

ИНТЕГРАЦИЯ СИСТЕМЫ ПОДДЕРЖКИ С ОБРАТНОЙ СВЯЗЬЮ В ВЕБ-ОРИЕНТИРОВАННУЮ МЕДИЦИНСКУЮ ПЛАТФОРМУ: АРХИТЕКТУРА, РЕАЛИЗАЦИЯ И ИНЖЕНЕРНЫЕ ЭКСПЕРИМЕНТЫ

Топузов Э. Р.¹, аспирант, ✉ edem.topuzov@gmail.com, orcid.org/0009-0003-9208-7931

¹ Санкт-Петербургский государственный университет,
Университетский пр., д. 28, Петергоф, 198504, Санкт-Петербург, Россия

Аннотация

В статье описана архитектура и реализация подсистемы поддержки с обратной связью, интегрированной в ранее разработанную веб-ориентированную систему для ведения медицинских исследований. Предложенное решение обеспечивает автоматическую и полуавтоматическую регистрацию ошибок, а также позволяет пользователям и службе поддержки взаимодействовать в рамках замкнутого цикла улучшения качества. Также представлены результаты инженерных экспериментов, направленных на оценку устойчивости и масштабируемости системы.

Ключевые слова: *система обращений, система поддержки, веб-ориентированная система*

Цитирование: Топузов Э. Р. Интеграция системы поддержки с обратной связью в веб-ориентированную медицинскую платформу: архитектура, реализация и инженерные эксперименты // Компьютерные инструменты в образовании. 2025. № 1. С. 76–90. [doi:10.32603/2071-2340-2025-1-76-90](https://doi.org/10.32603/2071-2340-2025-1-76-90)

1. ВВЕДЕНИЕ

Современные веб-ориентированные системы, разрабатываемые для ведения и анализа данных медицинских исследований, предъявляют высокие требования к надежности, масштабируемости и взаимодействию с конечными пользователями [1]. Одной из ключевых характеристик таких систем становится способность не только корректно обрабатывать и хранить данные, но и своевременно реагировать на возникающие ошибки, запросы и предложения со стороны пользователей [2].

Публикации, посвященные поддержке пользователей в медицинских информационных системах, трактуют ее преимущественно как организационную функцию-сервис — выделение обученного персонала, разработку регламентов коммуникации и программ повышения квалификации врачей после внедрения МИС. При этом механизмы быстрого замыкания цикла «ошибка — обратная связь — исправление» остаются неформализованными [3–5].

Предлагаемая в работе подсистема восполняет этот пробел, так как реализует единый технологический цикл регистрации, категоризации и аналитики обращений непосредственно в инфраструктуре веб-платформы, что позволяет измеримо сокращать время реагирования и снижать эксплуатационные риски.

Обеспечить устойчивое функционирование медицинской информационной системы, на которой ежедневно работают врачи, невозможно без формализованного канала взаимодействия между конечным пользователем и службой сопровождения [6, 7]. Даже при высоком качестве самого программного решения отсутствие понятного коммуникационного шлюза для обращений пользователей приводит к падению пользовательской лояльности и росту сопротивления, особенно на этапе внедрения [8].

Ранее нами была разработана распределенная веб-ориентированная система, архитектура которой включает модули управления данными, статистической обработки, аутентификации и авторизации, построенные с применением современных средств контейнеризации и реляционного хранения данных [9].

Однако в процессе эксплуатации системы была выявлена необходимость в расширении ее функциональности посредством внедрения полноценной подсистемы поддержки — как с целью сбора обратной связи от пользователей, так и для автоматического реагирования на сбои.

В данной работе описывается проектирование, реализация и интеграция в существующую архитектуру подсистемы обработки обращений с обратной связью, а также проводится серия инженерных экспериментов, направленных на оценку ее масштабируемости, устойчивости к отказам и эффективности в различных сценариях использования. Подсистема обеспечивает автоматическую регистрацию ошибок, ручное создание обращений пользователями и сотрудниками поддержки, а также она построена с учетом требований к безопасности, отслеживаемости и масштабируемости [2].

Таким образом, цель настоящего исследования заключается в разработке и эмпирической валидации инженерного решения, обеспечивающего поддержку полного жизненного цикла обработки обращений, включая регистрацию, категоризацию, эскалацию, разрешение и закрытие, с минимальной задержкой реакции и высокой степенью отказоустойчивости.

2. АРХИТЕКТУРА СИСТЕМЫ ПОДДЕРЖКИ С ОБРАТНОЙ СВЯЗЬЮ

Разработанная подсистема поддержки с обратной связью (далее — Подсистема) интегрирована в существующую архитектуру веб-ориентированной платформы медицинских исследований и реализована с учетом требований к масштабируемости, отказоустойчивости и безопасности.

В связи с тем, что основная часть приложения написана на Ruby, для Подсистемы также было принято решение использовать данный язык программирования. Был применен комплекс *kicks* — высокопроизводительная программная платформа для фоновой обработки задач, использующий брокер сообщений RabbitMQ [10]. Работа указанного комплекса осуществляется в связке с библиотекой *bunny*, представляющей собой полнофункциональный клиент для Расширенного протокола организации очереди сообщений (Advanced Message Queuing Protocol), то есть AMQP-клиент [11], который предоставляет разработчику программный интерфейс для установки соединения, управления очередями, публикации и получения сообщений, а также обработки событий в асинхронных или синхронных сценариях обмена сообщениями.

Архитектура подсистемы включает следующие компоненты (рисунок 1):

- модуль регистрации и классификации обращений (А);
- очередь обработки событий и ошибок (Б);
- интерфейс взаимодействия с пользователем (В);
- административный модуль поддержки (Г);
- модуль аналитики и мониторинга (Д).

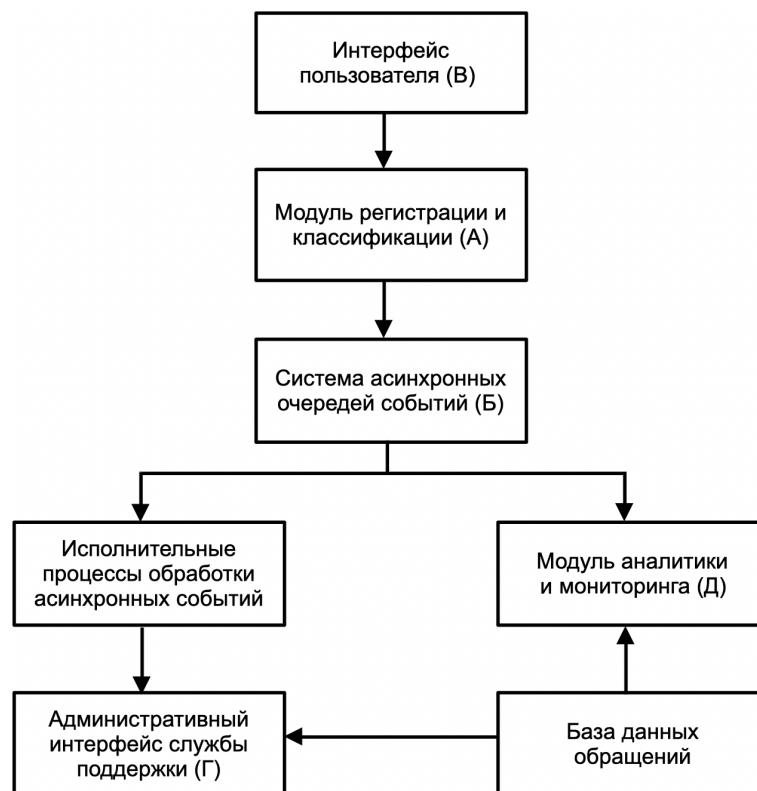


Рис. 1. Схема архитектуры системы поддержки с обратной связью

2.1. Модуль регистрации и классификации обращений

Данный модуль отвечает за первичную фиксацию входящих обращений. Источниками обращений могут быть:

- автоматически генерируемые события системы (ошибки выполнения, исключения, таймауты);
- действия пользователей (жалобы, предложения, запросы через пользовательский интерфейс);
- обращения, создаваемые вручную специалистами технической поддержки.

Каждое обращение снабжается метаданными: временная метка, автор, тип обращения (ошибка, предложение, запрос), критичность, категория (интерфейс, база данных, обработка данных и др.), а также контекст исполнения (например: адрес запроса, параметры запроса, идентификатор пользователя, стек вызовов в случае ошибок).

Если обращение инициировано системой (через журнал обработки исключений) — оно получает категорию «автоматически сгенерированное» и критичность «высокая».

Поддерживается также возможность последующего редактирования категории и критичности обращения специалистом.

На данный момент классификация обращений реализована с использованием формализованных логических правил, задаваемых вручную. Например, при наличии таких слов, как «ошибка», «не загружается», «502», обращение автоматически классифицируется как техническая ошибка. Если обращение создается пользователем с ролью «исследователь» и в нем присутствуют слова из словаря улучшений («хотелось бы», «предлагаю»), оно классифицируется как предложение по улучшению.

Разумеется, эвристические методы обладают рядом ограничений, включая низкую гибкость, высокую чувствительность к формулировкам и необходимость постоянной поддержки правил в актуальном состоянии по мере расширения словаря и изменяющихся сценариев использования системы. В связи с этим в рамках будущих этапов развития подсистемы планируется внедрение механизмов автоматической классификации на основе методов обработки естественного языка (Natural Language Processing, NLP) и машинного обучения. Внедрение интеллектуальных методов классификации, обладающих адаптивностью и устойчивостью к вариативности языка, позволит повысить точность и консистентность категоризации обращений, уменьшить долю ручной переклассификации со стороны службы поддержки, а также ускорить процессы устранения неполадок и добавления нового функционала [12, 13].

2.2. Очередь обработки событий и ошибок

Центральным элементом асинхронной обработки является система очередей сообщений, реализованная в виде брокера сообщений, обеспечивающего надежную доставку и временное хранение обращений в виде сериализованных структур данных. Все зарегистрированные обращения (включая как пользовательские, так и автоматически сгенерированные системой) направляются в очередь соответствующего уровня критичности и категории.

Каждая очередь обрабатывается пулом специализированных рабочих процессов (worker-процессов) [14], которые осуществляют:

- десериализацию сообщения;
- валидацию структуры данных;
- выполнение требуемых процедур обработки (например: запись в базу данных, анализ контекста, оповещение ответственных лиц).

Рабочие процессы запускаются на отдельных экземплярах приложений, могут масштабироваться горизонтально и функционируют независимо друг от друга, что обеспечивает устойчивость к сбоям отдельных обработчиков.

Для гарантированной доставки сообщений и исключения потерь реализована поддержка подтверждений обработки (acknowledgement) и повторных попыток обработки при сбоях (retry mechanism). В случае возникновения ошибки при обработке сообщения, оно временно помещается в отложенную очередь (dead-letter queue) с последующим повторным захватом. Такое решение минимизирует вероятность потери данных при кратковременных отказах компонентов и позволяет зафиксировать системные ошибки для последующего анализа [14].

Брокер сообщений сохраняет информацию о состоянии очередей и подтверждениях доставки в отказоустойчивом хранилище, что обеспечивает корректное восстановление работы подсистемы после перезапуска.

2.3. Интерфейс взаимодействия с пользователем

Для пользователей предоставляется простой и интуитивный интерфейс создания обращения. Пользователь может выбрать тип обращения, указать описание проблемы, приложить скриншоты или иные материалы. Все обращения отображаются в личном кабинете пользователя, с возможностью отслеживания статуса и получения ответов от поддержки.

Пользовательский интерфейс предоставляет следующие функции:

- 1. Создание обращения: пользователь может зафиксировать новую заявку, выбрав из предложенного списка тип обращения (техническая ошибка, предложение по улучшению, вопрос по функциональности и т.д.), указав краткое и полное описание, прикрепив при необходимости вспомогательные материалы (изображения, документы). Дополнительно указывается критичность и контекст (например: идентификатор пациента или исследования, в рамках которого возникла проблема).
- 2. Отслеживание статуса: каждое обращение отображается в личной области пользователя с указанием текущего состояния обработки (новое, в работе, ожидает ответа, закрыто, исполнено), времени создания, а также истории переписки.
- 3. Обратная связь: после обработки обращения пользователь имеет возможность оценить качество решения и при необходимости повторно открыть обращение в случае неудовлетворительного результата.
- 4. Фильтрация и поиск: реализована возможность фильтрации обращений по дате, категории, статусу и ключевым словам, а также полнотекстовый поиск по описанию и комментариям.

Клиентская часть интерфейса реализована как модуль основного веб-приложения, с использованием адаптивной верстки, обеспечивающей корректное отображение на различных устройствах (в том числе планшетах и смартфонах). Обмен данными между интерфейсом и серверной частью осуществляется посредством защищенных HTTP-запросов с использованием REST-подхода [15]. Все операции выполняются в рамках сессии аутентифицированного пользователя с соблюдением ролевой модели доступа.

Снимки экранов пользовательского интерфейса представлены на рисунках 2, 3, 4.

Мои обращения Уведомления

Мои обращения

Вы вошли как исследователь

Создать обращение

Поиск по тексту

Статус

Тип обращения

29/04/2025

№	Тип обращения	Статус	Дата создания	Последнее обновление	Действия
#12345	Ошибка	В обработке	25.04.2025	27.04.2025	Просмотр
#12346	Предложение	Новое	26.04.2025	26.04.2025	Просмотр

Рис. 2. Список обращений

[Мои обращения](#) [Уведомления](#)

Просмотр обращения

 Вы вошли как исследователь

Основная информация

Номер обращения: #12345

Тип: Ошибка

Статус: В обработке

Дата создания: 25.04.2025

Последнее обновление: 27.04.2025

Описание проблемы

Не загружается страница после ввода данных в форме исследования.

Вложения

[Скриншот ошибки](#)

Переписка

Служба поддержки: Пожалуйста, уточните браузер и версию системы.

Отправить ответ

Рис. 3. Карточка обращения

[Мои обращения](#) [Уведомления](#)

Уведомления

 Вы вошли как исследователь

Ваше обращение #12345 получило новый ответ от службы поддержки.
27.04.2025

Ваше обращение #12346 изменило статус на "В обработке".
26.04.2025

Рис. 4. Список уведомлений

Доступ к функциям взаимодействия ограничен авторизованными пользователями; в зависимости от уровня прав возможно как создание собственных обращений, так и просмотр обращений, созданных членами группы (например, в рамках одного исследовательского проекта).

Все действия пользователя в интерфейсе фиксируются в журнале аудита с привязкой к идентификатору пользователя, временной метке и контексту выполнения. При работе с конфиденциальной информацией обеспечивается шифрование передаваемых данных, а также маскирование персональных данных в пользовательском интерфейсе, за исключением случаев, когда доступ к ним предусмотрен ролью.

Механизмы защиты от недобросовестного использования включают ограничение частоты обращений (rate-limiting), а также фильтрацию вложений по типу и объему [16].

2.4. Административный модуль поддержки

Для сотрудников технической поддержки разработан отдельный административный интерфейс. Основной задачей модуля является обеспечение эффективной обработки обращений, их маршрутизации, ведения переписки с пользователями и анализа текущей эксплуатационной ситуации. Проектирование административного модуля было ориентировано на высокую степень управляемости и отслеживаемости процессов.

Реализован следующий функционал:

1. Просмотр и фильтрация обращений: отображение обращений в табличной форме с возможностью фильтрации по различным параметрам: статусу, дате создания, критичности, категории, исполнителю, ключевым словам в описании.
2. Категоризация и переклассификация: по результатам анализа обращения возможно изменение изначально заданной категории и уровня критичности, что влияет на статистическую отчетность и правила эскалации.
3. Изменение статуса: в процессе работы оператор изменяет статус обращения (например, «в обработке», «ожидает дополнительной информации», «решено», «закрыто»), что отражается в пользовательском интерфейсе и аналитических отчетах.
4. Ведение переписки: оператор может добавлять комментарии к обращению, направлять пользователю уточняющие вопросы и уведомления. Переписка ведется в связанной форме, обеспечивая контекстное отображение истории взаимодействия.
5. Назначение ответственного: каждое обращение может быть вручную или автоматически назначено на конкретного сотрудника поддержки. Назначение фиксируется в журнале аудита.
6. Экспорт и отчетность: предусмотрена возможность формирования сводных отчетов по обращению, выгрузка данных в формате CSV и формирование управленческих аналитических срезов.

Снимки экранов интерфейса поддержки представлены на рисунках 5, 6, 7.

Каждое действие оператора фиксируется в системе аудита с указанием времени, пользователя и типа операции. Доступ к административному модулю строго ограничен ролевой моделью доступа, при этом все действия в модуле контролируются механизмами журналирования.

Для предотвращения ошибок и злоупотреблений реализована система обязательного подтверждения критически значимых операций, таких как удаление или закрытие обращений без ответа пользователя. Кроме того, предусмотрены автоматические оповещения при попытках массовой модификации данных или при нарушении нормальных рабочих шаблонов (например, превышение времени бездействия по обращению).

[Обращения](#) [Аналитика](#) [Настройки](#)

Настройки системы обращений

Вы вошли как сотрудник поддержки

Параметры обработки обращений

Максимальное время ответа (часы):

Критические категории обращений:

Ошибка

Максимальный размер вложения (МБ):

Сохранить настройки

Рис. 5. Настройки системы обращений

[Обращения](#) [Аналитика](#) [Настройки](#)

Обращения пользователей

Вы вошли как сотрудник поддержки

Поиск по тексту

Статус

Тип обращения

29/04/2025

№	Пользователь	Тип обращения	Статус	Дата создания	Последнее обновление	Действия
#12345	Иванов И.И.	Ошибка	В обработке	25.04.2025	27.04.2025	Просмотр
#12346	Петрова А.А.	Предложение	Новое	26.04.2025	26.04.2025	Просмотр

Рис. 6. Список обращений пользователей

[Обращения](#) [Аналитика](#) [Настройки](#)

Аналитика обращений

Вы вошли как сотрудник поддержки

Сводная статистика

Всего обращений: 124

Открытые обращения: 32

Среднее время обработки: 2ч 15м

Количество закрытых обращений за месяц: 92

Распределение по типам

Ошибки: 65%

Предложения: 25%

Вопросы: 10%

Рис. 7. Статистика обращений

2.5. Модуль аналитики и мониторинга

Модуль аналитики и мониторинга предназначен для автоматизированного сбора, хранения, обработки и визуализации эксплуатационных метрик, характеризующих состояние подсистемы обработки обращений. Его основная задача — обеспечение постоянного контроля за ключевыми параметрами функционирования системы, оперативное обнаружение отклонений от нормы и предоставление информации для принятия инженерных и управленческих решений.

В качестве базовой платформы для сбора и хранения метрик используется система Prometheus — специализированное хранилище временных рядов, обладающее высокой производительностью, встроенным языком запросов (PromQL) и возможностью создания правил оповещения [17, 18]. Для визуального представления агрегированных данных используется Grafana — система построения визуальных панелей, предоставляющая широкие возможности по фильтрации, сравнительному анализу и отображению показателей в реальном времени [19].

Каждый компонент системы (включая веб-приложение, очередь сообщений, обработчики обращений, базу данных и административный модуль) содержит собственный экспортер метрик, реализованный в виде HTTP-интерфейса. Экспортеры собирают и предоставляют в централизованное хранилище данные о следующих характеристиках:

1. Нагрузочные характеристики: количество обращений в единицу времени, среднее время обработки, распределение обращений по категориям, приоритетам и источникам.
2. Показатели надежности: количество системных ошибок, доля повторных обращений, среднее время реакции и полного разрешения обращения.
3. Информационно-поведенческие характеристики: активность пользователей, средняя длина переписки по обращению, количество обращений, не получивших ответа в течение контрольного периода.
4. Состояние очередей и рабочих процессов: длина очереди, число активных обработчиков, задержки между регистрацией и началом обработки.
5. Все метрики агрегируются с указанием временной метки, идентификатора компонента, а также уникального контекста выполнения, что обеспечивает высокую степень детализированности анализа.

В целях оперативного реагирования на отклонения от нормативных значений реализована система автоматического формирования оповещений, основанная на правилах, определяемых в языке запросов PromQL [18]. К числу пороговых условий относятся:

- превышение количества новых обращений выше установленного значения за фиксированный интервал (больше 1 жалобы от пользователей в час);
- рост среднего времени ответа на обращения более допустимого порога (больше 2 часов);
- превышение предельного количества системных ошибок на единицу времени (больше 5 исключений в минуту);
- отсутствие активности обработчиков очередей при наличии необработанных событий.

При срабатывании одного или нескольких условий формируется уведомление, которое направляется в утвержденные каналы взаимодействия (в данный момент реализовано уведомление по почте, но возможно дальнейшее расширение списка провайдеров, например уведомление в Telegram, Slack и т. д.). Таким образом, обеспечивается не только

пассивное наблюдение за состоянием системы, но и активное вмешательство в режиме реального времени.

3. ИНЖЕНЕРНЫЕ ЭКСПЕРИМЕНТЫ

С целью верификации архитектурных решений, оценки эксплуатационных характеристик и определения пределов масштабируемости подсистемы обработки обращений был проведен цикл инженерных экспериментов. Экспериментальная методология опиралась на принцип искусственной нагрузки с варьированием параметров среды и сценариев использования, приближенных к реальным эксплуатационным условиям.

Основными целями экспериментов являлись:

- оценка устойчивости подсистемы при увеличении числа параллельных обращений, имитирующих массовую активность пользователей и автоматических агентов;
- анализ времени отклика и латентности обработки в зависимости от длины очереди и количества активных рабочих процессов;
- проверка поведения системы при сбоях отдельных компонентов, включая отключение брокера сообщений, баз данных и отказов интерфейсных узлов;
- определение предельных значений производительности, при которых наблюдается деградация сервисов.

Эксперименты проводились в контролируемой среде на стенде, идентичном продуктивной конфигурации системы. Для генерации нагрузки применялся программный комплекс имитационного тестирования Apache JMeter, генерирующий обращения по REST-интерфейсу с регулируемой интенсивностью [20].

Параметры нагрузки включали:

- число параллельных клиентов: от 50 до 2000 симулированных пользователей;
- частота генерации обращений: от 1 до 100 обращений в секунду;
- соотношение типов обращений: 70 % пользовательские, 20 % автоматические (ошибки), 10 % обращения от службы поддержки;
- продолжительность тестов: от 10 до 60 минут на каждый сценарий.

В процессе испытаний фиксировались следующие метрики:

- среднее и максимальное время регистрации обращения;
- среднее время постановки обращения в очередь;
- загрузка и время реакции рабочих процессов;
- частота ошибок (в том числе HTTP 5xx, ошибки брокера, задержки логирования).

Эксперименты показали, что при числе активных обращений до 10000 и частоте до 60 обращений в секунду система сохраняет стабильность: среднее время регистрации составляет менее 250 мс, а задержка постановки в очередь не превышает 100 мс. Обработка очередей остается устойчивой при условии не менее 8 параллельных обработчиков.

В условиях отказа одного из узлов брокера сообщений была зафиксирована временная задержка порядка 10 секунд с последующим восстановлением потока сообщений и повторной обработкой с сохранением порядка. При отключении одного из приложений, обрабатывающих обращения, оставшиеся экземпляры корректно перераспределили нагрузку.

При достижении критической нагрузки (>100 обращений в секунду) наблюдалось значительное увеличение времени отклика интерфейса (до 5 секунд) и задержки в логировании (до 8 секунд). Система продолжала функционировать, однако требовала внешнего масштабирования.

Эксперименты показали, что на данный момент имеющейся конфигурации вполне достаточно для функционирования системы. Результаты испытаний подтвердили, что реализованная архитектура подсистемы демонстрирует высокую устойчивость к нагрузке и отказам, соответствующую требованиям систем медицинского назначения. Были зафиксированы зоны потенциального роста: необходимость динамического масштабирования обработчиков и вынесения операций логирования в асинхронный канал с буферизацией.

4. ОБСУЖДЕНИЕ И ПЕРСПЕКТИВЫ РАЗВИТИЯ

Разработка и внедрение подсистемы обработки обращений с обратной связью в рамках веб-ориентированной системы медицинских исследований продемонстрировали высокую эффективность предложенной архитектурной модели на этапе эксплуатационных испытаний. Реализованная интеграция механизмов регистрации обращений, их асинхронной обработки, мониторинга и аналитики позволила обеспечить устойчивость системы к возрастанию нагрузки и оперативное реагирование на аномальные ситуации.

Тем не менее, текущая версия подсистемы обладает рядом ограничений. В частности, классификация обращений осуществляется на основе заранее заданных эвристических правил, что снижает адаптивность к вариативности пользовательских формулировок [21]. Кроме того, рост объема данных в перспективе потребует дополнительной оптимизации хранения журналов и метрик, а также введения иерархических моделей категоризации обращений.

В качестве направлений дальнейшего развития предполагается:

1. Внедрение алгоритмов машинного обучения для автоматической классификации обращений, с последующим активным дообучением моделей на верифицированных данных.
2. Интеграция подсистемы обработки обращений с внешними сервисами управления инцидентами и поддержкой (например, ServiceNow, Jira Service Management) [13].
3. Реализация механизмов предиктивной аналитики на основе временных рядов, позволяющих прогнозировать пики нагрузки и заранее инициировать масштабирование ресурсов [21].
4. Расширение функциональности системы оповещений, включая построение комплексных сценариев эскалации в зависимости от многомерных критериев риска [22].

Таким образом, дальнейшие исследования будут направлены на повышение интеллектуальности, предиктивности и автономности системы в рамках единой платформы управления данными медицинских исследований.

5. ЗАКЛЮЧЕНИЕ

В рамках настоящего исследования была разработана и интегрирована в существующую архитектуру веб-ориентированной медицинской платформы подсистема обработ-

ки обращений, обеспечивающая полный цикл работы с инцидентами и пользовательской обратной связью. Описана архитектурная структура системы, принципы ее функционирования, а также механизм взаимодействия между компонентами в условиях высокой нагрузки.

Тем самым устранена ранее идентифицированная уязвимость — отсутствие формализованного централизованного канала взаимодействия для врачей-исследователей, влияющее на их удовлетворенность системой и готовность применять ее в практике.

Проведенные инженерные эксперименты подтвердили устойчивость системы к эксплуатационным нагрузкам, ее способность сохранять работоспособность при частичных отказах инфраструктуры, а также высокую эффективность механизмов автоматического мониторинга и оповещения. Разработанные решения обеспечивают не только улучшение качества пользовательского опыта, но и повышение общей надежности и управляемости платформы.

Формализованный механизм маршрутизации и эскалации обращений позволит сократить среднее время реакции службы поддержки, тем самым минимизируя риск латентного распространения инцидентов вне регламентированных коммуникационных путей и повышая удовлетворенность пользователей системой.

Полученные результаты могут быть использованы в дальнейших проектах, направленных на разработку систем управления медицинскими данными с поддержкой обратной связи и автоматической адаптацией к меняющимся условиям эксплуатации.

Список литературы

1. Pfeiffer R.-H., Aaen J. Tools for monitoring software quality in information systems development and maintenance: five key challenges and a design proposal // *International Journal of Information Systems and Project Management*. 2024. Vol. 12, № 1. P. 19–40. doi:10.12821/ijispm120102
2. Ferreira F. H. C., Nakagawa E. Y., Bertolino A., Lonetti F., Neves V. de O., dos Santos R. P. A framework for the design of fault-tolerant systems-of-systems // *Journal of Systems and Software*, 2024. Vol. 211. P. 112010. doi:10.1016/j.jss.2024.112010
3. Watson C. H., Masalonis A., Arnold T., Chumbler N. R., Plew W. Methods and Lessons Learned from a Current State Workflow Assessment following Transition to a New Electronic Health Record System // *Perspectives in health information management*. 2023. Vol. 20, № 2. P. 1–10.
4. Arabi Y. M., Al Ghamdi A. A., Al-Moamary M., Al Mutrafy A., AlHazme R. H., Al Knawy B. A. Electronic medical record implementation in a large healthcare system from a leadership perspective // *BioMed Central Medical Informatics and Decision Making*. 2022. Vol. 22, № 1. P. 1–10. doi:10.1186/s12911-022-01801-0
5. Boonstra A., Broekhuis M. Barriers to the acceptance of electronic medical records by physicians from systematic review to taxonomy and interventions // *BMC Health Services Research*. 2010. Vol. 10, № 1. P. 1–17. doi:10.1186/1472-6963-10-231
6. Gentry M. T. et al. Clinician Satisfaction with Rapid Adoption and Implementation of Telehealth Services During the COVID-19 Pandemic // *Telemedicine and e-Health*. 2021. Vol. 27, № 12. P. 1385–1392. doi:10.1089/tmj.2020.0575
7. Kalankesh L. R., Nasiry Z., Fein R., Damanabi S. Factors Influencing User Satisfaction with Information Systems: A Systematic Review // *Galen Medical Journal*. 2020. Vol. e1686. P. 1–9. doi:10.31661/gmj.v9i0.1686
8. Castillo V. H., Martínez-García A. I., Pulido J. A knowledge-based taxonomy of critical factors for adopting electronic health record systems by physicians: a systematic literature review // *BMC Medical Informatics and Decision Making*. 2010. Vol. 10, № 1. P. 1–17. doi:10.1186/1472-6947-10-60
9. Топузов Э. Р., Ампилова Н. Б. Веб-ориентированная система для обработки данных медицинских исследований // *Компьютерные инструменты в образовании*. 2024. № 1. С. 45–57.

10. Dobbelaere P., Esmaili K. S. Kafka versus RabbitMQ // Proceedings of the 11th ACM International Conference on Distributed and Event-based Systems — DEBS '17. 2017. P. 227–238. doi:10.1145/3093742.3093908
11. Vinoski S. Advanced Message Queuing Protocol // IEEE Internet Computing. 2006. Vol. 10, № 6. P. 87–89. doi:10.1109/mic.2006.116
12. Thomas M. R., Pavlick E., Linzen T. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. 2019. P. 3428–3448. doi:10.48550/arxiv.1902.01007
13. Truss M., Boehm S. AI-based Classification of Customer Support Tickets: State of the Art and Implementation with AutoML // arXiv (Cornell University), P. 357–381, Jun. 2024,
14. Videla A., Williams J. RabbitMQ in Action. Shelter Island. NY, USA: Manning Publications, 2012.
15. Farias K., Lazzari L. Event-driven Architecture and REST Architectural Style: An Exploratory Study on Modularity // Journal of applied research and technology. 2023. Vol. 21, № 3, pp. 338–351. doi:10.22201/icat.24486736e.2023.21.3.1764
16. Serbout S., El Malki A., Pautasso C., Zdun U. API Rate Limit Adoption — A pattern collection // EuroPLOP '23: Proceedings of the 28th European Conference on Pattern Languages of Programs, 2023. P. 1–20. doi:10.1145/3628034.3628039
17. Liu Y., Yu Z., Wang Q., Mei H., Song G., Li H. Research on cloud-native monitoring system based on Prometheus // Fourth International Conference on Sensors and Information Technology (ICSI 2024), 2024. P. 308–315. doi:10.1117/12.3029320
18. Sabharwal N., Pandey P. Working with Prometheus Query Language (PromQL) // Monitoring Microservices and Containerized Applications: Deployment, Configuration, and Best Practices for Prometheus and Alert Manage. Berkeley, CA, USA: Apress, 2020. P. 141–167. doi:10.1007/978-1-4842-6216-0
19. Chakraborty M., Kundan A. P. Grafana // Monitoring cloud-native applications: Lead agile operations confidently using open source software. Berkeley, CA, USA: Apress, 2021. P. 187–240. doi:10.1007/978-1-4842-6888-9
20. Tiwari V., Upadhyay S., Goswami J. K., Agrawal S. Analytical Evaluation of Web Performance Testing Tools: Apache JMeter and SoapUI // 2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT). 2023. P. 519–523. doi:10.1109/csnt57126.2023.10134699
21. Ahmed S. et al. An Empirical Analysis of State-of-Art Classification Models in an IT Incident Severity Prediction Framework // Applied Sciences. 2023. Vol. 13, № 6. P. 1–27. doi:10.3390/app13063843
22. Qiu J., Lin Y., Zwetsloot I. M. An LSTM-based predictive monitoring method for data with time-varying variability // International Journal of Production Research, 2024. Vol. 63, № 7. P. 2622–2637. doi:10.1080/00207543.2024.2408435

Поступила в редакцию 20.03.2025, окончательный вариант — 15.04.2025.

Топузов Эдем Русланович, аспирант, Математико-механический факультет, СПбГУ,
✉ edem.topuzov@gmail.com

Computer tools in education, 2025

№ 1: 76–90

<http://cte.eltech.ru>

[doi:10.32603/2071-2340-2025-1-76-90](https://doi.org/10.32603/2071-2340-2025-1-76-90)

Integration of a Support System with Feedback into a Web-Based Medical Platform: Architecture, Implementation, and Engineering Experiments

Topuzov E. R.¹, Postgraduate, ✉ edem.topuzov@gmail.com, orcid.org/0009-0003-9208-7931

¹ Saint Petersburg State University, 28 Universitetskii pr., Peterhof, 198504, Saint Petersburg, Russia

Abstract

This paper presents the architecture and implementation of a feedback support subsystem integrated into a previously developed web-oriented platform for managing medical research data. The proposed solution enables both automatic and semi-automatic error registration, as well as structured interaction between users and the support team within a closed-loop quality improvement cycle. Additionally, the paper reports the results of engineering experiments conducted to assess the system's resilience and scalability.

Keywords: *ticketing system, support system, web-based system.*

Citation: E. R. Topuzov, "Integration of a Support System with Feedback into a Web-Based Medical Platform: Architecture, Implementation, and Engineering Experiments," *Computer tools in education*, no. 1, pp. 76–90, 2025 (in Russian); [doi:10.32603/2071-2340-2025-1-76-90](https://doi.org/10.32603/2071-2340-2025-1-76-90)

References

1. R.-H. Pfeiffer and J. Aaen, "Tools for monitoring software quality in information systems development and maintenance: five key challenges and a design proposal," *International Journal of Information Systems and Project Management*, vol. 12, no. 1, pp. 19–40, 2024, [doi:10.12821/ijispm120102](https://doi.org/10.12821/ijispm120102)
2. F. H. C. Ferreira et al., "A framework for the design of fault-tolerant systems-of-systems," *Journal of Systems and Software*, vol. 211, p. 112010, 2024; [doi:10.1016/j.jss.2024.112010](https://doi.org/10.1016/j.jss.2024.112010)
3. C. H. Watson et al., "Methods and Lessons Learned from a Current State Workflow Assessment following Transition to a New Electronic Health Record System," *Perspectives in health information management*, vol. 20, no. 2, pp. 1–10, 2023.
4. Y. M. Arabi et al., "Electronic medical record implementation in a large healthcare system from a leadership perspective," *BioMed Central Medical Informatics and Decision Making*, vol. 22, no. 1, pp. 1–10, 2022; [doi:10.1186/s12911-022-01801-0](https://doi.org/10.1186/s12911-022-01801-0).
5. A. Boonstra and M. Broekhuis, "Barriers to the acceptance of electronic medical records by physicians from systematic review to taxonomy and interventions," *BMC Health Services Research*, vol. 10, no. 1, pp. 1–17, 2010; [doi:10.1186/1472-6963-10-231](https://doi.org/10.1186/1472-6963-10-231)
6. M. T. Gentry et al., "Clinician Satisfaction with Rapid Adoption and Implementation of Telehealth Services During the COVID-19 Pandemic," *Telemedicine and e-Health*, vol. 27, no. 12, pp. 1385–1392, 2021; [doi:10.1089/tmj.2020.0575](https://doi.org/10.1089/tmj.2020.0575)
7. L. R. Kalankesh et al., "Factors Influencing User Satisfaction with Information Systems: A Systematic Review," *Galen Medical Journal*, vol. e1686, pp. 1–9, 2020; [doi:10.31661/gmj.v9i0.1686](https://doi.org/10.31661/gmj.v9i0.1686)

8. V. H. Castillo, A. I. Martínez-García, and J. Pulido, “A knowledge-based taxonomy of critical factors for adopting electronic health record systems by physicians: a systematic literature review,” *BMC Medical Informatics and Decision Making*, vol. 10, no. 1, pp. 1–17, 2010; doi:10.1186/1472-6947-10-60
9. E. Topuzov and N. Ampilova, “Web-based system for processing medical research data,” *Computer tools in education*, no. 1, pp. 45–57, 2024; doi:10.32603/2071-2340-2024-1-45-57 (In Russian).
10. P. Dobbelaere and K. S. Esmaili, “Kafka versus RabbitMQ,” in *Proc. of the 11th ACM International Conference on Distributed and Event-based Systems — DEBS '17*, pp. 227–238, 2017; doi:10.1145/3093742.3093908
11. S. Vinoski, “Advanced Message Queuing Protocol,” *IEEE Internet Computing*, vol. 10, no. 6, pp. 87–89, 2006; doi:10.1109/mic.2006.116
12. M. R. Thomas, E. Pavlick, and T. Linzen, “Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference,” pp. 3428–3448, 2019; *arXiv:1902.01007 [cs.CL]*
13. M. Truss and S. Boehm, “AI-based Classification of Customer Support Tickets: State of the Art and Implementation with AutoML,” pp. 357–381, 2024, *arXiv:2406.01789 [cs.LG]*
14. A. Videla and J. Williams, *RabbitMQ in Action*. Shelter Island, NY, USA: Manning Publications, 2012.
15. K. Farias and L. Lazzari, “Event-driven Architecture and REST Architectural Style: An Exploratory Study on Modularity,” *Journal of applied research and technology*, vol. 21, no. 3, pp. 338–351, 2023; doi:10.22201/icat.24486736e.2023.21.3.1764
16. S. Serbout et al., “API Rate Limit Adoption — A pattern collection,” in *Proc. of EuroPLoP '23: Proceedings of the 28th European Conference on Pattern Languages of Programs*, pp. 1–20, 2023; doi:10.1145/3628034.3628039
17. Y. Liu et al., “Research on cloud-native monitoring system based on Prometheus,” in *Proc. of the Fourth International Conference on Sensors and Information Technology (ICSI 2024)*, pp. 308–315, 2024; doi:10.1117/12.3029320
18. N. Sabharwal and P. Pandey, “Working with Prometheus Query Language (PromQL),” in *Monitoring Microservices and Containerized Applications: Deployment, Configuration, and Best Practices for Prometheus and Alert Manage*. Berkeley, CA, USA: Apress, pp. 141–167, 2020; doi:10.1007/978-1-4842-6216-0
19. M. Chakraborty and A.P. Kundan, “Grafana,” in *Monitoring cloud-native applications: Lead agile operations confidently using open source software*. Berkeley, CA, USA: Apress, pp. 187–240, 2021; doi:10.1007/978-1-4842-6888-9
20. V. Tiwari et al., “Analytical Evaluation of Web Performance Testing Tools: Apache JMeter and SoapUI,” in *Proc. of 2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 519–523, 2023; doi:10.1109/csnt57126.2023.10134699
21. S. Ahmed et al., “An Empirical Analysis of State-of-Art Classification Models in an IT Incident Severity Prediction Framework,” *Applied Sciences*, vol. 13, no. 6, pp. 1–27, 2023; doi: 10.3390/app13063843
22. J. Qiu, Y. Lin, and I. M. Zwetsloot, “An LSTM-based predictive monitoring method for data with time-varying variability,” *International Journal of Production Research*, vol. 63, no. 7, pp. 2622–2637, 2024; doi:10.1080/00207543.2024.2408435

Received 20-03-2025, the final version — 15-04-2025.

Edem Topuzov, Postgraduate, Faculty of Mathematics and Mechanics, SPbSU,
 ✉ edem.topuzov@gmail.com