

РЕФЕРАТ

Выпускная квалификационная работа: 54 страницы, 3 раздела, 25 рисунков, 2 таблицы, 26 источников.

Ключевые слова: прокси-сервер, мониторинг сети, проектирование информационных систем, информационная безопасность.

Целью работы является разработка системы мониторинга прокси-серверов для ПАО «Совкомбанк».

Объект исследования – IT-инфраструктура ПАО «Совкомбанк», включающая в себя прокси-серверы, обеспечивающие управление сетевым трафиком и защиту данных.

Предмет исследования – процессы мониторинга и контроля состояния прокси-серверов.

Методы решения поставленных задач: системный анализ, модельное описание систем, прототипирование и быстрая разработка приложений, технико-экономический анализ.

Результатом работы является создание информационной системы, которая позволит автоматизировать процессы контроля за состоянием прокси-серверов в банке, своевременно выявлять проблемы и оперативно реагировать на возникающие сбои.

Практическая значимость работы: автоматизация мониторинга прокси-серверов сокращает время на выявление и устранение проблем, повышает общую эффективность работы IT-инфраструктуры банка.

Разработанная система прошла опытное внедрение в эксплуатацию в организации.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Аналитическая часть	7
1.1 Техничко-экономическая характеристика предметной области	7
1.2 Анализ функционирования объекта исследования	9
1.3 Определение цели и задач проектирования информационной системы	12
1.4 Обзор и анализ существующих разработок, выбор технологии проектирования	15
1.5 Выбор и обоснование проектных решений.....	18
2 Проектная часть.....	20
2.1 Разработка функционального обеспечения	20
2.2 Разработка информационного обеспечения.....	24
2.2.1 Характеристика результатной информации.....	24
2.2.2 Информационная модель и ее описание	26
2.3 Разработка программного обеспечения	30
2.3.1 Описание программных модулей и функций.....	30
2.3.2 Реализация базы данных	37
2.3.3 Компоненты пользовательского интерфейса	38
3 Оценка эффективности внедрения информационной системы	45
3.1 Общие положения	45
3.2 Расчёт экономической эффективности	47
ЗАКЛЮЧЕНИЕ	50
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	52

ВВЕДЕНИЕ

В условиях стремительного развития цифровых технологий и увеличения объемов сетевого взаимодействия, обеспечение надежности и безопасности IT-инфраструктуры становится одной из ключевых задач для организаций, особенно в банковской сфере. Прокси-серверы играют важную роль в управлении сетевым трафиком, защите данных и обеспечении конфиденциальности. Однако их эффективное функционирование требует постоянного мониторинга и контроля. Сбои в работе прокси-серверов могут привести к утечке данных, снижению качества обслуживания клиентов и нарушению бизнес-процессов.

Для ПАО «Совкомбанк», одного из крупнейших универсальных коммерческих банков России, внедрение системы мониторинга прокси-серверов является актуальной задачей. Это позволит повысить надежность IT-инфраструктуры, минимизировать риски, связанные с отказами оборудования, и обеспечить высокий уровень безопасности данных. Таким образом, тема разработки системы мониторинга прокси-серверов для ПАО «Совкомбанк» является актуальной и востребованной в современных условиях.

Стоит отметить, что системы мониторинга прокси-серверов для банка имеет и экономическую целесообразность, которая заключается в снижении рисков финансовых потерь как от простоев, так и от разных сбоев сетевого оборудования. Автоматизация мониторинга позволяет оптимизировать расходы, сокращая затраты на устаревшую систему, тем самым повышая эффективность работы IT-отдела банка. Это также способствует повышению удовлетворённости клиентов за счёт улучшения надёжности и скорости банковских операций.

Объектом исследования является IT-инфраструктура ПАО «Совкомбанк», включающая в себя прокси-серверы, обеспечивающие управление сетевым трафиком и защиту данных.

Предметом исследования выступают процессы мониторинга и контроля состояния прокси-серверов, а также методы автоматизации этих процессов.

Целью работы является разработка системы мониторинга прокси-серверов для ПАО «Совкомбанк», которая позволит автоматизировать процессы контроля за состоянием прокси-серверов, своевременно выявлять проблемы и оперативно реагировать на возникающие сбои.

Объектом автоматизации является процесс мониторинга прокси-серверов, включающий проверку их состояния, подсчет уникальных IP-адресов, измерение времени ротации IP и отправку уведомлений о сбоях.

Предметом автоматизации выступают программные и аппаратные средства, используемые для реализации системы мониторинга.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ текущей IT-инфраструктуры ПАО «Совкомбанк», выявить недостатки в информационных процессах;
- определить требования к системе мониторинга прокси-серверов;
- разработать архитектуру создаваемой системы;
- реализовать программное обеспечение системы;
- провести опытное внедрение системы мониторинга в эксплуатацию;
- оценить экономическую эффективность от внедрения системы.

В качестве исходных данных в работе используются технические характеристики IT-инфраструктуры ПАО «Совкомбанк», включая параметры прокси-серверов и сетевого оборудования. Для разработки системы использовался язык программирования Python, фреймворки для работы с сетевыми протоколами, а также инструмент для интеграции с мессенджером Telegram Bot API. Методы выполнения проекта включают анализ требований, проектирование архитектуры системы, программирование, тестирование и опытное внедрение.

Реализация системы мониторинга прокси-серверов позволит ПАО «Совкомбанк» повысить надежность и безопасность своей IT-

инфраструктуры. Система обеспечит автоматизацию процессов контроля за состоянием прокси-серверов, что сократит время на выявление и устранение сбоев. Это, в свою очередь, улучшит качество обслуживания клиентов, минимизирует риски утечки данных и повысит общую эффективность работы банка. Полученные результаты могут быть использованы для дальнейшего развития IT-инфраструктуры и внедрения аналогичных решений в других организациях.

1 Аналитическая часть

1.1 Технико-экономическая характеристика предметной области

В качестве предметной области проекта выступает ПАО «Совкомбанк», который является одним из ведущих универсальных коммерческих банков в России. ПАО «Совкомбанк» предоставляет широкий спектр финансовых услуг, охватывающих как физических, так и юридических лиц. К числу предлагаемых услуг относятся кредитование, депозитные продукты, расчетно-кассовое обслуживание, а также различные банковские услуги, которые помогают клиентам эффективно управлять своими финансами.

Основной целью «Совкомбанка» является обеспечение высококачественного обслуживания клиентов, что достигается за счет внедрения современных технологий и инновационных решений в области финансов. Банк активно развивает свои финансовые технологии, стремясь соответствовать современным требованиям рынка и удовлетворять потребности своих клиентов. Важным аспектом работы банка является его участие в различных социальных и экономических инициативах, направленных на устойчивое развитие и повышение финансовой грамотности населения. Это включает в себя программы по обучению клиентов основам финансовой грамотности, а также поддержку местных сообществ и инициатив.

Организационно-штатная структура ПАО «Совкомбанк» включает в себя различные подразделения, каждое из которых отвечает за определенные функции и задачи. Это позволяет банку эффективно управлять своими ресурсами и обеспечивать высокую степень координации между различными отделами. Структура включает в себя как фронт-офисные, так и бэк-офисные

подразделения, что способствует оптимизации процессов обслуживания клиентов и повышению общей эффективности работы банка.

Важным элементом стратегии «Совкомбанка» является постоянное совершенствование услуг и продуктов, предлагаемых клиентам. Банк активно внедряет новые финансовые инструменты, такие как карты рассрочки, ипотечные кредиты и программы по автокредитованию, что позволяет ему оставаться конкурентоспособным на рынке. Кроме того, «Совкомбанк» уделяет особое внимание цифровизации своих услуг, предлагая клиентам удобные мобильные приложения и онлайн-сервисы для управления своими финансами.

Таким образом, ПАО «Совкомбанк» представляет собой динамично развивающуюся финансовую организацию, которая стремится к высокому качеству обслуживания и внедрению инновационных решений, что делает его важным игроком на российском банковском рынке. Организационно-штатная структура банка, представленная на рисунке 1.1, демонстрирует его стремление к эффективному управлению и оптимизации бизнес-процессов, что в свою очередь способствует достижению поставленных целей и задач.

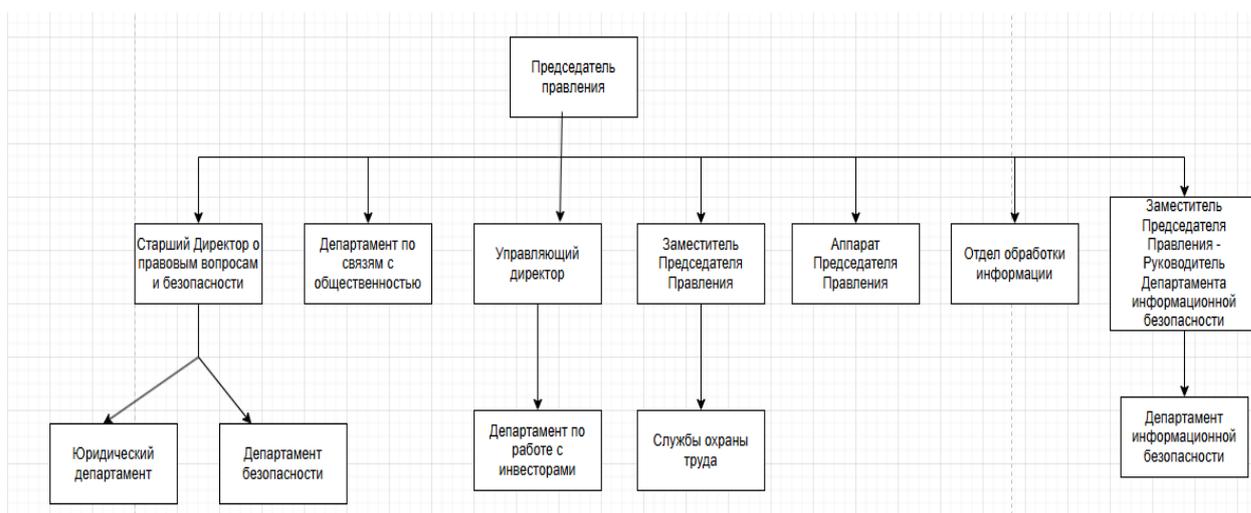


Рисунок 1.1 – Организационно-штатная структура предприятия ПАО «Совкомбанк»

1.2 Анализ функционирования объекта исследования

Для анализа деятельности ПАО «Совкомбанк» использовалась методология моделирования бизнес-процессов IDEF0.

Методология функционального моделирования IDEF – это технология описания системы в целом, как множества взаимозависимых действий или функций. Важно отметить функциональную направленность: IDEF0-функции системы исследуются независимо от объектов, которые обеспечивают их выполнение [1].

«Функциональная» точка зрения позволяет четко отделить аспекты назначения системы от аспектов ее физической реализации. В соответствии с методологией IDEF0 процесс представляется в виде функционального блока, который преобразует входы (входящие стрелки слева) в выходы (выходящие стрелки справа) при наличии необходимых механизмов (ресурсов) в управляемых условиях [1]. Механизмы (ресурсы) описываются входящими стрелками снизу, а управление – входящими стрелками сверху. Взаимосвязи и взаимодействия процессов в IDEF0 представляются дугами, соединяющими выходы одних функциональных блоков со входами других [1].

Первый шаг при построении модели IDEF0 заключается в определении назначения модели – набора вопросов, на которые должна ответить модель. Границы моделирования предназначены для обозначения ширины охвата предметной области и глубины детализации и являются логическим продолжением уже определенного назначения модели. Следующим шагом указывается предполагаемая целевая аудитория, для нужд которой создается модель. Зачастую от этого зависит уровень детализации, с которым должна создаваться модель.

Перед построением модели необходимо иметь представление о том, какие сведения о предмете моделирования уже известны, какие дополнительные материалы и/или техническая документация для понимания модели могут быть необходимы для целевой аудитории, какие языки и стиль

изложения являются наиболее подходящими. Под точкой зрения понимается перспектива, с которой наблюдалась система при построении модели. Точка зрения выбирается таким образом, чтобы учесть уже обозначенные границы моделирования и назначение модели.

Однажды выбранная точка зрения остается неизменной для всех элементов модели. При необходимости могут быть созданы другие модели, отображающей систему с других точек зрения [7].

Одним из доступных инструментов, предназначенных для реализации описанной методологии, является программа Figma – это редактор для верстки графических изображений. Дизайнеры и digital-специалисты используют его для отрисовки элементов интерфейса, создания интерактивных макетов, прототипов и векторных файлов.

На рисунке 1.2 представлена диаграмма в нотации IDEF0 «Система мониторинга прокси-серверов» в представлении «Как есть». Цель построения модели – предпроектный анализ предметной области. Точка зрения – системный аналитик.

Входная информация:

1. Логи работы прокси-серверов.
2. Конфигурационные данные (параметры сети, настройки безопасности, списки разрешенных и запрещенных адресов).

Управление:

1. Внутренние политики безопасности и регламенты Совкомбанка.
2. Требования законодательства, включая нормативы Центробанка РФ по информационной безопасности.

Механизмы:

1. IT-персонал (администраторы сетей, специалисты по информационной безопасности).
2. Программное обеспечение для мониторинга и анализа прокси-трафика.
3. Серверы и сетевое оборудование.

Выходная информация:

1. Отчёты о работе прокси-серверов:

- доступ к ресурсам
- время отклика.

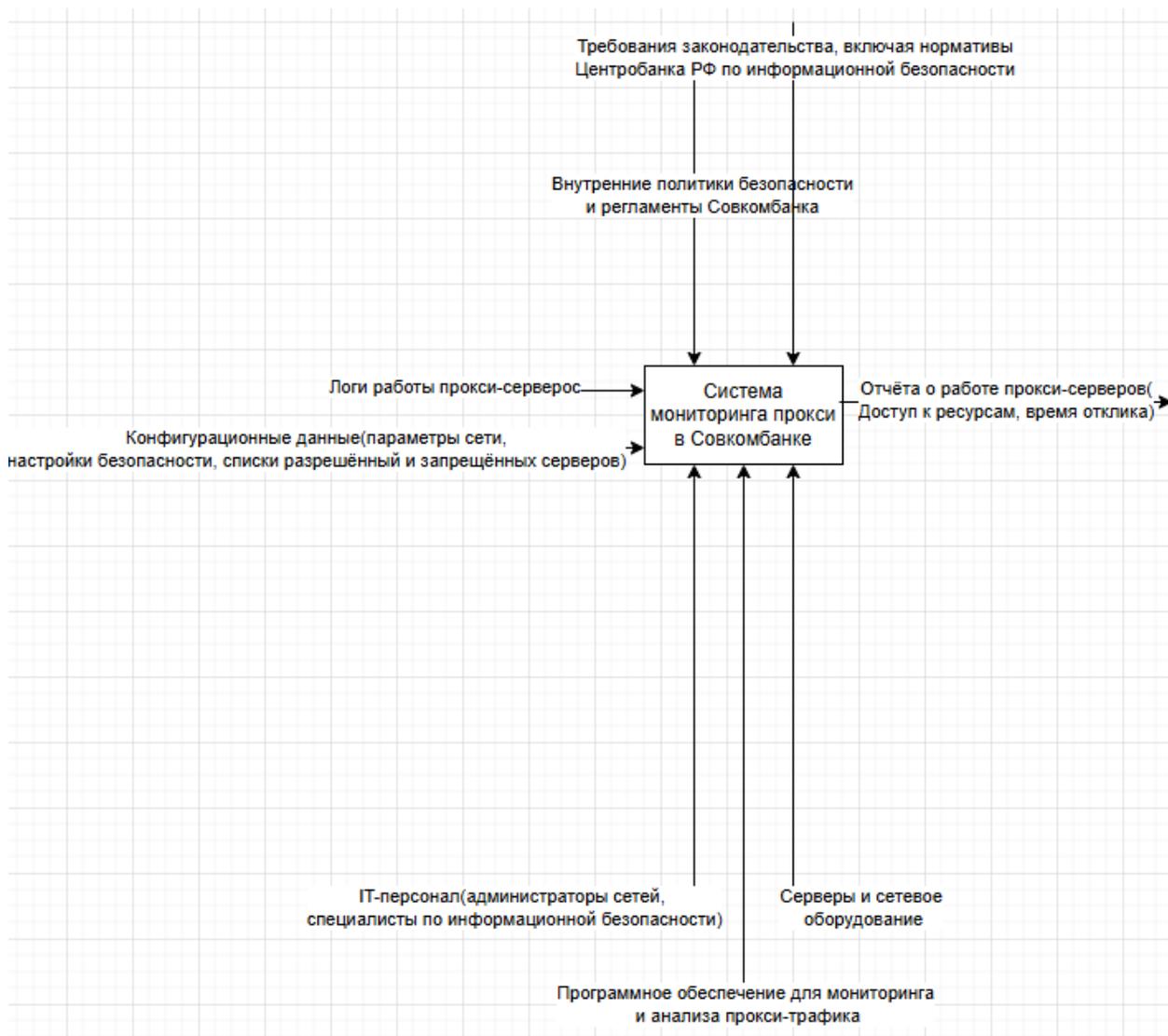


Рисунок 1.2 – Контекстная диаграмма IDEF0
«Система мониторинга прокси-серверов» «Как есть»

Декомпозиция в информатике, также известная как факторинг, разбивает сложную проблему или систему на части, которые легче понять, запрограммировать и поддерживать. Диаграмма декомпозиции позволяет обнаружить дублирование потоков и функций, отсутствие управления или

выходов у функций [8]. Декомпозиция процесса мониторинга прокси-серверов в ПАО «Совкомбанк» представлена на рисунке 1.3.

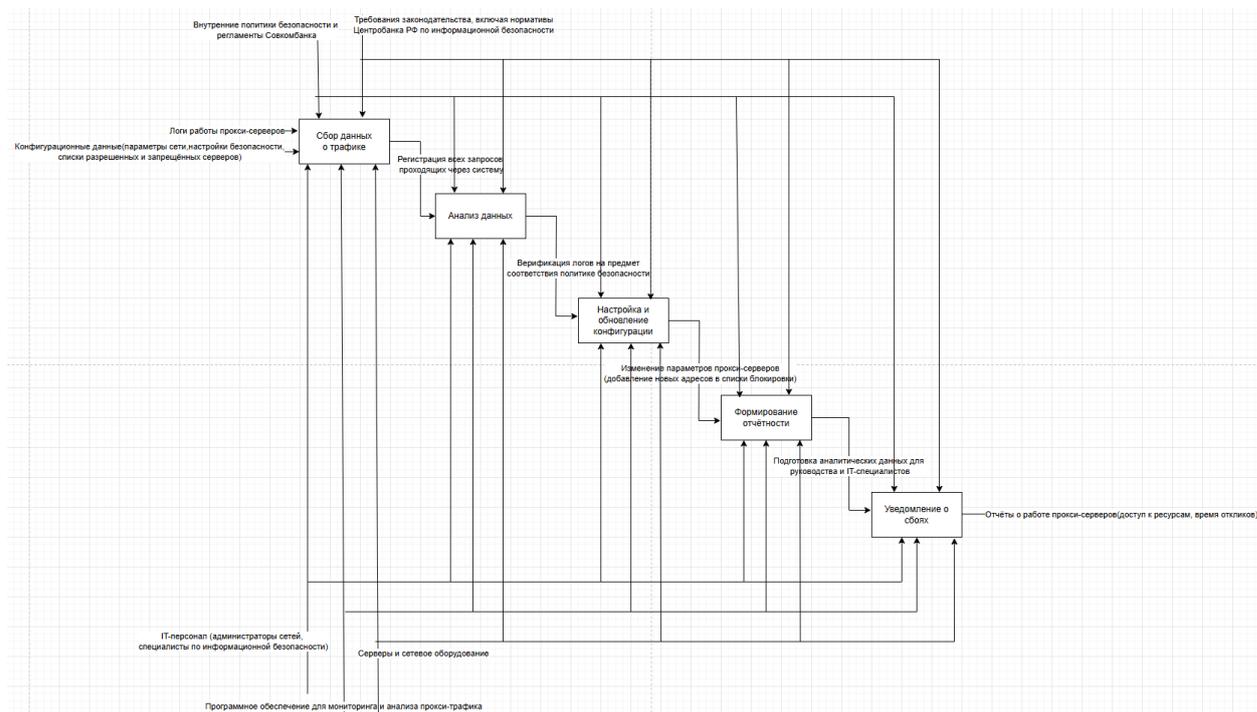


Рисунок 1.3 – Декомпозиция процесса мониторинга прокси-серверов в ПАО «Совкомбанк»

Проанализировав полученные модели предприятия в представлении «как есть», можно прийти к выводу, что существующая система мониторинга прокси-серверов не выполняет такие важные задачи, как уведомление о сбоях, не предоставляет данные для настройки системы и не даёт такую важную информацию: как время отклика и количество уникальных IP адресов для каждого прокси.

1.3 Определение цели и задач проектирования информационной системы

Цели создания системы и её задачи были сформулированы на основании технического задания, полученного от заказчика (ПАО «Совкомбанк»), которое имело следующее содержание:

1 Описание проекта.

Необходимо разработать веб-приложение для мониторинга мобильных прокси. Приложение должно проверять работоспособность прокси-серверов, отслеживать ротацию IP и отправлять уведомления в Telegram при сбоях.

Для получения IP адреса использовать наш чекер.

Прокси брать из таблицы «Прокси лист», где используются 25-02-09.

2 Функциональные требования.

2.1 Таблица с прокси (Frontend).

На странице «proxies» должна быть таблица с данными:

- прокси (ip:port);
- статус (работает / не работает);
- ротация (минимальное, среднее и максимальное время обновления IP);
- пул IP-адресов (количество доступных IP).

2.2 Админ-панель.

В стандартном кабинете администратора Django должна быть возможность:

- добавлять новые прокси;
- отключать старые прокси.

2.3 Проверка работоспособности прокси-серверов.

Приложение должно в бесконечном цикле проверять каждый прокси-сервер через «requests» и определять его статус в соответствии со следующим правилом:

- если прокси отвечает и получен именно IP адрес, он "работает";
- если прокси недоступен (timeout, ошибка 403/500 и т.д.), он "не работает".

2.4 Отслеживание ротации IP.

- для каждого прокси фиксировать время между изменениями IP.
- рассчитывать минимальное, среднее и максимальное время ротации.

2.5 Отслеживание пула IP адресов.

- каждый запрос на проверку должен записываться в БД: прокси, полученный IP, время отправки запроса, время получения ответа, ошибка (если IP не получен), статус (ок или ошибка)

2.6 Уведомления в Telegram.

Если прокси перестал работать, отправлять сообщение в Telegram через `pyTelegramBotApi`:

"⚠ Прокси [IP:PORT] перестал работать:[Причина]!".

Если прокси снова начал работать:

"✅ Прокси [IP:PORT] снова работает!".

3 Технологии

Используемые технологии:

- Backend: Django, Python 3.10
- База данных: PostgreSQL
- API: requests
- Уведомления: `pyTelegramBotApi`
- Контейнеризация: Docker, Docker Compose

Таким образом, была сформулирована основная цель проекта – разработка информационной системы мониторинга прокси-серверов, которая обеспечит:

- автоматизированный сбор и анализ данных о состоянии прокси-серверов;
- генерацию подробных отчетов о работе прокси;
- оперативное уведомление об инцидентах и неисправностях;

Для достижения поставленной цели разрабатываемая система должна решать следующие задачи:

1. Сбор и анализ данных о прокси:

- разработка механизма автоматического получения информации о прокси-серверах;
- создание алгоритмов подсчета уникальных IP-адресов;

- измерение времени ротации IP;
 - оценка производительности прокси-серверов.
2. Система уведомлений:
- реализация механизма мгновенного оповещения об неисправностях;
 - интеграция с мессенджерами (Telegram);
 - формирование детальных уведомлений о статусе прокси.
3. Генерация отчетности:
- разработка функционала для создания комплексных отчетов;
 - визуализация данных о работе прокси;
 - формирование статистических и аналитических материалов.
4. Интеграция и масштабируемость:
- обеспечение совместимости с существующей IT-инфраструктурой;
 - создание гибкой архитектуры для будущего развития системы.

1.4 Обзор и анализ существующих разработок, выбор технологии проектирования

Рассмотрим популярное программное обеспечение, используемое для мониторинга прокси-серверов.

Squid – программный пакет, реализующий функцию кэширующего прокси-сервера для протоколов HTTP, FTP, Gopher и (в случае соответствующих настроек) HTTPS. Разработан сообществом как программа с открытым исходным кодом (распространяется в соответствии с GNU GPL). Все запросы выполняет как один неблокируемый процесс ввода-вывода.

Используется в UNIX-подобных системах и в ОС семейства Windows NT. Имеет возможность взаимодействия с Active Directory Windows Server путём аутентификации через LDAP, что позволяет использовать разграничения доступа к интернет ресурсам пользователей, которые имеют

учётные записи на Windows Server, также позволяет организовать «нарезку» интернет-трафика для различных пользователей.

Используется вместе с движками Mediawiki на wiki-хостингах. Использование кэширующего прокси-сервера для сайтов становится выгодно примерно с 2000 посетителей в сутки. Имеет следующие преимущества:

- высокая производительность кэширования;
- поддержка множества протоколов;
- гибкие настройки фильтрации.

CCProxy – это программа для настройки и управления прокси-серверами. Инструмент предоставляет широкие возможности, включая кэширование веб-страниц, фильтрацию контента и мониторинг сетевой активности. CCProxy поддерживает протоколы HTTP, HTTPS, FTP, SOCKS, и даже Telnet, что делает его безопасным и универсальным программным обеспечением для получения высокого уровня анонимности. CCProxy предоставляет множество дополнительных настроек для улучшения производительности и безопасности. Основные функции программы:

- мониторинг и журналирование: позволяет отслеживать активность на вашем прокси-сервере;
- ограничение доступа: можно настроить ограничения доступа к прокси-серверу на основе IP-адресов или пользователей;
- кэширование: CCProxy может кэшировать веб-страницы, ускоряя загрузку и снижая нагрузку на сеть.

Ключевые возможности программы:

- кэширование веб-страниц;
- фильтрация контента;
- мониторинг сетевой активности;
- управление пропускной способностью;
- протоколирование трафика.

Несмотря на функциональность рассмотренных систем, они имеют ряд ограничений:

- отсутствие комплексного подхода к мониторингу, что затрудняет получение полной картины о состоянии прокси-серверов;
- ограниченные возможности уведомлений, что может привести к задержкам в реагировании на инциденты;
- сложность кастомизации под специфические требования банка, что ограничивает гибкость решений;
- недостаточный уровень детализации отчетности, что затрудняет анализ и принятие решений на основе данных.

При выборе технологии проектирования системы мониторинга прокси-серверов для ПАО «Совкомбанк» необходимо учитывать:

1. Масштабируемость решения, чтобы система могла адаптироваться к растущим объемам данных и увеличению числа прокси-серверов.
2. Возможности интеграции с существующими системами, чтобы обеспечить совместимость и упрощение процессов.
3. Производительность, чтобы система могла обрабатывать большие объемы данных в реальном времени.
4. Безопасность, чтобы защитить данные и предотвратить несанкционированный доступ.
5. Удобство администрирования, чтобы минимизировать затраты на поддержку и обучение персонала.

Рекомендуемый технологический стек:

- язык программирования: Python, который обеспечивает высокую производительность и гибкость [9, 10];
- фреймворк: Django/Flask, позволяющий быстро разрабатывать веб-приложения [3, 15];
- база данных: PostgreSQL, обеспечивающая надежное хранение и обработку данных [5, 6];
- система уведомлений: Telegram Bot API, позволяющая оперативно информировать администраторов о проблемах;

– контейнеризация: Docker, что упрощает развертывание и масштабирование системы [2, 4, 11];

– мониторинг: Prometheus, Grafana для визуализации и анализа данных [13, 19, 20, 21].

Разрабатываемая информационная система мониторинга прокси будет:

– полностью кастомизированной под нужды ПАО «Совкомбанк»;

– обеспечивать высокий уровень безопасности;

– предоставлять детальную аналитику;

– иметь гибкую архитектуру для будущего развития

Такой подход позволит создать эффективный инструмент управления прокси-серверами, который значительно повысит надежность и производительность IT-инфраструктуры банка.

1.5 Выбор и обоснование проектных решений

Для работы информационной системы мониторинга прокси в ПАО "Совкомбанк" необходим сервер, обеспечивающий высокую производительность, надежность и безопасность, поскольку банк работает с большими объемами данных и требует максимальной защиты от внешних угроз. Минимальные характеристики аппаратной базы для сервера следующие:

1. Операционная система: Windows 10.
2. Жесткий диск: 250 ГБ SSD.
3. Центральный процессор: Intel Xeon E-2236.
4. Оперативная память: 16 ГБ.

Для разработки информационной системы мониторинга прокси-серверов (на примере ПАО «Совкомбанк») использовались:

1. PyCharm – интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода,

графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA [9, 10, 11].

2. pyTelegramBotAPI – это простая, но расширяемая библиотека на Python для работы с Telegram Bot API. Она позволяет разработчикам легко создавать и управлять ботами в Telegram, предоставляя удобный интерфейс для взаимодействия с API.

3. python-dotenv – это библиотека для Python, предназначенная для работы с переменными окружения, которая позволяет загружать конфигурационные параметры из файла «.env».

4. psycopg2 – это пакет для Python, который предоставляет адаптер для работы с базами данных PostgreSQL. Он является оберткой над библиотекой libpq, официальным клиентом PostgreSQL, и позволяет легко взаимодействовать с PostgreSQL из Python-приложений.

5. Gunicorn (Green Unicorn) – это WSGI HTTP-сервер для Python, предназначенный для развертывания и запуска веб-приложений в production-среде.

6. Docker Compose – это инструмент, предназначенный для определения и запуска многоконтейнерных приложений с использованием Docker. Он позволяет легко управлять несколькими контейнерами, которые составляют приложение, с помощью единого конфигурационного файла [15, 17].

2 Проектная часть

2.1 Разработка функционального обеспечения

Анализ функциональной модели «как есть» (AS-IS) представляет собой важный этап в процессе проектирования и внедрения информационных систем [1]. Он позволяет выявить слабые стороны текущих бизнес-процессов, определить их неэффективные элементы, а также оценить преимущества, которые могут быть достигнуты за счет внедрения новых процессов. Такой анализ помогает понять, какие изменения необходимо внести в существующую структуру организации для повышения её эффективности и соответствия современным требованиям. Модель AS-IS предоставляет полное представление о текущем состоянии бизнес-процессов, включая их структуру, последовательность операций, используемые ресурсы и взаимодействие между различными подразделениями. Однако в процессе анализа могут быть обнаружены недостатки, такие как избыточность операций, дублирование функций, низкая производительность или несоответствие требованиям безопасности. Эти недостатки становятся основой для разработки новой модели «как должно быть» (TO-BE), которая направлена на устранение выявленных проблем и оптимизацию процессов.

Модель TO-BE разрабатывается на основе анализа текущей модели AS-IS и представляет собой целевое состояние бизнес-процессов после их реинжиниринга. Она позволяет на этапе проектирования будущей информационной системы определить необходимые изменения, которые обеспечат повышение эффективности работы организации. Использование модели TO-BE способствует сокращению сроков автоматизации предметной области, так как она позволяет заранее выявить и устранить потенциальные проблемы. Кроме того, модель TO-BE помогает эффективно распределить ресурсы между операциями бизнес-процессов, что позволяет оценить их

использование после внедрения изменений. Основная задача модели ТО-ВЕ заключается в описании системы с учетом функционально-ориентированного подхода. Это включает в себя разработку мер по устранению негативного влияния выявленных проблемных факторов, таких как избыточные операции, недостаточная автоматизация или несоответствие требованиям безопасности [8].

При постановке задачи важно четко определить назначение программного обеспечения и сформулировать основные требования к нему. Каждое требование описывает необходимое или желаемое свойство программного обеспечения, включая его функциональность, производительность, надежность и безопасность. Например, для системы мониторинга прокси-серверов в ПАО «Совкомбанк» требования могут включать:

- генерацию отчетов о работе прокси, включая данные о времени отклика, объеме трафика и доступе к ресурсам;
- интеграцию с существующими системами безопасности и управления сетью;
- уведомления о сбоях и инцидентах в режиме реального времени.

Для выполнения структурно-функционального анализа предметной области используется функционально-ориентированное моделирование бизнес-процессов. Это предполагает создание диаграмм «как должно быть» в стандарте IDEF0, которые позволяют визуализировать целевую структуру процессов и их взаимодействие. Основной целью построения моделей ТО-ВЕ является автоматизация учета аппаратного обеспечения автоматизированной информационной системы (АИС). Это включает в себя мониторинг прокси-серверов, анализ их работы и управление настройками. В результате внедрения таких моделей специалисты получают больше времени для выполнения своих основных профессиональных обязанностей, так как рутинные задачи автоматизируются. Например, система мониторинга прокси может автоматически собирать данные о работе серверов, анализировать их и предоставлять отчеты, что значительно сокращает нагрузку на IT-персонал.

Контекстная диаграмма IDEF0 «Как должно быть» для системы мониторинга прокси в ПАО «Совкомбанк» представлена на рисунке 2.1. Она описывает основные входные данные, механизмы, управление и выходные данные системы.

Входная информация:

- логи работы прокси-серверов, которые содержат данные о запросах, времени отклика и объеме трафика;
- конфигурационные данные, включая параметры сети, настройки безопасности, списки разрешенных и запрещенных адресов.

Управление:

- внутренние политики безопасности и регламенты Совкомбанка, которые определяют правила работы системы;
- требования законодательства, включая нормативы Центробанка РФ по информационной безопасности, которые обеспечивают соответствие системы правовым нормам.

Механизмы:

- IT-персонал, включая администраторов сетей и специалистов по информационной безопасности, которые управляют системой и реагируют на инциденты;
- программное обеспечение для мониторинга и анализа прокси-трафика, которое автоматизирует сбор и обработку данных;
- серверы и сетевое оборудование, обеспечивающие работу системы.

Выходная информация:

- отчеты о работе прокси-серверов, включая данные о доступе к ресурсам, времени отклика и объемах трафика;
- данные для настройки системы и устранения инцидентов, которые помогают IT-персоналу оперативно реагировать на проблемы;
- уведомления о сбоях.

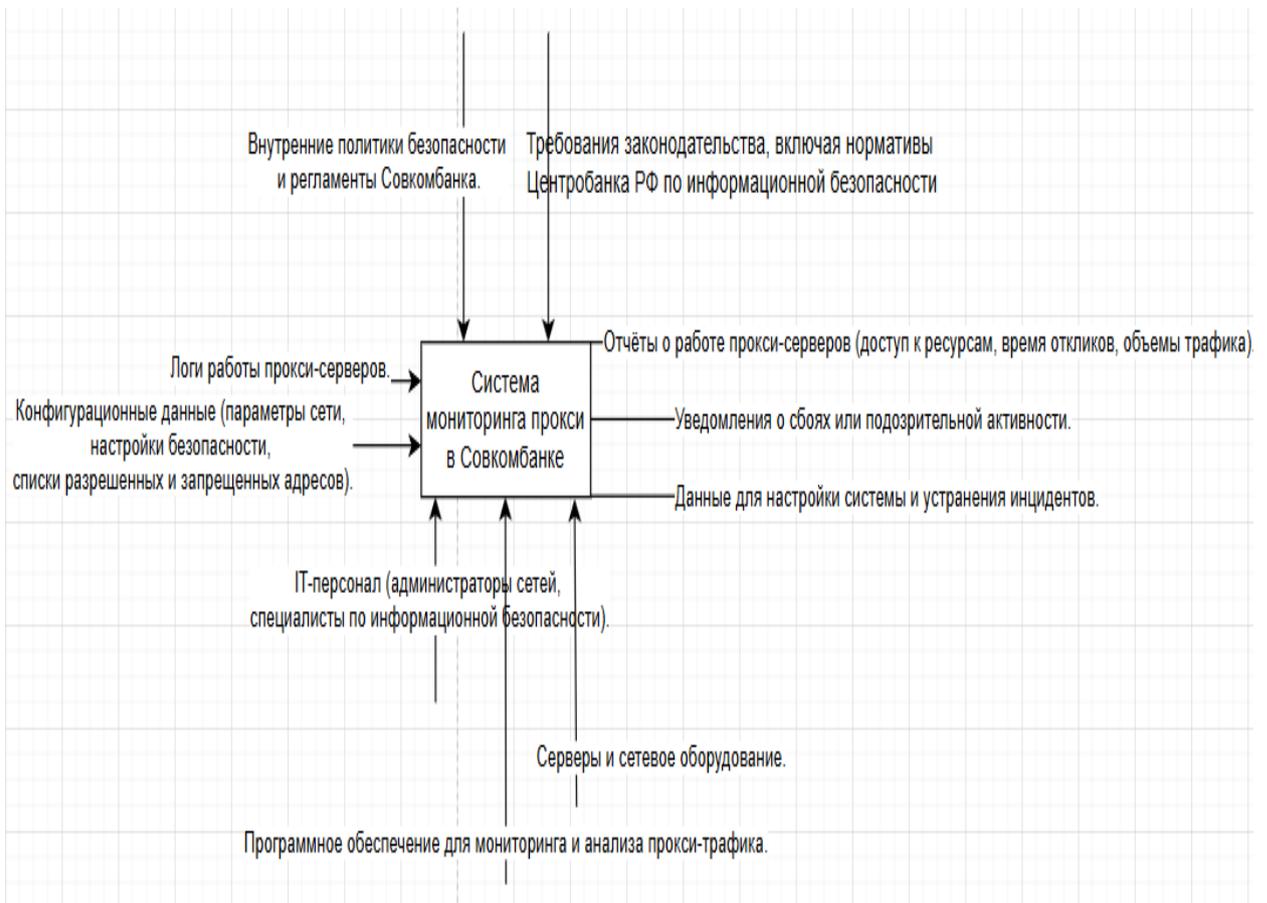


Рисунок 2.1 – Контекстная диаграмма IDEF0 «Процесс мониторинга прокси-серверов в ПАО «Совкомбанк» в представлении «как должно быть»

На рисунке 2.2 представлена контекстная диаграмма «Декомпозиция процесса мониторинга прокси в Совкомбанке». Она детализирует основные этапы работы системы, включая сбор данных, их анализ, генерацию отчетов и уведомлений. Такая декомпозиция позволяет лучше понять структуру системы и определить ключевые точки для её оптимизации. Внедрение системы мониторинга прокси на основе модели ТО-ВЕ позволит ПАО «Совкомбанк» повысить надежность и безопасность своей IT-инфраструктуры, а также улучшить качество обслуживания клиентов за счет автоматизации рутинных процессов и повышения эффективности работы сотрудников.

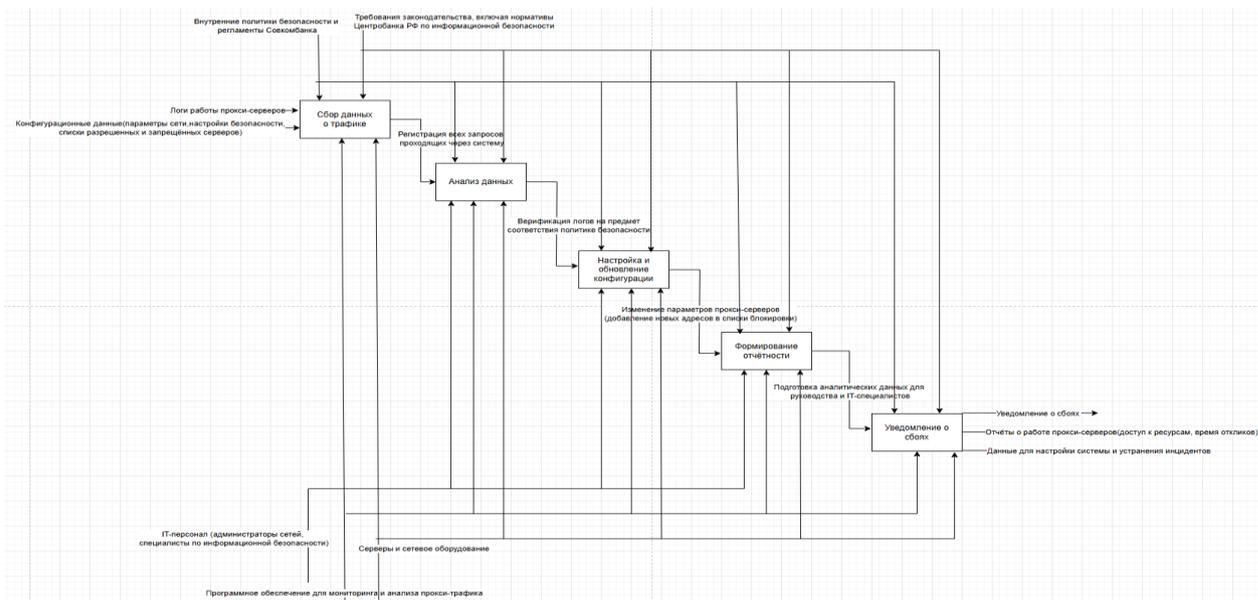


Рисунок 2.2 – Декомпозиция процесса мониторинга прокси-серверов в ПАО «Совкомбанк» в представлении «как должно быть»

2.2 Разработка информационного обеспечения

Информационное обеспечение представляет собой сложную и многоуровневую систему, которая включает в себя не только классификацию и кодирование информации, но и унифицированные системы документации, схемы информационных потоков, циркулирующих в банке, а также методологии построения баз данных. Все эти компоненты играют важную роль в эффективном управлении данными и процессами внутри организации.

2.2.1 Характеристика результатной информации

Результирующая информация, формируемая информационной системой мониторинга прокси, играет ключевую роль в обеспечении эффективного управления и анализа работы прокси-серверов. Эта информация включает в себя различные отчеты и метрики, которые позволяют администраторам и IT-специалистам принимать обоснованные решения на основе собранных данных.

Статус прокси: Информация о текущем состоянии каждого прокси-сервера (активен/неактивен), что позволяет быстро выявлять проблемы и реагировать на них.

Время работы: Данные о времени, в течение которого прокси-серверы были активны, что помогает оценить их надежность и производительность.

Количество уникальных IP-адресов: Подсчет уникальных IP-адресов, использованных через прокси, что позволяет анализировать нагрузку и выявлять возможные проблемы с перегрузкой.

Время ротации IP: Данные о частоте смены IP-адресов, что важно для обеспечения анонимности и безопасности при использовании прокси.

Записи о запросах: Логи всех запросов, проходящих через прокси, включая временные метки, статус ответов и объем переданных данных. Эти данные позволяют проводить детальный анализ трафика и выявлять аномалии.

Ошибки и сбои: Записи о возникших ошибках и сбоях, что помогает в диагностике проблем и улучшении работы системы.

Системные уведомления: Автоматические уведомления о неработающих прокси, которые отправляются администраторам через интеграцию с мессенджерами (например, Telegram). Это позволяет оперативно реагировать на проблемы.

История уведомлений: Хранение истории всех уведомлений, что позволяет анализировать частоту и причины сбоев.

Общие отчеты: Сводные отчеты, содержащие ключевые показатели эффективности (KPI) работы прокси-серверов, что позволяет оценивать их эффективность в целом.

Результирующая информация может быть представлена в различных форматах, включая:

Текстовые отчеты: Подробные текстовые отчеты, содержащие все необходимые данные о работе прокси.

Электронные таблицы: Форматы, удобные для дальнейшего анализа и обработки данных (например, CSV или Excel).

Визуальные панели: Интерактивные дашборды, которые позволяют в реальном времени отслеживать состояние прокси и ключевые метрики.

Структура результатной информации, формируемой системой мониторинга прокси, обеспечивает всесторонний анализ работы прокси-серверов и позволяет принимать обоснованные решения для оптимизации их работы. Эффективное использование этой информации способствует повышению надежности и безопасности ИТ-инфраструктуры, что является критически важным для успешной деятельности ПАО «Совкомбанк».

2.2.2 Информационная модель и ее описание

Одна из основных целей семантического моделирования состоит в том, чтобы результаты анализа предметной области были отражены в достаточно простом, наглядном, но в то же время формализованном и достаточно информативном виде. В этом смысле ER-диаграмма является очень удачным решением [5]. В ней сочетаются функциональный и информационный подходы, что позволяет представлять, как совокупность выполняемых функций, так и отношения между элементами системы, задаваемые структурами данных. При этом графическая форма позволяет отобразить в компактном виде типологию и свойства сущностей и связей за счет наглядных условных обозначений, а формализмы, положенные в основу ER-диаграмм, позволяют использовать на следующем шаге проектирования логической структуры базы данных строгий аппарат нормализации.

Сущности. Каждый тип сущности в ER-диаграммах представляется в виде прямоугольника, содержащего имя сущности. В качестве имени обычно используются существительные (или обороты существительного) в единственном числе.

Связь – графически изображаемая ассоциация, устанавливаемая между сущностями. Каждый тип связи на ER-диаграмме отображается в виде ромба с именем связи внутри. Как и в реляционных схемах баз данных, в ER-

диаграммах вводится понятие нормальных форм, причем их смысл очень близок смыслу реляционных нормальных форм.

Задачей следующей стадии проектирования системы базы данных является выбор подходящей СУБД и отображение в ее среду (структуру данных) спецификаций инфологической модели предметной области. Другими словами, модель предметной области разрабатываемой системы должна быть представлена в терминах модели данных концептуального уровня выбранной конкретной СУБД. Эту стадию называют логическим (или даталогическим) проектированием базы данных, а ее результатом является концептуальная схема базы данных, включающая определение всех информационных элементов (единиц) и связей, в том числе задание типов, характеристик и имен.

Существует много вариантов отображения инфологической модели предметной области в даталогическую модель базы. Здесь следует учитывать влияние двух следующих значимых факторов, связанных с практикой разработки базы данных [6].

Во-первых, связи предметной области могут отображаться двумя путями: как декларативным – в логической схеме, так и процедурным – отработкой связей через программные модули, обрабатывающие (связывающие) соответствующие хранимые данные.

Во-вторых, существенным фактором может оказаться характер обработки информации. Например, частые обращения к совместно обрабатываемым данным, очевидно, предполагают их совместное хранение, а данные (особенно большого объема), к которым обращаются редко, целесообразно хранить отдельно от часто используемых. Стадия физического проектирования базы данных в общем случае включает:

- выбор способа организации базы данных;
- разработку спецификации внутренней схемы средствами модели данных ее внутреннего уровня;
- описание отображения концептуальной схемы во внутреннюю.

Способ хранения базы данных определяется механизмами СУБД автоматически «по умолчанию» на основе спецификаций концептуальной схемы базы данных, и внутренняя схема в явном виде в таких системах не используется. Следует также отметить, что внешние схемы базы данных обычно конструируются на стадии разработки приложений.

Структура базы данных мониторинга прокси-серверов разрабатывалась на основе функциональных требований к системе, главное из которых – обеспечивать эффективное хранение и управление данными о прокси-серверах, их запросах и статистике. Разработанная логическая модель БД представлена на рисунке 2.3.

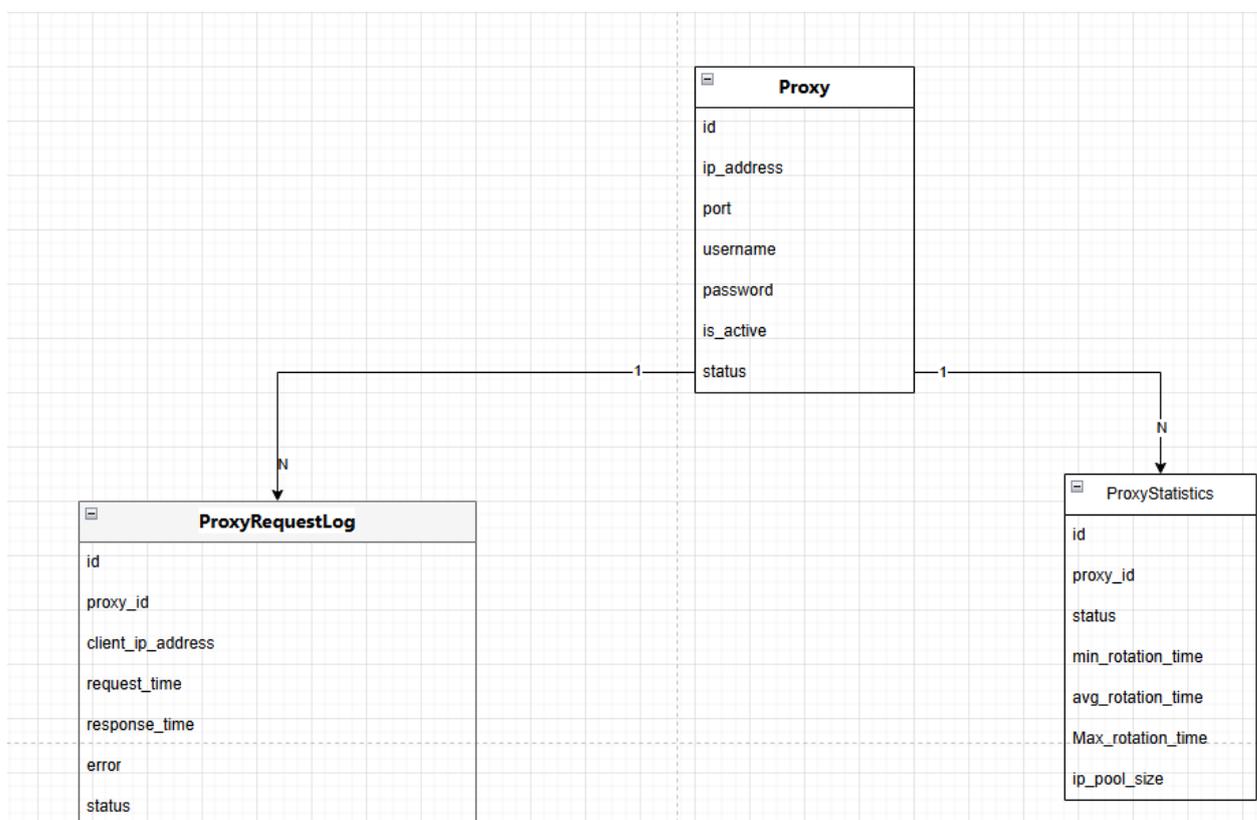


Рисунок 2.3 – Логическая модель БД «Система мониторинга прокси»

Накапливаемая и хранимая в БД информация должна позволить администраторам и ИТ-специалистам быстро получать необходимую информацию для анализа и принятия решений, что в свою очередь способствует повышению надежности и безопасности ИТ-инфраструктуры.

Таким образом, для создания БД проектируемой системы требуется три таблицы:

1. Таблица Proxy:

Хранит информацию о каждом прокси-сервере, включая его IP-адрес, порт, учетные данные для аутентификации и статус активности. Эта модель позволяет управлять прокси-серверами и отслеживать их текущее состояние.

2. Таблица ProxyRequestLog:

Логирует все запросы, проходящие через прокси-серверы. Каждая запись содержит информацию о времени запроса, времени ответа, статусе и возможных ошибках. Это позволяет проводить анализ трафика и выявлять проблемы в работе прокси.

3. Таблица ProxyStatistics:

Содержит статистику по каждому прокси, включая его статус, время ротации IP-адресов и размер пула доступных IP. Эта модель помогает в оценке производительности прокси и выявлении аномалий в их работе.

Физическое описание создаваемых таблиц представлено в таблице 2.1.

Таблица 2.1 – Описание таблиц Proxy, ProxyRequestLog и ProxyStatistics

Таблица	Поле	Тип данных	Описание
Proxy	id	Integer (Primary Key)	Уникальный идентификатор прокси-сервера
	ip_address	GenericIPAddressField	IP-адрес прокси-сервера
	port	Integer	Порт, используемый прокси-сервером
	username	CharField(max_length=255)	Имя пользователя для аутентификации (если требуется).
	password	CharField(max_length=255)	Пароль для аутентификации (если требуется).
	is_active	Boolean	Статус активности прокси-сервера (true - активен, false - неактивен).
	status	CharField(max_length=50)	Текущий статус прокси-сервера (например, "working", "not working", "unknown").

Продолжение таблицы 2.1

Таблица	Поле	Тип данных	Описание
ProxyRequestLog	id	Integer (Primary Key)	Уникальный идентификатор записи лога запроса
	proxy	ForeignKey(Proxy, on_delete=models.CASCADE)	Связь с моделью Proxy, указывающая, к какому прокси относится данный лог.
	ip_address	GenericIPAddressField	IP-адрес клиента, который сделал запрос через прокси.
	request_time	DateTimeField	Время, когда был сделан запрос.
	response_time	DateTimeField	Время, когда был получен ответ от прокси.
	error	TextField	Сообщение об ошибке, если таковая произошла во время запроса.
	status	CharField(max_length=10)	Статус ответа (например, "200", "404", "500").
ProxyStatistics	id	Integer (Primary Key)	Уникальный идентификатор статистики прокси.
	proxy	CharField(max_length=255, unique=True)	Прокси в формате "ip:port".
	min_rotation_time	FloatField	Минимальное время ротации IP-адресов (в секундах).
	avg_rotation_time	FloatField	Среднее время ротации IP-адресов (в секундах).
	max_rotation_time	FloatField	Максимальное время ротации IP-адресов (в секундах).
	ip_pool_size	Integer	Количество доступных IP-адресов в пуле.

2.3 Разработка программного обеспечения

2.3.1 Описание программных модулей и функций

Для разработки системы мониторинга прокси-серверами необходимо организовать проект таким образом, чтобы структура была логичной, удобной и масштабируемой для дальнейшей работы. Это особенно важно, так как

проект предполагает работу с большим объемом данных, обработку логов и выполнение множества операций, связанных с мониторингом и анализом прокси-серверов.

Программный модуль можно определить как независимую и функционально законченную часть программы, оформленную в виде самостоятельного фрагмента кода, упакованного в отдельный файл или обособленного другим способом. Эти модули формируют структуру программного продукта и позволяют избежать повторяющихся участков кода, что делает размер приложения меньше, а его работу – быстрее и более эффективной. Созданные программные модули в информационной системе мониторинга прокси-серверов включают в себя различные компоненты, отвечающие за обработку данных, коммуникацию с пользователями и интеграцию с существующими системами банка. Каждый модуль разработан с учетом принципов модульности и переиспользуемости, что позволяет легко адаптировать систему под изменяющиеся требования и добавлять новые функции без необходимости переработки всего программного продукта. Схема взаимосвязи программных модулей представлена на рисунке 2.4.

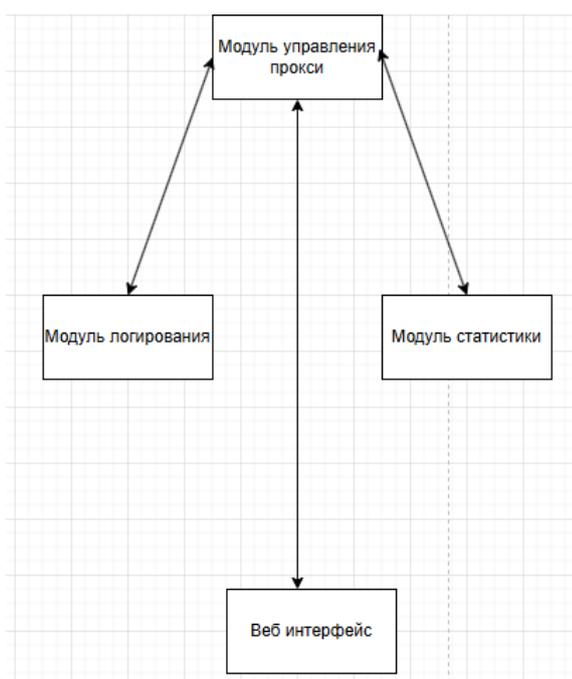


Рисунок 2.4 – Взаимосвязь программных модулей

Внутри основного проекта создается область `proxies`, которая будет содержать все файлы и модули, связанные с мониторингом прокси. Такая структура позволяет четко разделить функциональные компоненты системы, что упрощает её поддержку и развитие. В области `proxies` находятся модели прокси и модуль уведомления в телеграмме (рисунок 2.5).

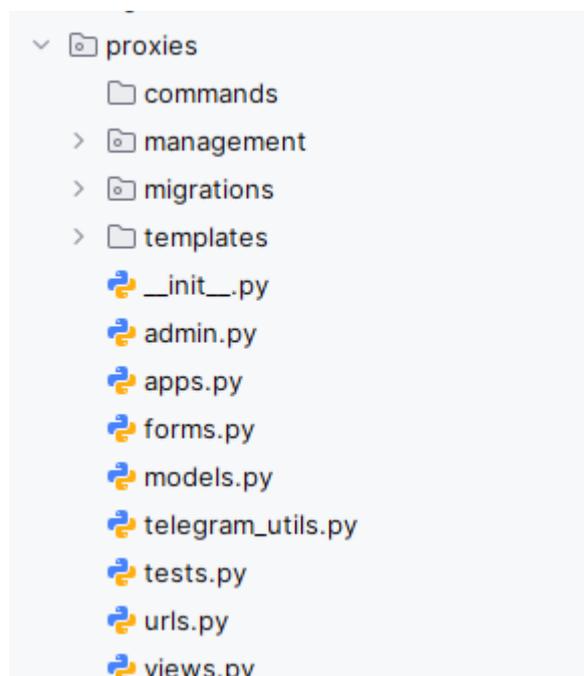


Рисунок 2.5 – Программный модуль «proxies»

Модуль `proxy_monitor` создан для хранения данных о статусе прокси-сервера после их получения (рисунок 2.6).

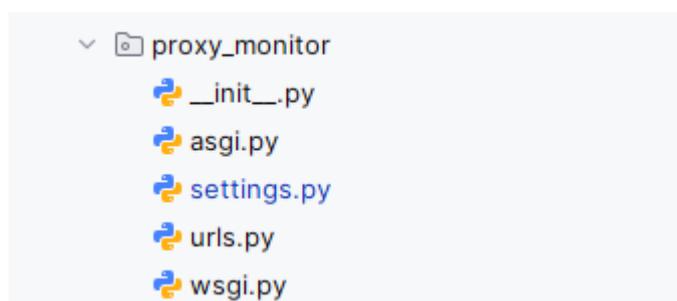


Рисунок 2.6 – Программный модуль «proxy_monitor»

В серверном обеспечении информационной системы мониторинга прокси-серверов были созданы два основных модуля, с которыми напрямую взаимодействует клиентская часть разработанного приложения. Эти модули

обеспечивают функциональность системы и позволяют эффективно управлять прокси-серверами:

1. Модуль управления прокси.

Первым и важнейшим элементом серверного обеспечения является модуль управления прокси, который отвечает за аутентификацию, авторизацию и управление состоянием прокси-серверов. Этот модуль реализован на основе фреймворка Django, который предоставляет мощные инструменты для разработки веб-приложений и управления базами данных.

Модуль управления прокси выполняет следующие функции:

Аутентификация и авторизация: обеспечивает проверку учетных данных пользователей и их прав доступа к различным функциям системы.

Управление прокси: позволяет добавлять, удалять и изменять параметры прокси-серверов, а также отслеживать их статус и производительность.

Мониторинг состояния: регулярно проверяет активность прокси-серверов и собирает статистику о их работе, включая время ротации IP-адресов и количество уникальных IP.

Модуль управления прокси взаимодействует с базой данных, хранящей информацию о прокси-серверах, их запросах и статистике. Это позволяет системе предоставлять актуальные данные для анализа и принятия решений.

2. Веб-сервер.

Другим ключевым элементом серверной архитектуры является веб-сервер, который предоставляет доступ к информации о состоянии прокси и управляет пользовательскими запросами. Веб-сервер разработан на языке PHP, который является одним из самых популярных языков программирования для создания динамических веб-приложений.

Функции веб-сервера включают:

Предоставление интерфейса: обеспечивает пользователям доступ к веб-интерфейсу системы, где они могут просматривать информацию о прокси, их статусе и статистике.

Создание новых пользователей: позволяет администраторам системы добавлять новых пользователей и управлять их правами доступа.

Отчетность: генерирует отчеты о работе прокси-серверов, включая данные о производительности и активности.

Веб-сервер взаимодействует с модулем управления прокси, обеспечивая передачу данных между клиентской частью приложения и серверной частью. Это позволяет пользователям получать актуальную информацию о состоянии прокси и управлять ими в режиме реального времени.

Таким образом, серверные программные модули, включающие модуль управления прокси и веб-сервер, обеспечивают эффективное функционирование информационной системы мониторинга прокси. Эти модули взаимодействуют друг с другом, предоставляя пользователям необходимые инструменты для управления и анализа работы прокси-серверов, что способствует повышению надежности и безопасности ИТ-инфраструктуры.

Внутри области proxies, как показано на рисунке 2.7, создается папка commands, где будут размещены команды для выполнения различных операций. Эти команды включают проверку прокси, подсчет статистики, измерение времени ротации IP-адресов и отправку уведомлений.

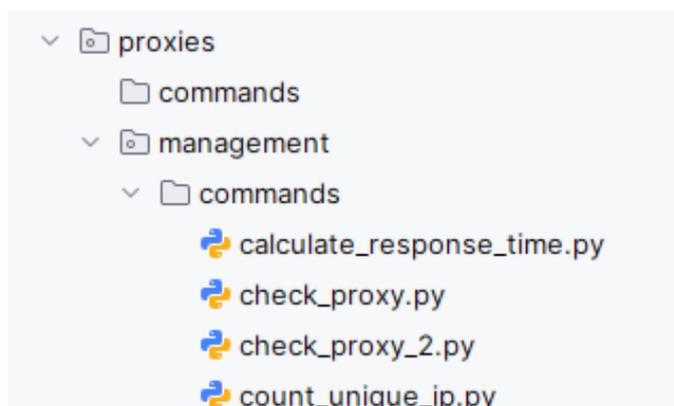


Рисунок 2.7 – Структура commands в папке proxies

Такой подход к организации проекта позволяет структурировать код и избежать избыточности, что особенно важно для сложных систем.

Одной из ключевых команд является команда `check_proxy.py`, которая выполняет следующие функции:

- получение списка прокси-серверов и их параметров;
- подсчет уникальных IP-адресов для каждого прокси;
- измерение времени ротации IP-адресов;
- проверка статуса активности прокси за последние 3 минуты;
- отправка уведомлений в Telegram-бот о неактивных прокси или восстановлении их работы.

Программный код функции получения данных прокси-сервера, которая является основой для работы команды, имеет вид:

```
def get_proxy_ip(proxy): 1 usage ± Денис
    """Получает IP-адрес через прокси."""
    response = requests.get(url='http://[REDACTED]', proxies={'http': proxy, 'https': proxy}, timeout=10)
    response.raise_for_status()
    return response.text
```

Она обеспечивает доступ к данным о прокси-серверах, включая их IP-адреса, порты и текущий статус. Далее была разработана функция подсчета уникальных IP-адресов для каждого прокси, программный код которой имеет следующий вид:

```
def count_unique_ips(proxy_id): 1 usage ± Денис
    """Подсчитывает количество уникальных IP для данного прокси."""
    return ProxyRequestLog.objects.filter(proxy_id=proxy_id, ip_address__isnull=False).values('ip_address').distinct().count()
```

Эта функция позволяет анализировать использование прокси и выявлять потенциальные проблемы, связанные с их перегрузкой или неправильной настройкой.

Следующим этапом являлась разработка функции измерения времени ротации IP-адресов. Эта функция позволяет отслеживать, как часто меняются IP-адреса у каждого прокси-сервера, что важно для обеспечения их стабильной работы. Программный код этой функции выглядит следующим образом:

```
def measure_ip_rotation_time(proxy): 1 usage ± Денис
    """Измеряет время ротации IP для данного прокси."""
    logs = ProxyRequestLog.objects.filter(proxy=proxy, ip_address__isnull=False).order_by('request_time')
    if not logs.exists():
        print("Нет доступных логов для указанного прокси.")
        return None, None, None
    change_times = []
    previous_ip, previous_time = None, None
    for log in logs:
        if previous_ip is not None and log.ip_address != previous_ip:
            change_times.append(log.request_time - previous_time)
            previous_ip, previous_time = log.ip_address, log.request_time
    if change_times:
        min_change_time = min(change_times).total_seconds()
        avg_change_time = sum(change_times, timedelta()).total_seconds() / len(change_times)
        max_change_time = max(change_times).total_seconds()
        return min_change_time, avg_change_time, max_change_time
    print("Нет изменений IP для указанного прокси.")
    return None, None, None
```

Завершающим этапом являлось создание функции проверки прокси-серверов и отправки уведомлений. Программный код этой функции имеет следующий вид:

```
def check_proxy_status(proxy_instance): 1 usage ± Денис
    """Проверяет статус прокси за последние 3 минуты и обновляет статистику."""
    three_minutes_ago = timezone.now() - timedelta(minutes=3)
    recent_logs = ProxyRequestLog.objects.filter(proxy=proxy_instance, request_time__gte=three_minutes_ago,
                                                status='success')
    current_stats = ProxyStatistics.objects.filter(proxy=f"{proxy_instance.ip_address}:{proxy_instance.port}").first()
    previous_status = current_stats.status if current_stats else 'not_working'
    if recent_logs.exists():
        new_status = 'working'
    else:
        new_status = 'not_working'
    if previous_status == 'not_working' and new_status == 'working':
        bot.send_message(CHAT_ID, text=f"✅ Прокси {proxy_instance.ip_address}:{proxy_instance.port} снова работает!")
    elif previous_status == 'working' and new_status == 'not_working':
        bot.send_message(CHAT_ID, text=f"⚠️ Прокси {proxy_instance.ip_address}:{proxy_instance.port} не работает!")
    min_time, avg_time, max_time = measure_ip_rotation_time(proxy_instance)
    ip_pool_size = count_unique_ips(proxy_instance.id)
    ProxyStatistics.objects.update_or_create(
        proxy=f"{proxy_instance.ip_address}:{proxy_instance.port}",
        defaults={
            'status': new_status,
            'min_rotation_time': min_time,
            'avg_rotation_time': avg_time,
            'max_rotation_time': max_time,
            'ip_pool_size': ip_pool_size
        }
    )
    print(f"Прокси {proxy_instance.ip_address}:{proxy_instance.port} обновлен. Статус: {new_status}")
```

Эта функция автоматически уведомляет администратора о неработающих прокси-серверах, что позволяет оперативно реагировать на проблемы.

Таким образом, для разрабатываемой информационной системы можно выделить следующие ключевые функции:

- Статус прокси: Отслеживание текущего состояния каждого прокси-сервера (активен/неактивен).
- Минимальное, среднее и максимальное время ротации прокси: Анализ временных интервалов смены IP-адресов для каждого прокси.
- Пул IP-адресов: Хранение и управление списком IP-адресов, используемых прокси-серверами.
- Прокси (ip:port): Управление данными о прокси, включая их IP-адреса и порты.

Эти функции обеспечивают полный цикл управления и анализа данных о прокси-серверах, что позволяет автоматизировать их мониторинг и повысить эффективность работы системы.

2.3.2 Реализация базы данных

Для хранения данных о прокси-серверах была создана база данных, которая включает три основные таблицы:

Проху: хранит информацию о каждом прокси, такую как IP-адрес, порт, статус и другие параметры. Структура модели представлена на рисунке 2.8. Эта модель является основой для управления данными о прокси-серверах. .

```
class Proxy(models.Model): # Денис
    ip_address = models.GenericIPAddressField()
    port = models.IntegerField()
    username = models.CharField(max_length=255, null=True, blank=True)
    password = models.CharField(max_length=255, null=True, blank=True)
    is_active = models.BooleanField(default=True)
    status = models.CharField(max_length=50, default="unknown")
```

Рисунок 2.8 – Структура таблицы Проху

ProxyRequestLog: создаёт логи всех запросов, проходящих через прокси-сервер, включая время запроса, статус и другие метрики. Структура модели показана на рисунке 2.9. Логи позволяют анализировать использование прокси и выявлять потенциальные проблемы.

```
class ProxyRequestLog(models.Model): 15 usages ± Денис
    proxy = models.ForeignKey(Proxy, on_delete=models.CASCADE) # Связь с моделью Proxy
    ip_address = models.GenericIPAddressField(null=True, blank=True)
    request_time = models.DateTimeField()
    response_time = models.DateTimeField()
    error = models.TextField(default='', blank=True)
    status = models.CharField(max_length=10)
```

Рисунок 2.9 – Структура таблицы ProxyRequestLog

ProxyStatistics: содержит статистику по каждому прокси-серверу, включая количество уникальных IP, время ротации и другие показатели. Структура модели представлена на рисунке 2.10. Эта таблица используется для анализа производительности прокси-серверов и их оптимизации.

```
class ProxyStatistics(models.Model): 7 usages ± Денис
    proxy = models.CharField(max_length=255, unique=True, verbose_name="Прокси (ip:port)")
    status = models.CharField(max_length=50, choices=[('working', 'Работает'), ('not_working', 'Не работает')], verbose_name="Статус")
    min_rotation_time = models.FloatField(null=True, blank=True, verbose_name="Минимальное время ротации (сек)")
    avg_rotation_time = models.FloatField(null=True, blank=True, verbose_name="Среднее время ротации (сек)")
    max_rotation_time = models.FloatField(null=True, blank=True, verbose_name="Максимальное время ротации (сек)")
    ip_pool_size = models.IntegerField(default=0, verbose_name="Пул IP-адресов (количество доступных IP)")
```

Рисунок 2.10 – Структура модели ProxyStatistics.

Созданная база данных обеспечивает централизованное хранение и управление данными, что упрощает их обработку и анализ.

2.3.3 Компоненты пользовательского интерфейса

После завершения разработки проект переносится на Docker, что обеспечивает его удобное развертывание и масштабируемость. Для управления системой используется веб-интерфейс, доступный по адресу <http://localhost:81/admin/>. Этот интерфейс предоставляет удобные

инструменты для добавления, удаления, изменения данных о прокси, а также для просмотра логов и статистики.

На рисунке 2.11 показан вид страницы admin, который включает следующие основные пункты (рисунок 2.12):

- Proxies: Управление данными о прокси-серверах.
- Proxy request logs: Просмотр логов запросов, проходящих через прокси.
- Proxy statistics: Анализ статистики по каждому прокси.

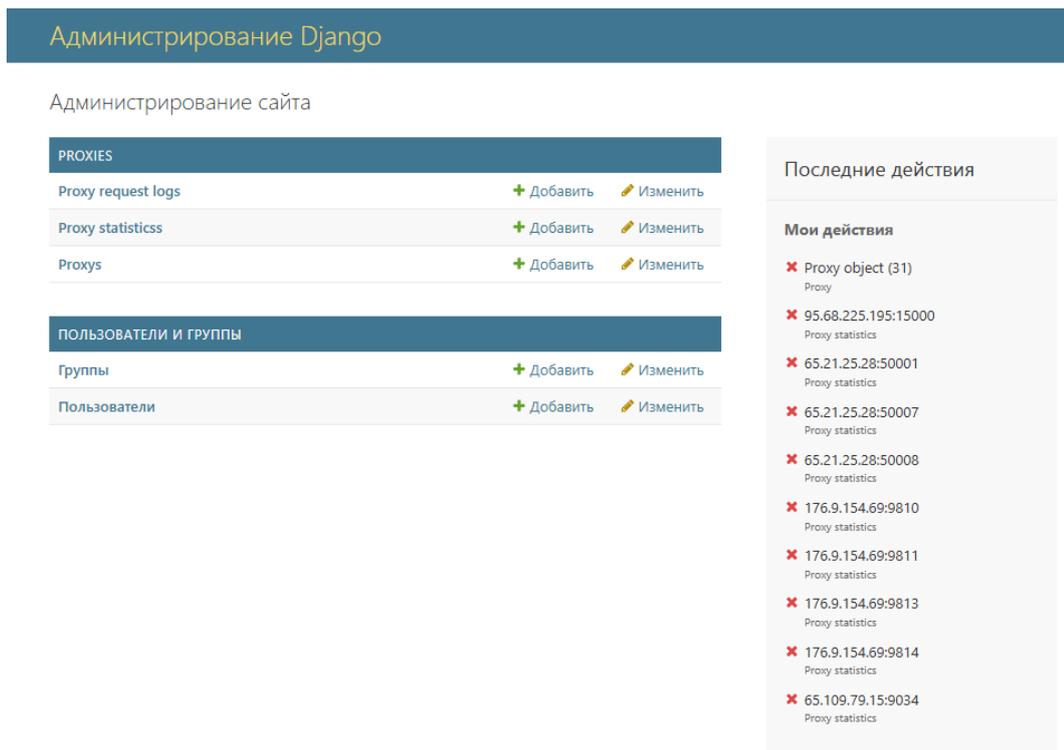


Рисунок 2.11 – Страница admin

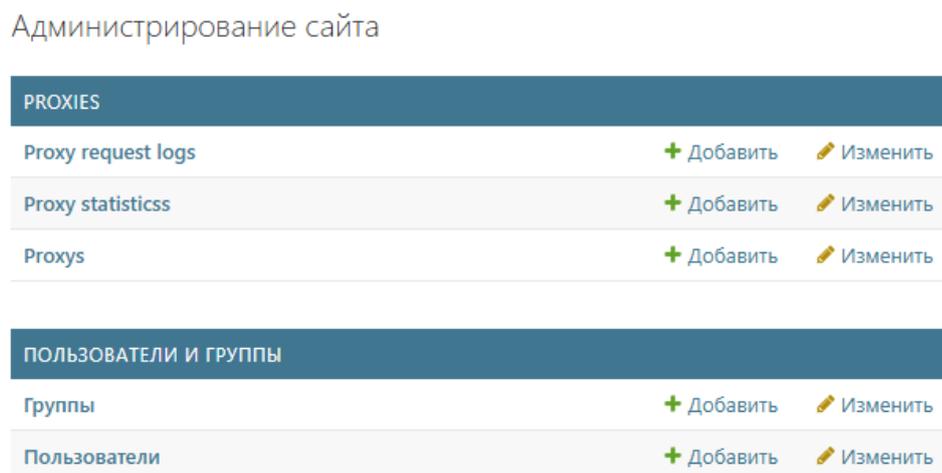


Рисунок 2.12 – Разделы на странице admin

При переходе на пункт «Proxies» открывается страница, показанная на рисунке 2.13. Здесь можно добавлять новые прокси-сервера, изменять или удалять существующие.



Рисунок 2.13 – Сайт admin после перехода на «Proxies»

Для добавления прокси используется форма, представленная на рисунке 2.14. Для изменения или удаления прокси достаточно кликнуть на соответствующий элемент, как показано на рисунке 2.15.

Для добавления прокси на странице из рисунка 2.13 необходимо нажать на кнопку «добавить» напротив «Proxys», после чего открывается окно добавления прокси в формате ссылки, как показано на примере рисунка 2.14.

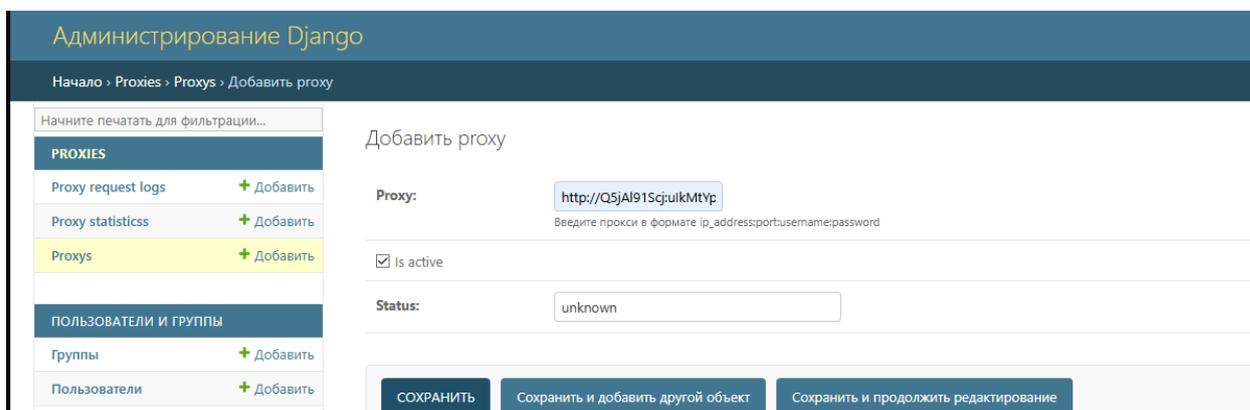


Рисунок 2.14 – Добавление прокси-сервера

Для удаления или же изменения прокси-сервера необходимо просто кликнуть на требуемый прокси-сервер и изменить или же удалить, как показано на рисунке 2.15.

В разделе «Proxies» (рисунок 2.12) имеется возможность перехода в разделы «Proxy request logs», «Proxy statistics» и «Proxys». На рисунках 2.16, 2.17 и 2.18 представлены результаты переходов на эти ссылки. Эти разделы имеют интуитивно понятный интерфейс и предоставляют администратору всю требуемую информацию.

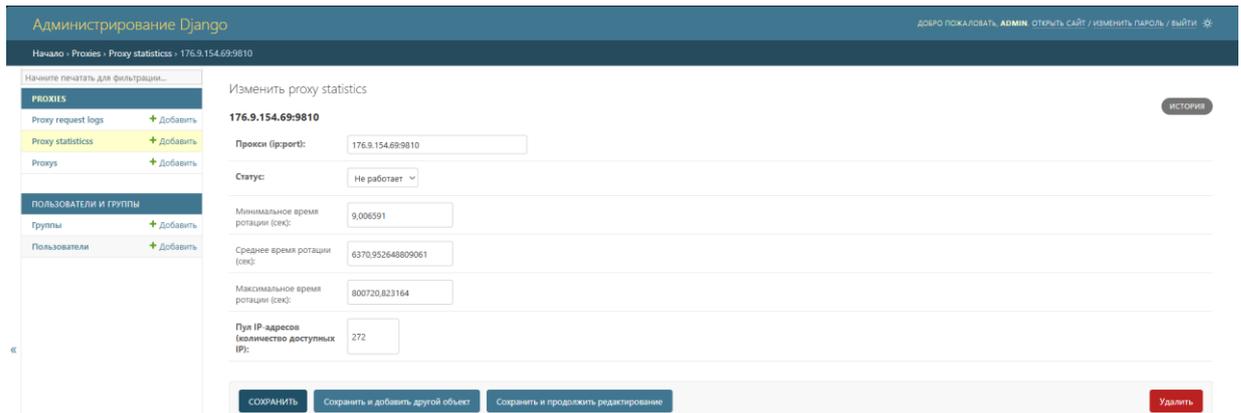


Рисунок 2.15 – Изменение или удаления прокси-сервера

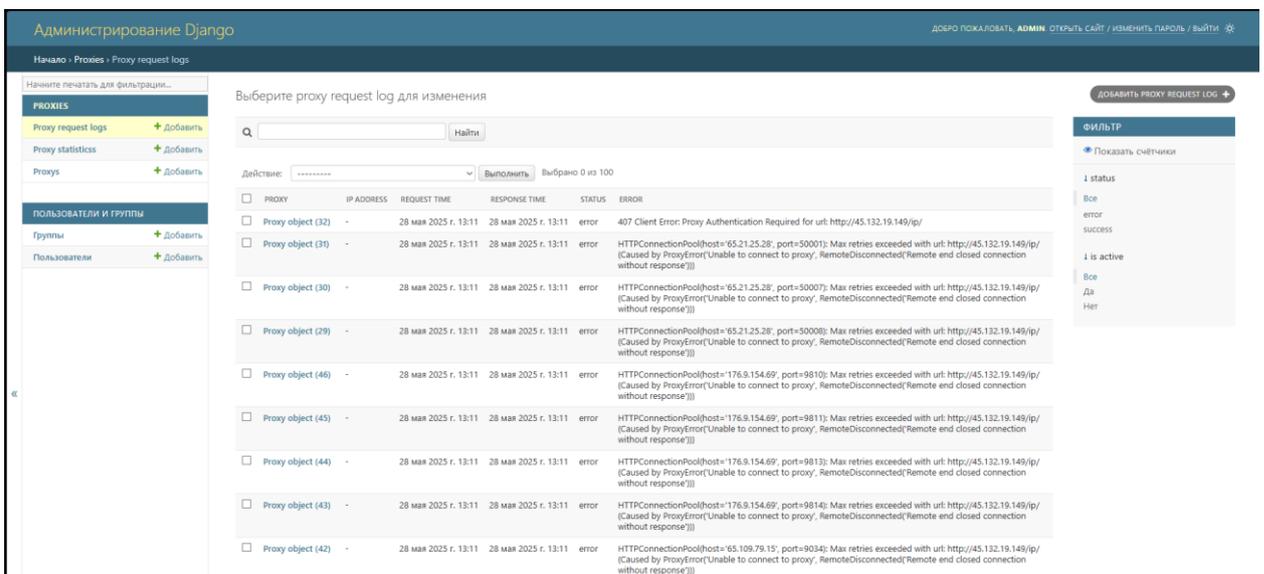


Рисунок 2.16 – Раздел «Proxy request logs»

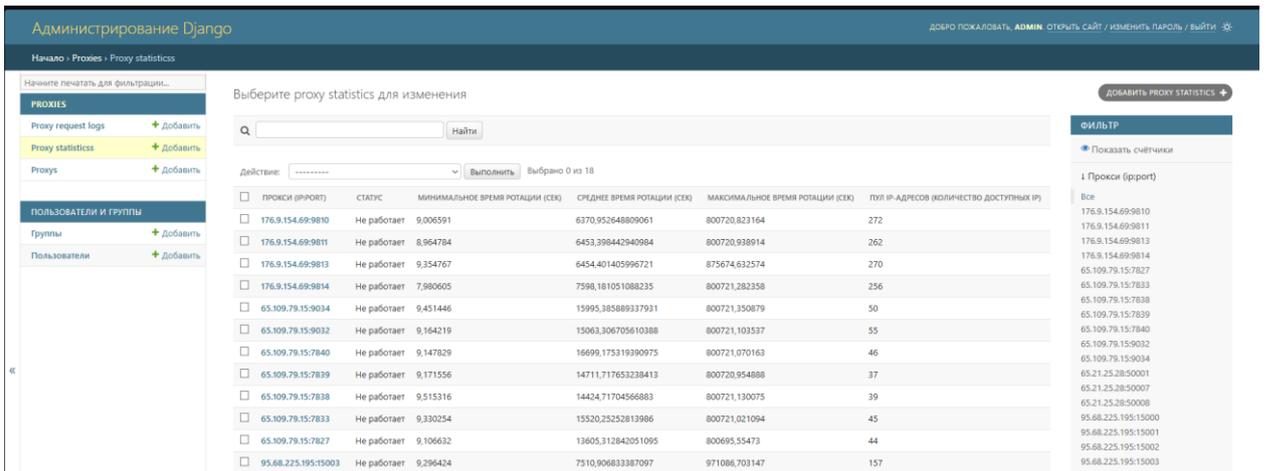


Рисунок 2.17 – Раздел «Proxy statistics»

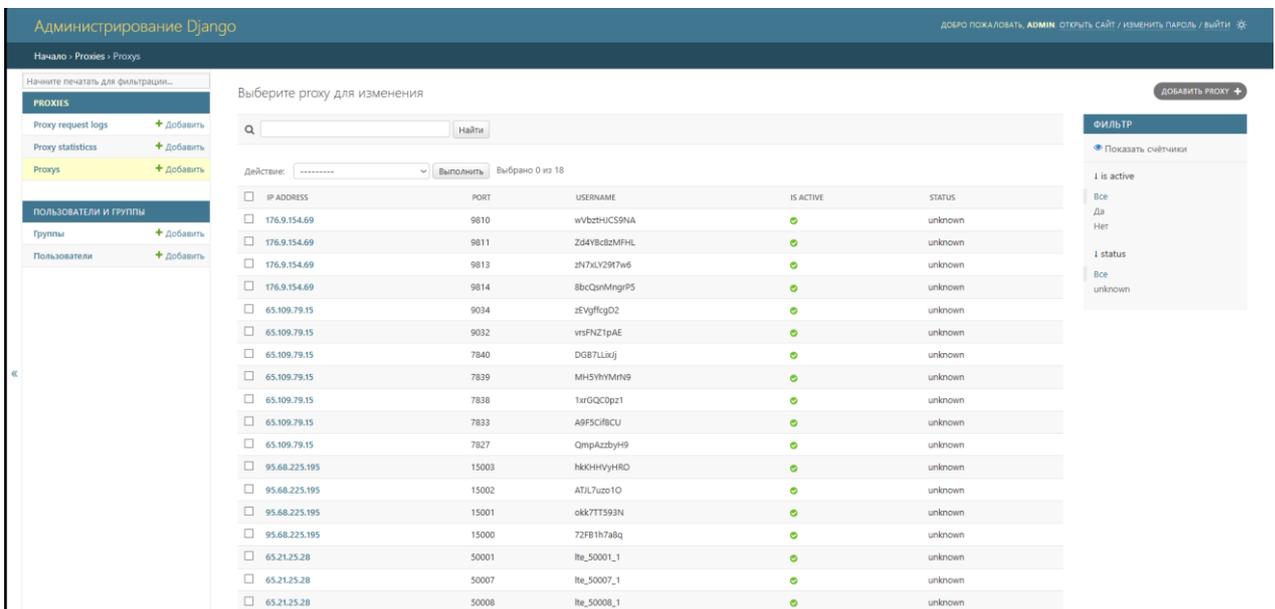


Рисунок 2.18 – Раздел «Proxys»

После перехода на страницу Proxy-statistics появляется соответствующий раздел, показанный на рисунке 2.19.

Proxy Statistics

Прокси (ip:порт)	Статус	Минимальное время ротации (сек)	Среднее время ротации (сек)	Максимальное время ротации (сек)	Пул IP-адресов
65.21.25.28-50008	Работает	10,169936	45138,46152997674	800664,121636	38
65.21.25.28-50007	Работает	9,353918	7644,796059552941	800664,397901	175
65.21.25.28-50001	Работает	1,482828	3418,1154183881117	800664,47901	424
95.68.225.195-15000	Не работает	9,272793	2866,817086118211	339276,845732	155
95.68.225.195-15001	Не работает	9,306393	3009,458097942953	339276,868998	167

Рисунок 2.19 – «Proxy-statistics»

На странице «Proxy statistics» (рисунок 2.19) отображаются данные о каждом прокси, включая его статус, время ротации и пул IP-адресов.

Так же для удобства анализа собранной статистики, ниже на данной странице была добавлена возможность переводить данные в графический вид. Так, на рисунке 2.20 показан график, который содержит в себе данные о времени «безотказной» работы прокси и времени отклика.

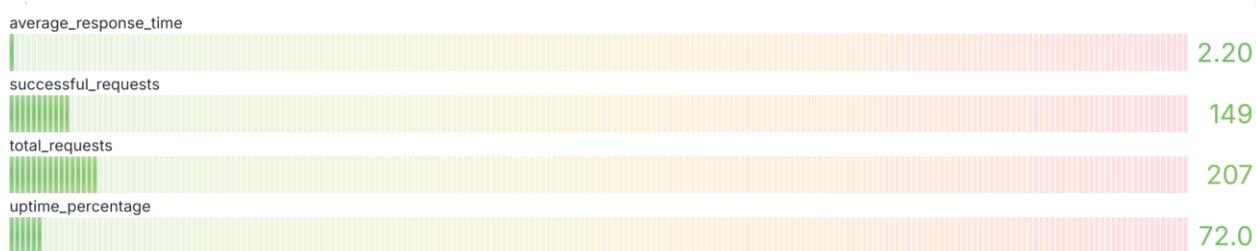


Рисунок 2.20 – График сбора статистики

Данный график можно перевести в разные его виды, как показано на рисунке 2.21

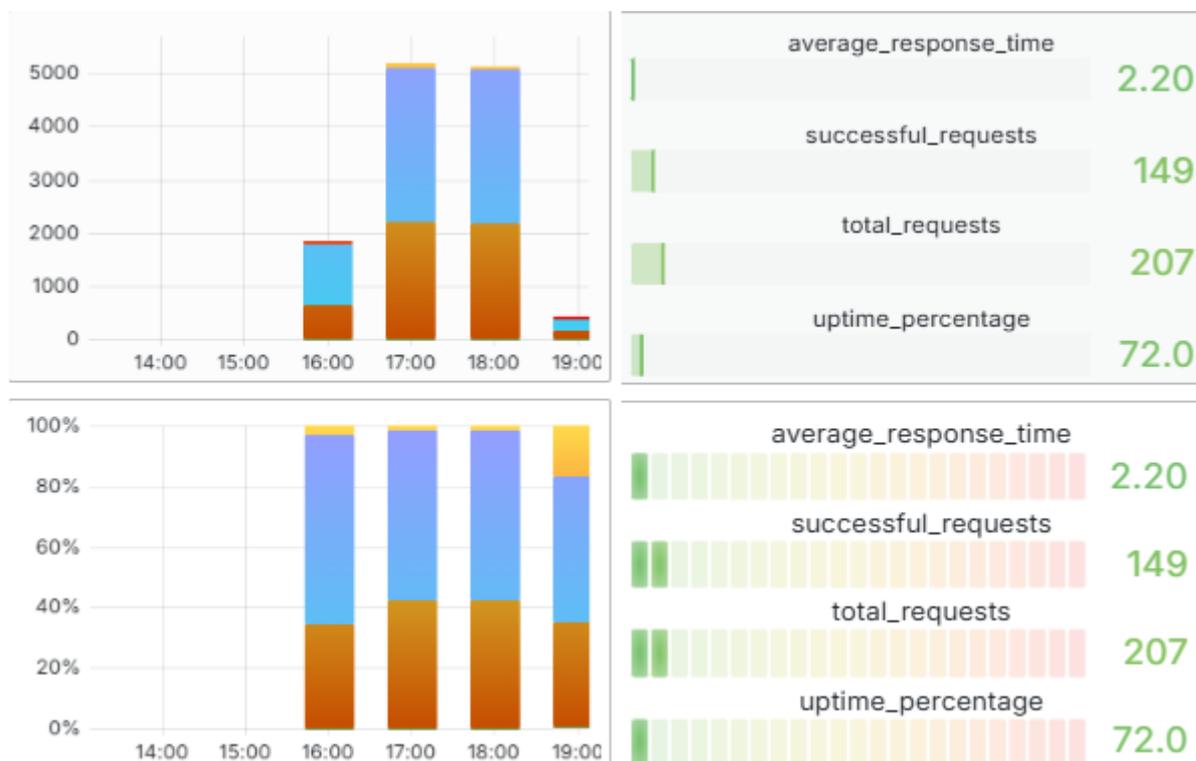


Рисунок 2.21 – Виды графика сбора статистики

Если прокси не работает более 3 минут, система отправляет уведомление в Telegram-бот. Примеры соответствующих сообщений в

Telegram-боте показаны на рисунке 2.22. Это позволяет администраторам оперативно реагировать на проблемы и поддерживать стабильную работу системы.

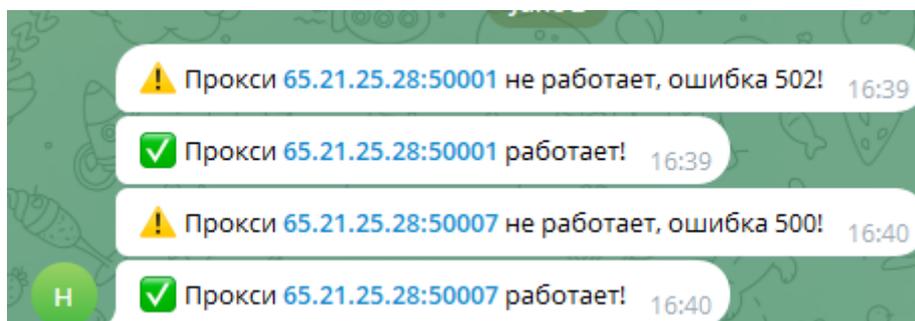


Рисунок 2.22 – Уведомления о рабочих и не рабочих прокси в Telegram-боте

3 Оценка эффективности внедрения информационной системы

3.1 Общие положения

Основными факторами эффективного применения информационных технологий в системе мониторинга прокси являются повышение качества управления прокси-серверами, надежность их функционирования, сокращение времени на обнаружение и устранение неисправностей, увеличение объема обрабатываемых запросов, снижение времени отклика прокси, а также повышение производительности труда разработчиков и пользователей системы мониторинга [25]. Важными аспектами также являются снижение затрат на эксплуатацию прокси, улучшение качества обслуживания клиентов, увеличение доступности прокси-серверов и снижение количества ошибок в их работе. Для реализации конкретного проекта системы мониторинга прокси необходимо четко определить, какие показатели и параметры необходимо включить в технико-экономическое обоснование. Это включает оценку затрат на разработку системы, анализ ожидаемых доходов от её внедрения, а также расчет графика возврата вложенных средств, чтобы обосновать необходимость проектирования и внедрения системы. Основными задачами, стоящими при создании системы мониторинга прокси-серверов, являются:

- минимизация стоимости разработки и эксплуатации системы.
- обеспечение требуемого качества мониторинга и управления прокси.

Основными качествами функционирования системы мониторинга прокси-серверов являются:

- простота и технологичность разработки: система должна быть легко масштабируемой и поддерживаемой;

- эффективность мониторинга: своевременное обнаружение проблем с прокси и их устранение;
- удобство использования и обслуживания: интуитивно понятный интерфейс для администраторов и пользователей;
- экономическая целесообразность: снижение затрат на эксплуатацию прокси;
- улучшение условий работы администраторов: автоматизация рутинных задач, таких как проверка доступности прокси и анализ их производительности.

В сфере мониторинга прокси оценка эффективности внедрения системы осуществляется с помощью множества показателей. Обобщающими показателями эффективности системы мониторинга прокси являются показатели экономической эффективности. Расчет затрат на разработку и эксплуатацию системы обычно не вызывает сложностей, однако расчет результатов (например, экономии времени и ресурсов) может быть сложной задачей. Часто такие показатели определяются экспертным путем или по аналогии с другими системами мониторинга. Для оценки эффективности системы мониторинга прокси могут использоваться две группы показателей: интегральные традиционные показатели и частные показатели. Обычно в качестве экономических показателей используются [26]:

- процент времени безотказной работы прокси (Uptime);
- среднее время отклика прокси (Response Time);
- количество успешно обработанных запросов за единицу времени (Throughput);
- экономия времени на ручной мониторинг и устранение неисправностей;
- снижение эксплуатационных затрат на прокси;
- экономия затрат на материалы и оборудование для поддержки прокси-серверов;
- снижение количества ошибок в работе прокси-серверов;

- годовая экономия затрат на обслуживание прокси-серверов.

Для успешного внедрения системы мониторинга прокси-серверов важно учитывать не только технические, но и экономические аспекты, чтобы обеспечить её рентабельность и эффективность.

3.2 Расчёт экономической эффективности

Для оценки экономической эффективности внедрения информационной системы мониторинга прокси-серверов были проведены расчёты, включающие анализ трудозатрат, стоимостных затрат и периода окупаемости.

1. Снижение трудозатрат

Показатели трудозатрат по базовому варианту (T_0) и предлагаемому варианту (T_j) представлены в таблице 3.1.

Таблица 3.1 – Показатели трудозатрат на обработку информации

Название работ	Базовая система(T_0)	Предлагаемая система (T_j)
Проверка доступности прокси	120 часов/год	40 часов/год
Анализ логов прокси	150 часов/год	50 часов/год
Формирование отчётов	100 часов/год	30 часов/год
итого	370 часов/год	120 часов/год

1. Показатель снижения трудовых затрат (ΔT) рассчитывается по формуле (3.1):

$$\Delta T = T_0 - T_j = 370 - 120 = 250 \text{ часов/год.} \quad (3.1)$$

Коэффициент снижения трудозатрат (K_T):

$$K_T = \Delta T / T_0 \times 100\% = 250/370 * 100 = 67.6\% \quad (3.2)$$

2. Снижение стоимостных затрат.

Часовая ставка оплаты труда администратора составляет 300 рублей/час.

Тогда затраты на оплату труда составляют:

Базовая система (T_0):

$$C_0 = T_0 \times \text{ставка} = 370 \times 300 = 111\,000 \text{ рублей/год.} \quad (3.3)$$

Предлагаемая система (T_j):

$$C_j = T_j \times \text{ставка} = 120 \times 300 = 36\,000 \text{ рублей/год} \quad (3.4)$$

Снижение затрат на оплату труда:

$$\Delta C = C_0 - C_j = 111\,000 - 36\,000 = 75\,000 \text{ рублей/год.} \quad (3.5)$$

3. Рассчитаем период окупаемости. Для этого оценим капитальные вложения в проект. Капитальные вложения на разработку и внедрение системы составляют:

- разработка ПО: 50 000 рублей.
- закупка оборудования: 40 000 рублей.
- лицензии и ПО: 20 000 рублей.

Общие капитальные вложения (K_j):

$$K_j = 50\,000 + 40\,000 + 20\,000 = 110\,000 \text{ рублей.} \quad (3.6)$$

Период окупаемости ($T_{ок}$) рассчитывается по формуле:

$$T_{ок} = \frac{K_j}{\Delta C} = \frac{110,000}{75,000} = 1.47 \text{ года (или 17.6 месяцев)} \quad (3.7)$$

4. Экономический эффект.

Экономический эффект рассчитывается по формуле:

$$E = \Delta C - E_n * K_j \quad (3.8)$$

где $E_n = 0.15$ – нормативный коэффициент эффективности.

Тогда:

$$E = 75\,000 - 0.15 \times 110\,000 = 75\,000 - 16\,500 = 58\,500 \text{ рублей/год.} \quad (3.9)$$

Таким образом, внедрение системы мониторинга прокси-серверов позволяет:

- сократить время проверки прокси с 370 часов/год до 120 часов/год;
- снизить трудозатраты на 67.6%, стоимостные затраты на 75 000 рублей/год;
- добиться экономического эффекта от внедрения системы в размере 58 500 рублей/год, что подтверждает её рентабельность;
- срок окупаемости проекта – 1.47 года.

Внедрение системы мониторинга прокси-серверов также позволит:

- уменьшить количество ошибок при анализе логов и формировании отчётов;
- повысить прозрачность и точность мониторинга;
- осуществить автоматизация процессов: система позволяет автоматически проверять прокси в мульти поточности, что значительно ускоряет обработку данных;
- обеспечить удобство использования: интуитивно понятный интерфейс и автоматическое формирование отчётов упрощают работу администратора;
- гибкость: возможность масштабирования системы под увеличение количества прокси;
- повысить качество предоставляемых услуг за счёт оперативного выявления и устранения проблем с прокси-серверами;
- увеличить удовлетворённость пользователей благодаря стабильной работе системы.

Внедрение системы мониторинга прокси-серверов позволило достичь значительных улучшений в эффективности работы, сократить затраты и повысить качество мониторинга. Экономическая и организационная выгода подтверждают целесообразность её использования.

ЗАКЛЮЧЕНИЕ

Целью выпускной квалификационной работы являлась разработка системы мониторинга прокси-серверов для ПАО «Совкомбанк», которая направлена на автоматизацию процессов контроля за состоянием прокси и обеспечение безопасности сетевой инфраструктуры банка. Для достижения поставленной цели были успешно решены все поставленные задачи:

- проведён анализ текущей ИТ-инфраструктуры ПАО «Совкомбанк», выявлены недостатки в информационных процессах;
- определены требования к системе мониторинга прокси-серверов;
- разработана архитектура создаваемой системы;
- реализовано программное обеспечение системы;
- проведено опытное внедрение системы мониторинга в эксплуатацию;
- оценена экономическая эффективность от внедрения системы.

Система была создана с учетом современных требований к надежности и эффективности, что позволяет минимизировать риски, связанные с отказами оборудования и утечкой данных. Архитектура системы включает функциональные модули для проверки состояния прокси, подсчета уникальных IP-адресов и измерения времени ротации IP. Реализованная система также предусматривает возможность отправки уведомлений о сбоях и проблемах в работе прокси-серверов, что позволяет оперативно реагировать на возникающие ситуации и обеспечивать высокое качество обслуживания клиентов.

Тестирование системы подтвердило ее работоспособность и соответствие заявленным требованиям. Внедрение данной системы в эксплуатацию позволит ПАО «Совкомбанк» существенно повысить уровень безопасности данных, улучшить контроль за сетевым трафиком и оптимизировать внутренние процессы. Автоматизация мониторинга прокси-серверов не только сократит время на выявление и устранение проблем, но и повысит общую эффективность работы ИТ-инфраструктуры банка.

Полученные результаты проекта могут быть применены не только в ПАО «Совкомбанк», но и в других финансовых учреждениях, стремящихся улучшить управление своей сетевой инфраструктурой. Разработанная система может быть адаптирована под специфические требования других организаций, что делает её универсальным решением для обеспечения безопасности и надежности работы прокси-серверов.

Таким образом, реализация системы мониторинга прокси-серверов является важным шагом на пути к цифровизации и оптимизации бизнес-процессов в банке, что соответствует современным тенденциям финансового сектора и позволяет ПАО «Совкомбанк» оставаться конкурентоспособным на рынке.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Долганова, О. И. Моделирование бизнес-процессов : учебник и практикум для вузов / О. И. Долганова, Е. В. Виноградова, А. М. Лобанова ; под редакцией О. И. Долгановой. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2025. – 322 с. – (Высшее образование). – ISBN 978-5-534-17914-9. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/560175> – Загл. с экрана.
2. Мониторинг Docker контейнеров [Электронный ресурс]. – Режим доступа: <https://www.docker.com/> – Загл. с экрана.
3. Как использовать Django для мониторинга [Электронный ресурс]. – Режим доступа: <https://www.djangoproject.com/> – Загл. с экрана.
4. Docker и Django: полное руководство [Электронный ресурс]. – Режим доступа: <https://testdriven.io/blog/dockerizing-django/> – Загл. с экрана.
5. Гордеев, С. И. Организация баз данных : учебник для вузов / С. И. Гордеев, В. Н. Волошина. – 2-е изд., испр. и доп. – Москва : Издательство Юрайт, 2025. – 691 с. – (Высшее образование). – ISBN 978-5-534-21115-3. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/559377> – Загл. с экрана.
6. Нестеров, С. А. Базы данных : учебник и практикум для вузов / С. А. Нестеров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2024. – 258 с. – (Высшее образование). – ISBN 978-5-534-18107-4. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/536687> – Загл. с экрана.
7. Зараменских, Е. П. Разработка информационных систем : учебник и практикум для вузов / Е. П. Зараменских. – 2-е изд. – Москва : Издательство Юрайт, 2025. – 78 с. – (Высшее образование). – ISBN 978-5-534-21420-8. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/571333> – Загл. с экрана.

8. Грекул, В. И. Проектирование информационных систем : учебник и практикум для вузов / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2025. – 404 с. – (Высшее образование). – ISBN 978-5-534-19505-7. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/560976> – Загл. с экрана.

9. Использование прокси в Python [Электронный ресурс]. – Режим доступа: <https://www.geeksforgeeks.org/using-proxies-in-python/> – Загл. с экрана.

10. Прокси-серверы и их использование в Python [Электронный ресурс]. – Режим доступа: <https://www.scrapingbee.com/blog/python-proxy/> – Загл. с экрана.

11. Docker для разработки Django [Электронный ресурс]. – Режим доступа: <https://www.freecodecamp.org/news/dockerizing-django-with-postgres-unicorn-and-nginx/> – Загл. с экрана.

12. Мониторинг сетевого трафика с помощью прокси [Электронный ресурс]. – Режим доступа: <https://www.telerik.com/blogs/how-to-monitor-network-traffic-with-proxy> – Загл. с экрана.

13. Мониторинг Docker контейнеров с помощью cAdvisor [Электронный ресурс]. – Режим доступа: <https://github.com/google/cadvisor> – Загл. с экрана.

14. Использование Django для создания API [Электронный ресурс]. – Режим доступа: <https://www.django-rest-framework.org/> – Загл. с экрана.

15. Docker Compose для Django [Электронный ресурс]. – Режим доступа: <https://docs.docker.com/compose/django/> – Загл. с экрана.

16. Прокси-серверы для тестирования [Электронный ресурс]. – Режим доступа: https://www.selenium.dev/documentation/en/guidelines_and_recommendations/proxy_servers/ – Загл. с экрана.

17. Docker и CI/CD для Django [Электронный ресурс]. – Режим доступа: <https://www.jamescoyle.net/how-to/2020-docker-and-ci-cd-for-django> – Загл. с экрана.

18. Использование прокси для веб-скрейпинга [Электронный ресурс]. – Режим доступа: <https://www.analyticsvidhya.com/blog/2021/06/web-scraping-with-python-using-proxies/> – Загл. с экрана.
19. Мониторинг и логирование в Django [Электронный ресурс]. – Режим доступа: <https://www.cactusgroup.com/blog/2020/06/30/logging-in-django/> – Загл. с экрана.
20. Мониторинг приложений с помощью ELK Stack [Электронный ресурс]. – Режим доступа: <https://www.elastic.co/what-is/elk-stack> – Загл. с экрана.
21. Мониторинг и управление контейнерами [Электронный ресурс]. – Режим доступа: <https://www.redhat.com/en/topics/containers/what-is-a-container> – Загл. с экрана.
22. Прокси-серверы для работы с API [Электронный ресурс]. – Режим доступа: <https://www.toptal.com/python/python-proxy-server> – Загл. с экрана.
23. Использование прокси в веб-приложениях [Электронный ресурс]. – Режим доступа: <https://www.smashingmagazine.com/2020/06/proxy-servers-web-applications/> – Загл. с экрана.
24. Создание RESTful API с Django [Электронный ресурс]. – Режим доступа: <https://www.django-rest-framework.org/tutorial/quickstart/> – Загл. с экрана.
25. Нетесова, О. Ю. Информационные системы и технологии в экономике: учебное пособие для вузов / О. Ю. Нетесова. – 4-е изд., испр. и доп. – Москва: Издательство Юрайт, 2024. – 178 с. – (Высшее образование). – ISBN 978-5-534-15926-4. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/538283> – Загл. с экрана.
26. Экономика информационных систем : учебник для вузов / А. Л. Рыжко, Н. А. Рыжко, Н. М. Лобанова, Е. О. Кучинская. – 2-е изд., испр. и доп. – Москва : Издательство Юрайт, 2025. – 176 с. – (Высшее образование). – ISBN 978-5-534-05545-0. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/563467/> – Загл. с экрана.