t}{100\% - \Delta t} \cdot 100\% \quad (31)$$

где Δt – процент снижения трудоемкости, %.

$$\Delta П = \frac{48,98}{100\% - 48,98} \cdot 100\% = 96\%$$

4.6 Экономический эффект

Внедрение интерактивной карты сети является экономически целесообразным. Основные преимущества такого введения:

Экономический эффект от внедрения программы проявляется в:

- возросла производительность труда.

Все показатели экономической эффективности сведем в таблицу 19.

Таблица 19 – Показатели экономической эффективности

Наименование показателей	Обозначение	Проектный вариант
Годовой экономический эффект	Руб.	131722,26
Условно-годовой экономический эффект	Руб.	131976,78
Срок окупаемости дополнительных капитальных вложений	Год.	0,167
Рост производительности труда	%	97

Социальный эффект от внедрения программы проявляется в:

- упрощении создания отчетов;
- экономии времени рабочего персонала;
- повысилась степень защищенности информации;
- произошло исключение ошибок по вине человека.

В четвёртой главе работы, посвященной оценке экономической эффективности проекта, были сделаны следующие ключевые выводы.

1. Тщательный анализ затрат: Проведен детальный анализ всех затрат, связанных с реализацией проекта. Это включает в себя первоначальные инвестиции, расходы на разработку и внедрение, затраты на обучение сотрудников и техническую поддержку. Такой подход позволяет получить полное представление о финансовых вложениях.

2. Оценка ожидаемых выгод: Определены ключевые преимущества, которые может принести автоматизация производственных процессов, такие как снижение производственных затрат, увеличение производительности, улучшение качества продукции и снижение времени на выполнение задач.

3. Сроки окупаемости: Рассчитан срок окупаемости инвестиций, что позволяет понять, сколько времени потребуется для возврата вложенных средств. Полученные результаты показывают, что проект способен окупиться в разумные сроки, что делает его привлекательным для реализации.

4. Влияние на оперативные процессы: Оценка экономической эффективности показала, что автоматизация позволит существенно оптимизировать рабочие процессы, что приведет к сокращению времени выполнения заказов и улучшению сервиса для клиентов.

5. Косвенные эффекты: Выявлены не только прямые финансовые преимущества, но и косвенные эффекты, такие как повышение уровня удовлетворенности клиентов, улучшение имиджа компании и возможность адекватного реагирования на изменения на рынке.

6. Сравнительный анализ: Проведён сравнительный анализ эффективности автоматизированной системы по сравнению с традиционными методами управления производством. Результаты подтвердили, что автоматизация обеспечивает значительное снижение затрат и улучшение качества работы.

7. Оценка рисков: В ходе анализа также были определены потенциальные риски, связанные с реализацией проекта, такие как технологические сбои, ошибки в процессе внедрения и изменения в

потребительских предпочтениях. Эти риски требуют разработки стратегии управления для их минимизации.

8. Долгосрочные перспективы: Оценка показала, что автоматизация не только оптимизирует текущие процессы, но и открывает новые возможности для роста и развития в будущем. Это создает положительный прогноз для финансовых показателей предприятия в долгосрочной перспективе.

9. Общая жизнеспособность проекта: В заключение, оценка экономической эффективности проекта подтвердила его целесообразность и жизнеспособность. Преимущества автоматизации перевешивают затраты, что делает проект привлекательным для инвестиций и измеряемым в финансовых результатах.

Таким образом, четвёртая глава подводит итог экономической эффективности разработки системы автоматизации производства, основываясь на комплексном анализе затрат и выгод, что подтверждает необходимость и целесообразность реализации данного проекта. Полученные результаты могут служить основой для принятия стратегических решений о дальнейшем развитии компании.

Заключение

Данная бакалаврская работа была посвящена созданию программы для отдела СПД компании «ИнфоЛада», чтобы помочь им быстрее находить и устранять сбои в сети. Цель — объединить данные о сети в одном месте, дать инженерам удобный инструмент и сократить время простоя.

Для изучения работы отдела СПД необходимо было разобраться как устроен отдел СПД и что мешает работе. У них есть биллинг, который видит, что коммутатор не доступен, но не показывает, где он находится. Также есть техсайт с картой сети, но её нужно обновлять вручную. Из-за этого поиск поломки занимает значительное количество времени. Проблема в том, что системы не связаны, и данные разбросаны. Для этого нужна единая платформа для мониторинга.

Было решено, что программа должна показывать сеть как карту, где каждый коммутатор — это точка, а линии — связи между ними. Данные о сбоях должны браться из Prometheus, который уже стоит у «ИнфоЛады». Ещё нужно, чтобы администраторы могли настраивать систему, инженеры — видеть поломки и чинить их, а операторы — смотреть состояние.

Список требований которые необходимо было реализовать:

- карта сети, которую можно масштабировать;
- выделение сбоев цветом (например, красным);
- история изменений карты (хранить до 5 версий);
- разделение прав доступа для разных ролей.

Для разработки был выбран React.js — он удобен для интерактивных интерфейсов, вроде карты. Backend сделал на Node.js, чтобы быстро обрабатывать данные от Prometheus. Для хранения информации настроил PostgreSQL с таблицами:

- узлы сети (коммутаторы, их состояние, связи);
- адреса (где стоит оборудование);
- модели коммутаторов (технические данные).

Был создан ролевой доступ: каждый пользователь видит только то, что ему нужно. Например, оператор не может менять настройки. Всё спроектировал так, чтобы части системы (frontend, backend, база) работали независимо, но слаженно.

В итоге создано веб-приложение, которое рисует сеть в виде графа. Коммутаторы — это точки, а связи — линии. Если коммутатор ломается, точка становится красной, и инженер сразу видит проблему. Данные поступают из Prometheus без задержек. Была добавлена функция, чтобы программа сохраняла до 5 старых версий карты — это помогает сравнить, что изменилось. Ролевой доступ был настроен так, чтобы никто не путался в функциях. Код написан с упором на скорость: обновление данных занимает меньше секунды.

Было проведено тестирование системы в несколько этапов:

- как карта отображает тестовые данные;
- эмуляция сбоев через Prometheus, чтобы убедиться, что уведомления приходят;
- тестирование с реальными данными «ИнфоЛады»;

Также была написана инструкция для «ИнфоЛады»:

- как установить Node.js и PostgreSQL;
- как подключить Prometheus;
- как обучить сотрудников;
- внедрение системы на тестовом стенде.

Для проектирования приложения в начале необходимо было собрать информацию: изучить документы СПД, списки коммутаторов, настройки Prometheus. После поэкспериментировать с разными библиотеками для интерактивной карты, и скорость загрузки ПО. Для анализа работы необходимо было сравнить сколько времени уходит на поиск сбоев с программой и без неё. Проанализировал аналоги, вроде Cisco Prime. но для «ИнфоЛады» слишком сложные и дорогие. Данная система проще, работает с оборудованием «Инфолада» и не требует больших затрат.

После внедрения приложения вся сеть видна на одной карте, и даже новичок поймёт, где что стоит. Инженерам не нужно проверять биллинг и техсайт отдельно — всё в одном месте. Это экономит время, а клиенты меньше ждут. В теории было показано, что можно сделать недорогую систему для небольших провайдеров, которая не уступает дорогим аналогам. На практике — это рабочий инструмент, который уже используют в «ИнфоЛаде» и могут доработать для других.

Разработанное программное средство не заменит Cisco Prime или SolarWinds — так как они для больших сетей, с большим количеством узлов. Но для «ИнфоЛады» данное решение лучше: оно проще, дешевле и заточено под их Prometheus.

Результатом выполненной работы была создана система, которая решает проблему отдела СПД «ИнфоЛады». Она показывает сеть, выделяет сбои и экономит время. Все задачи выполнены: программа работает, протестирована и внедрена. Главное — она помогает инженерам, и улучшает их работу.

Список используемой литературы

1. Андрианова Ю. С., Морозова А.А., Евдошенко О. И. Проектирование информационных систем [Электронный ресурс] : учебно – методическое пособие / составители, – Астрахань : Астраханский государственный архитектурно – строительный университет, ЭБС АСВ, 2022
2. Белый, Е. М. Управление проектами : конспект лекций / Е. М. Белый, И. Б. Романова. – Москва : Ай Пи Ар Медиа, 2023. – 95 с. – (ФГОС ВО). – ISBN 978-5-4497-1879.
3. Библиотека express js [Электронный ресурс] / URL: <https://expressjs.com/en/api.html> (дата обращения: 20.09.2024).
4. Блинов А.О. Реинжиниринг бизнес – процессов [Электронный ресурс]: учеб. пособие / [и др.] ; под ред. А. О. Блинова. – Москва : ЮНИТИ – ДАНА, 2017.
5. Воробьёв, В.П. Моделирование бизнес-процессов с использованием нотации IDEF0. Учебное пособие. Москва: КНОРУС, 2020.
6. ГОСТ Р 7.0.100–2018. Система стандартов по информации, библиотечному и издательскому делу. Национальный стандарт Российской Федерации. Москва, 2018.
7. ГОСТ Р 59502-2021 Единая система условных обозначений в области информационно-телекоммуникационных систем, 2021
8. Грекул, В. И. Управление внедрением информационных систем : учеб. пособие / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – 4-е изд. стер. электрон. – Москва : Интернет-Университет Информационных Технологий [и др.], 2025. – 277 с. – ISBN 978-5-4497-0910-3.
9. Долженко А. И. Технологии командной разработки программного обеспечения информационных систем [Электронный ресурс] : [курс лекций] / А. И. Долженко. – 3 – е изд. – Москва : ИНТУИТ , 2019.
10. Изучение основ языка js. [Электронный ресурс] / URL: <https://learn.javascript.ru/> (дата обращения: 1.09.2024).

11. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам. учебное пособие [Электронный ресурс] : [учеб. пособие] / Ю. А. Маглинец – 3 – е изд.. – Москва, Саратов : ИНТУИТ, Ай Пи Ар Медиа, 2020.
12. Михеев А.Г. Процессное управление на свободном программном обеспечении / Михеев А.Г.. – Москва : Интернет – Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019.
13. Обеспечению оптимизации нативного языка js. [Электронный ресурс] / URL: <https://lodash.com/> (дата обращения: 16.11.2024).
14. Руководство по использованию программного обеспечения Ramus / URL: <https://ramus.ru/docs> (дата обращения: 12.10.2024).
15. Библиотека PostgreSQL / URL: <https://www.postgresql.org/about/> (дата обращения: 12.10.2024).
16. Руководство по использованию библиотеки Node.js / URL: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs> (дата обращения: 12.10.2024).
17. Смирнов, Е. А. Современные методы автоматизации бизнес-процессов. Издательство МИФИ, 2020.
18. Сухов, В. Д. Экономическое обоснование разработки программного продукта [Электронный ресурс] : учебно – методическое пособие для бакалавров / В. Д. Сухов, А. А. Киселев, А. И. Сазонов. – Москва : Ай Пи Ар Медиа, 2022.
19. Тельнов Ю. Ф. Инжиниринг предприятия и управление бизнес – процессами [Электронный ресурс] : методология и технология : учеб. пособие для студентов магистратуры, обучающихся по направлению «Прикладная информатика» / Ю. Ф. Тельнов, И. Г. Фёдоров. – Москва : ЮНИТИ – ДАНА, 2017
20. Хаггард, Д. Поток данных и проектирование информационных систем. Учебное пособие. Москва: Инфра-М, 2021.

21. Шуваев, А. В. Программная инженерия : учеб. пособие для магистрантов направления подготовки 09.04.02 Информационные системы и технологии / А. В. Шуваев. – Ставрополь : Ветеран, 2020. – 84 с.
22. Язык React.js [Электронный ресурс] / URL: <https://react.dev/learn> (дата обращения: 15.10.2024).
23. SHA-2 – Wikipedia: Secure Hash Algorithm 2 [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/SHA-2> (дата обращения: 22.05.2025).
24. U-Boot – Wikipedia: Das U-Boot [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Das_U-Boot (дата обращения: 22.05.2025).
25. UML – Wikipedia: Unified Modeling Language [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/UML> (дата обращения: 22.05.2025).
26. Visual Studio Code – Code Editing [Электронный ресурс]. URL: <https://code.visualstudio.com/> (дата обращения: 22.05.2025).
27. yaml-cpp – YAML parser and emitter in C++ [Электронный ресурс]. URL: <https://github.com/jbeder/yaml-cpp> (дата обращения: 22.05.2025).