

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

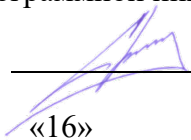
Департамент программной инженерии


СОГЛАСОВАНО

УТВЕРЖДАЮ

Научный руководитель,
доцент департамента
программной инженерии, канд. пед. наук

Академический руководитель
образовательной программы
«Программная инженерия»

 С. А. Виденин
«16» _____ мая _____ 2025 г.

 Н. А. Павлов
«16» _____ мая _____ 2025 г.

ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ
ИНЦИДЕНТАМИ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ


RU.17701729.05.05-01 ПМИ 01-1-ЛУ

Исполнители:


Студент группы БПИ-216

 А.И. Шарипов /

Студент группы БПИ-216

 / С. А. Пономарев/

Студент группы БПИ-214

 / Е. К. Фортов/

«16» мая 2025

Москва 2025

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

УТВЕРЖДЕН

RU.17701729.05.05-01 ПМИ 01-1-ЛУ

**ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ
ИНЦИДЕНТАМИ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ****Программа и методика испытаний****RU.17701729.05.05-01 ПМИ 01-1****Листов 29**

Инв. № полл.	Полп. и дата	Взам. Инв. №	Инв. № лубл.	Полп. и дата

СОДЕРЖАНИЕ

1. ОБЪЕКТ ИСПЫТАНИЙ.....	2
1.1. Наименование программы.....	2
1.2. Краткая характеристика области применения.....	2
2. ЦЕЛЬ ИСПЫТАНИЙ.....	3
3. ТРЕБОВАНИЯ К ПРОГРАММЕ.....	4
3.1 Состав выполняемых функций.....	4
3.2 Требования к временным характеристикам.....	10
3.3 Требования к интерфейсу.....	10
3.4 Требования к надежности.....	10
4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	11
4.1 Состав программной документации.....	11
4.2 Специальные требования к программной документации.....	12
5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ.....	13
5.1 Технические средства, используемые во время испытаний.....	13
5.2 Технические средства, используемые во время испытаний.....	13
5.3 Порядок проведения испытаний.....	13
6. МЕТОДЫ ИСПЫТАНИЙ.....	14
6.1 Проверка требований к программной документации.....	14
6.2 Проверка требований к программному интерфейсу.....	14
6.3 Проверка требований к функциональным характеристикам.....	14
6.3.1 Функционал регистрации и аутентификации.....	14
6.3.2 Функционал работы с дежурствами.....	15
6.3.3 Функционал работы с датчиками.....	17
6.3.4 Функционал работы со статусами.....	18
6.3.5 Функционал работы с графами статусов.....	19
6.3.6 Функционал работы с очередями (процессами) и тикетами.....	22
СПИСОК ИСТОЧНИКОВ.....	26
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	28

1. ОБЪЕКТ ИСПЫТАНИЙ

1.1. Наименование программы

Наименование программы – «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности».

Наименование программы на английском языке – «Information and Analytical Incident Management System in the Field of Security».

1.2. Краткая характеристика области применения

Система «Аргус» предназначена для автоматизации управления инцидентами на промышленных и коммерческих предприятиях, где критически важно минимизировать простои и оперативно реагировать на внештатные ситуации, такие как поломки оборудования, срабатывание пожарных датчиков, утечки или иные аварийные события. Она объединяет мониторинг физических датчиков с гибкой настройкой бизнес-процессов, позволяя не только фиксировать инциденты, но и автоматически запускать протоколы их устранения. При срабатывании датчика система создает тикет, назначает ответственного через ротацию дежурств и контролирует выполнение задач по настраиваемым графам статусов, обеспечивая прозрачность каждого этапа — от момента обнаружения проблемы до её решения.

2. ЦЕЛЬ ИСПЫТАНИЙ

Основная цель испытаний – проверка заявленных в техническом задании требований к выполняемым функциям, а также программным документам. Также проверка надежности и корректности работы программы, а также соответствие функционала заявленному в техническом задании.

3. ТРЕБОВАНИЯ К ПРОГРАММЕ

3.1 Состав выполняемых функций

Все функции реализуются через взаимодействие клиентской части с сервером (REST API).

Основные технологии клиента: React.js, TypeScript, Redux Toolkit.

Основные технологии сервера: ASP.NET, Java.

3.1.1.1. Регистрация

3.1.1.1.1. Пользователь регистрируется, указывая:

- Email (валидация формата, уникальность);
- Пароль (не менее 8 символов, обязательное наличие цифр и букв);
- Имя и фамилия (не менее 2 символов каждое).

3.1.1.1.2 После успешной регистрации:

- Запись о новом пользователе появляется в базе данных
- Автоматический редирект на главную страницу

3.1.1.1.3 Форма регистрации должна содержать:

- Поле ввода email с валидацией в реальном времени
- Поле ввода пароля с индикатором силы пароля
- Поле подтверждения пароля
- Поле ввода имени
- Поле ввода фамилии
- Кнопка "Зарегистрироваться"
- Ссылка на форму входа

3.1.1.1.4 Валидация:

- Подсветка полей красным при ошибке
- Текстовые подсказки об ошибках под полями
- Кнопка регистрации неактивна, пока форма невалидна

3.1.1.2. Аутентификация

3.1.1.2.1. Аутентификация по email и паролю (JWT-токен возвращается в ответе).

3.1.1.2.2 Пользователь, не прошедший аутентификацию, не допускается до страниц, не являющихся страницами регистрации и аутентификации. Такой пользователь не может отправлять запросы, кроме регистрации и аутентификации.

3.1.1.3. Управление тикетами

3.1.1.3.1. Интерфейс управления тикетами

Список тикетов:

- Таблица с колонками: ID, Название, Статус, Приоритет, Автор, Ответственный, Дата создания
- Сортировка по любой колонке
- Фильтрация по статусу и приоритету
- Пагинация (20 тикетов на страницу)
- Кнопка создания нового тикета

3.1.1.3.2. Форма создания/редактирования тикета:

- Модальное окно
- Поля:
 - Название (текстовое поле)
 - Описание (текстовая область)
 - Статус (выпадающий список)
 - Приоритет (выпадающий список)
 - Ответственный (выпадающий список с поиском)
- Кнопки: "Сохранить", "Отмена"
- Валидация полей в реальном времени

3.1.1.3.3. Форма создания/редактирования тикета:

- Верхняя панель с основной информацией
- Вкладки:
 - "Информация" - основные данные тикета
 - "Комментарии" - история комментариев

3.1.1.4. Управление очередями (процессами)

3.1.1.4.1. Создание очереди с параметрами:

- Название (уникальное);
- Максимальное количество тикетов (по умолчанию — 50);
- Приоритетные правила (например, "Высокий приоритет — в начало очереди").

3.1.1.4.2. Перемещение тикетов внутри очереди — drag-and-drop интерфейс.

3.1.1.4.3. Автоматическое обновление очереди при изменении статуса тикета.

3.1.1.4.4. Форма создания/редактирования очереди:

- Модальное окно
- Поля:
 - Название очереди
 - Максимальное количество тикетов
 - Правила приоритетов (динамический список)

3.1.1.4.5. Форма создания/редактирования очереди:

- Kanban-доска с колонками по статусам
- Drag-and-drop функционал для тикетов
- Фильтры (в рамках post-mvp):
 - По приоритету
 - По ответственным
 - По дате создания
- Счетчик тикетов в очереди
- Индикатор заполненности очереди

3.1.1.5. Управление датчиками

4.1.1.5.1 Пользователь может добавлять новые датчики, указывая:

- Название тикета, который датчик создаёт

- Описание тикета
- Дедлайн тикета
- Приоритет тикета
- Бизнес-процесс, к которому датчик будет привязан

3.1.1.5.2. Пользователь может редактировать в уже существующих датчиках поля

3.1.1.5.3 Интерфейс управления датчиками:

- Список датчиков:
 - Таблица с колонками: Название, Бизнес-процесс, Шаблон тикета, Статус
 - Фильтрация по бизнес-процессам
 - Поиск по названию
 - Кнопка создания нового датчика
- Форма создания/редактирования датчика:
 - Модальное окно с вкладками:
 - "Основная информация"
 - "Шаблон тикета"
 - "Настройки бизнес-процесса"
 - Поля:
 - Название датчика
 - Описание
 - Шаблон тикета (все поля из структуры тикета)
 - Выбор бизнес-процесса

3.1.1.6. Статусы и графы статусов

3.1.1.6.1 Пользователь может добавлять новые статусы

3.1.1.6.2. Пользователь может редактировать в уже существующих статусах следующие поля:

3.1.1.6.1 Название статуса

3.1.1.6.2 Описание статуса

3.1.1.6.3 Тип уведомления при статусе

3.1.1.6.4 Дежурство, в котором фигурирует статус

3.1.1.6.3 Пользователь может добавлять новые графы статусов путём рисования графа в редакторе (4.1.1.6.6.)

3.1.1.6.4. Пользователь может редактировать в уже существующих графах статусов следующие поля:

- Название графа
- Описание графа

3.1.1.6.5.Список статусов:

- Таблица с колонками: Название, Описание, Тип уведомления, Дежурство
- Кнопки редактирования и удаления для каждого статуса
- Кнопка создания нового статуса

3.1.1.6.6. Редактор графов статусов:

- “Холст” для рисования графа
- Панель инструментов:
 - Добавление узла
 - Добавление связи
 - Выбор элемента
 - Удаление элемента
- Свойства выбранного элемента:
 - Название
 - Описание
 - Тип уведомления
 - Настройки перехода
- Возможность сохранения графа как шаблона (post-mvp)

3.1.7. Дежурства

3.1.1.7.1 Пользователь может добавлять новые дежурства

3.1.1.7.2 Пользователь может настраивать дежурства:

3.1.1.7.2.1 Пользователь может редактировать дежурных

3.1.1.7.2.2 Пользователь настраивать смены внутри дежурств

3.1.1.7.2.3 Форма управления дежурством:

- Модальное окно
- Поля:
 - Название дежурства
 - Описание
 - Список дежурных (с возможностью поиска по пользователям)
 - Настройка смен (временные интервалы)
- Визуализация расписания смен (post-mvp)
- Конфликты с другими дежурствами подсвечиваются (post-mvp)

3.1.1.8. Поиск

3.1.1.8.1 Должна быть возможность осуществлять поиск по ключевым полям (именам) следующих объектов:

3.1.1.8.1.1. Поиск очередей

3.1.1.8.1.2. Поиск пользователей

3.1.1.8.1.3. Поиск дежурств

3.1.1.8.1.4. Поиск датчиков

4.1.1.8.2. Реализация алгоритмов поиска (фильтрации) предоставляется на усмотрение разработчиков.

3.1.1.9. Общие требования

3.1.1.9.1 Навигация:

- Боковое меню с основными разделами

- Хлебные крошки
- Быстрый доступ к часто используемым функциям

3.1.1.9.2 Отзывчивость:

- Адаптивный дизайн (мобильная версия) (post-mvp)
- Индикаторы загрузки
- Обратная связь при действиях пользователя

3.1.1.9.3 Уведомления (post-mvp):

- Toast-уведомления о результатах операций
- Система оповещений о важных событиях
- Центр уведомлений

3.1.1.9.4 Тематизация:

- Светлая и темная темы
- Настраиваемые цвета для статусов и приоритетов (post-mvp)

3.2 Требования к временным характеристикам

Совокупное время загрузки страниц приложения и отклика сервера не должно превышать 7 секунд.

3.3 Требования к интерфейсу

Интерфейс должен быть удобным для администрирования и интуитивно понятным. Дизайнерские решения остаются на усмотрение разработчика.

3.4 Требования к надежности

Приложение не должно завершаться аварийно при любом наборе входных данных. Приложение должно обеспечивать проверку входных данных на соответствие требуемому формату (см. раздел 3.1).

Приложение должно обеспечивать сохранность данных пользователей посредством средств криптографии и соблюдением мер по сохранности данных.

4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

4.1 Состав программной документации

- 1) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности (клиентская часть)». Техническое задание (ГОСТ 19.201-78) [7];
- 2) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности (клиентская часть)». Программа и методика испытаний (ГОСТ 19.301-78) [10];
- 3) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности (клиентская часть)». Руководство оператора (ГОСТ 19.505-79) [12];
- 4) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности (клиентская часть)». Текст программы (ГОСТ 19.401-78) [13].
- 5) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: процессы интеграции, обработки данных и управления». Техническое задание (ГОСТ 19.201-78) [7];
- 6) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: процессы интеграции, обработки данных и управления». Программа и методика испытаний (ГОСТ 19.301-78) [10];
- 7) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: процессы интеграции, обработки данных и управления». Руководство оператора (ГОСТ 19.505-79) [12];
- 8) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: процессы интеграции, обработки данных и управления». Текст программы (ГОСТ 19.401-78) [13];

- 9) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.». Техническое задание (ГОСТ 19.201-78) [7];
- 10) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.». Программа и методика испытаний (ГОСТ 19.301-78) [10];
- 11) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.». Руководство оператора (ГОСТ 19.505-79) [12];
- 12) «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.». Текст программы (ГОСТ 19.401-78) [13];

4.2 Специальные требования к программной документации

- 1) Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1).
- 2) Пояснительная записка должна быть загружена в систему Антиплагиат через LMS «НИУ ВШЭ». Лист, подтверждающий загрузку пояснительной записки, сдается в учебный офис вместе со всеми материалами не позже, чем за день до защиты курсовой работы.
- 3) Вся документация также воспроизводится в печатном виде, она должна быть подписана академическим руководителем образовательной программы 09.03.04 «Программная инженерия», руководителем разработки и исполнителями перед сдачей курсовой работы в учебный офис, не позже одного дня до защиты.
- 4) Документация также сдается в электронном виде в формате .pdf или .docx, а программа – в архиве формата .zip или .rar.
- 5) Все документы перед защитой ВКР должны быть загружены в информационно-образовательную среду НИУ ВШЭ LMS (Learning Management System) в личном кабинете одним архивом

5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

5.1 Технические средства, используемые во время испытаний

Во время испытаний использовался персональный компьютер, оснащенный следующими техническими компонентами:

- Процессор M1 Pro
- Более 20 GB свободного дискового пространства
- 16GB ОЗУ

5.2 Технические средства, используемые во время испытаний

Во время проведения испытаний использовался следующий состав программных средств:

- Персональный компьютер с операционной MacOS [10] 14.4.1
- Установленная программа Docker с утилитой Docker-Compose для развертывания серверной части на локальном устройстве
- Утилита grpcurl [12]

5.3 Порядок проведения испытаний

Испытания проводятся по порядку, указанному далее:

1. Испытание проверки выполнения требований к программной документации.
2. Испытание проверки выполнения требований к программному интерфейсу.
3. Испытание проверки выполнения требований к функциональным характеристикам.

6. МЕТОДЫ ИСПЫТАНИЙ

6.1 Проверка требований к программной документации

Проверка выполнения требований к программной документации происходит посредством визуального ознакомления с документацией и проверки её на соответствие описанным требованиям. Также визуально проверяется соответствие документации требованиям ГОСТ.

6.2 Проверка требований к программному интерфейсу

Соответствие требованиям к программному интерфейсу было выполнено путём проведения юзабилити-тестов на интерактивных макетах в Figma (к интерфейсу не предъявлялось строго формализованных требований с точки зрения визуальной составляющей, основной потребностью пользователя является удобство пользования и интуитивность).

Из пяти принимавших участие в тестировании людей все пятеро справились с прохождением по пользовательским путям для пользования функционалом из требований. Интерфейс считаем достаточно удобным и интуитивно понятным.

6.3 Проверка требований к функциональным характеристикам

Для проверки функциональных характеристик программы на локальный компьютер был клонирован удаленный репозиторий с программой, локально развернуто приложение и далее проведена проверка по перечню функциональных требований.

6.3.1 Функционал регистрации и аутентификации

На экране аутентификации (рис. 1) ввести корректные логин и пароль пользователя, данные о котором имеются в базе данных системы. При успешной авторизации произойдёт переход на главный экран (дашборд). При неверно введённых данных высветится соответствующее уведомление и перехода не произойдёт.

При нажатии на кнопку «зарегистрироваться», произойдёт переход на экран регистрации. При введении на нём корректных данных, запись о новой учётной записи попадёт в систему и пользователь будет перенаправлен на экран с дашбордом.

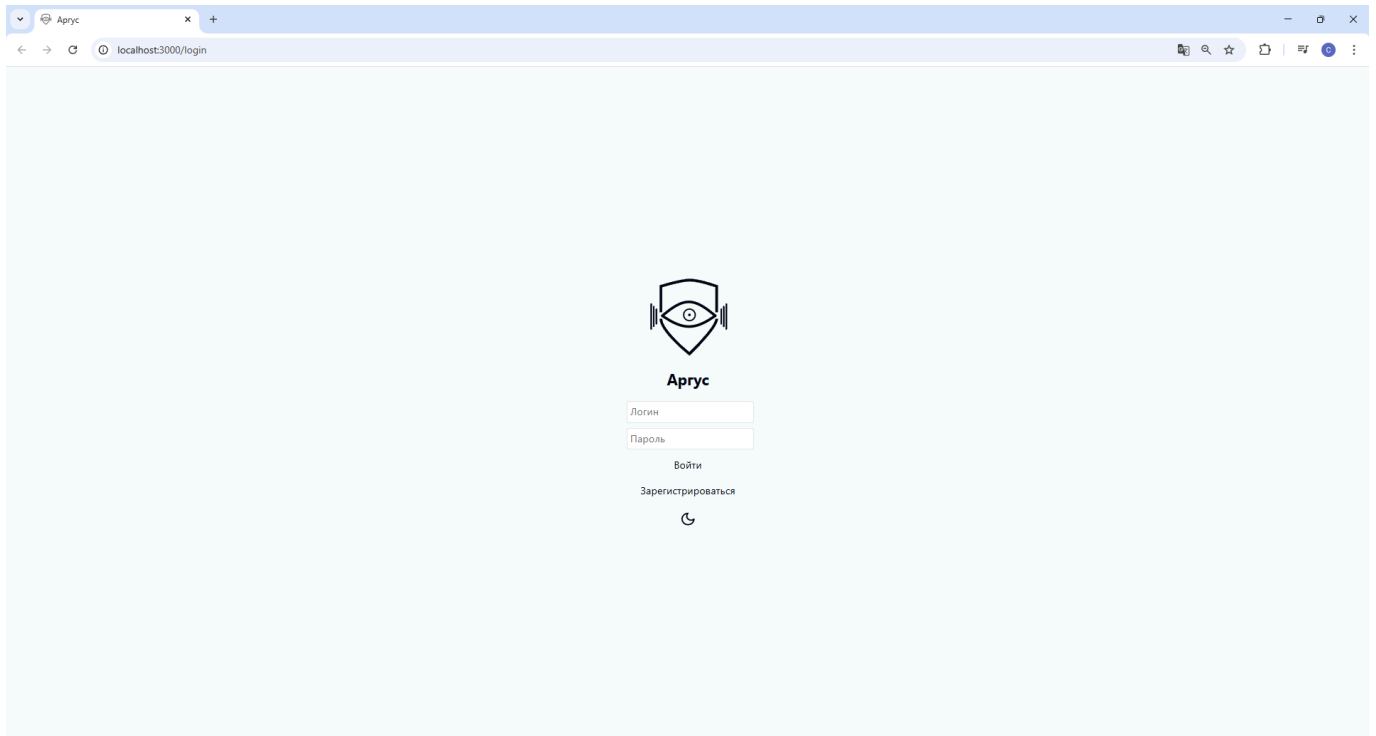


Рис 1. Экран авторизации

6.3.2 Функционал работы с дежурствами

При переходе на экран «дежурства» (рис. 2), начнётся загрузка доступных пользователю дежурств и отображение их в списке. На каждое дежурство можно нажать, чтобы посмотреть и редактировать информацию о нём.

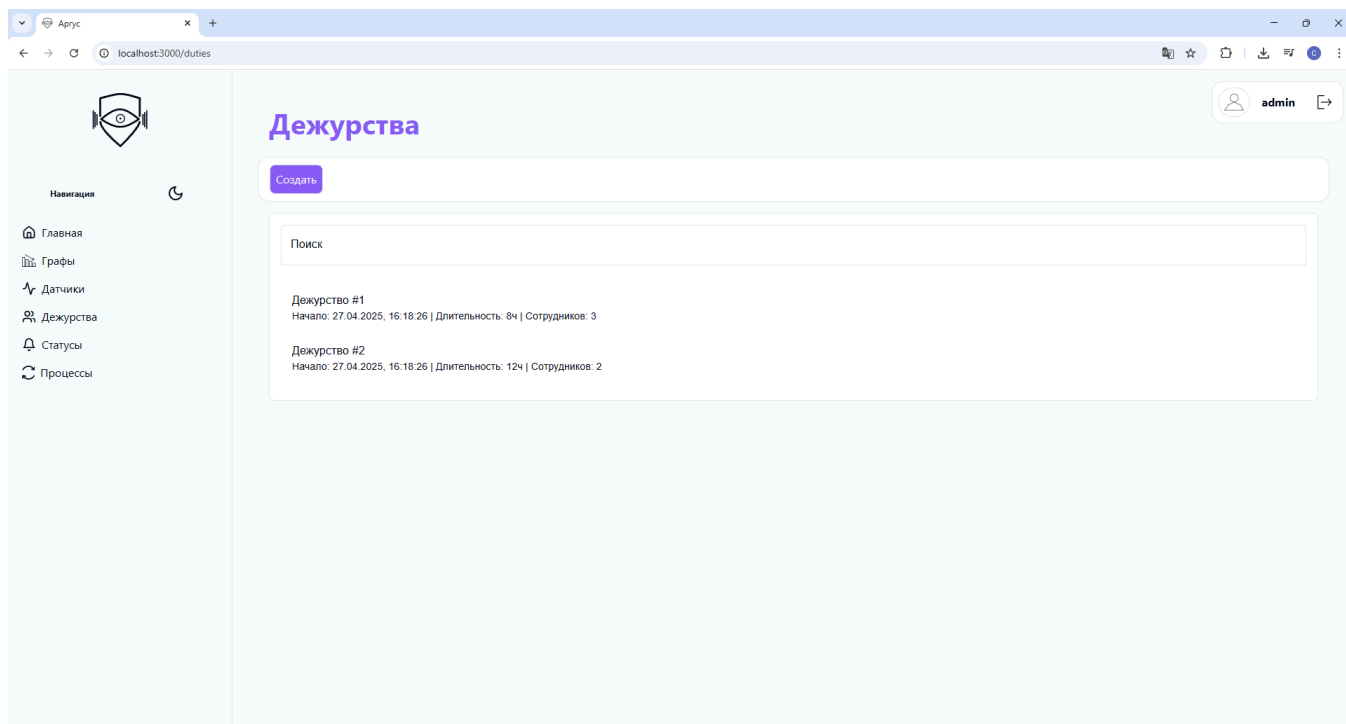


Рис 2. Экран дежурств

Создание дежурства проверяется следующим образом:

1. Перейти в раздел "Дежурства"
2. Нажать кнопку "Создать дежурство"
3. Заполнить раздел "Основная информация":
 - a. Ввести название
 - b. Ввести описание
4. В разделе "Дежурные":
 - a. Найти сотрудника через поиск
 - b. Добавить сотрудника в список
5. В разделе "Смены":
 - a. Выбрать даты в календаре
 - b. Установить время начала и окончания
 - c. Назначить дежурных
6. Нажать кнопку "Сохранить"

Ожидаемый результат: Дежурство создано и отображается в календаре.

Редактирование дежурства проверяется аналогичным образом, но нужно редактировать поля в уже существующем дежурстве.

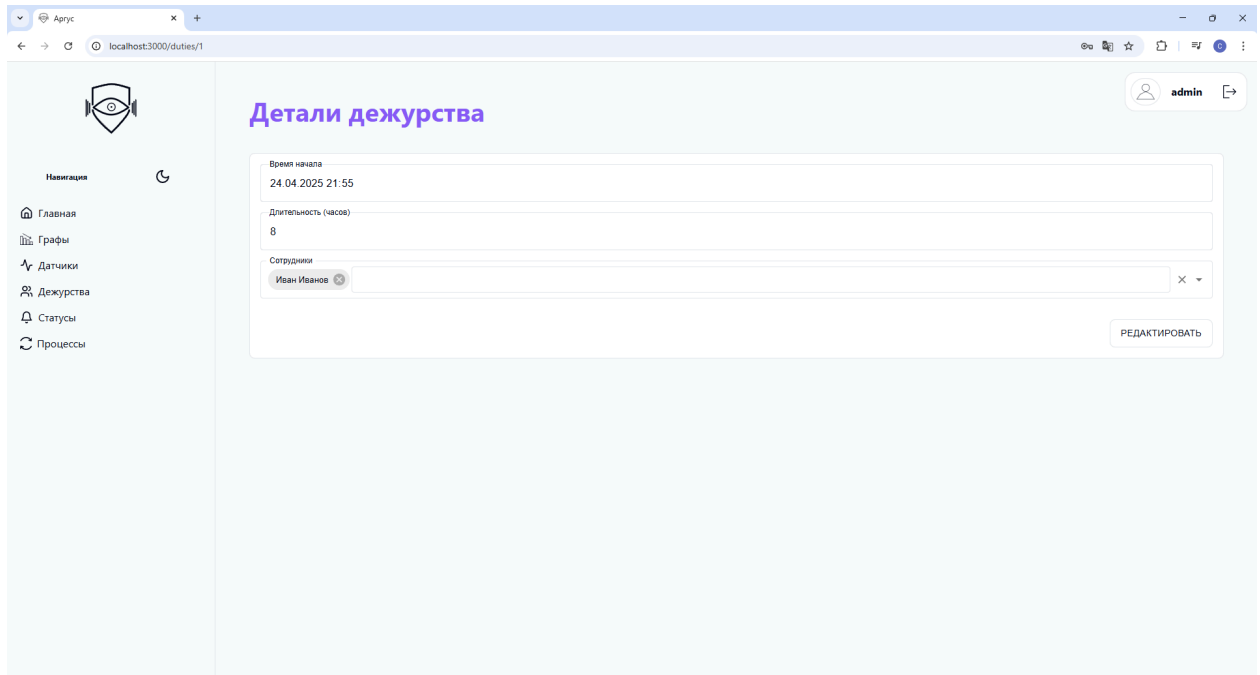


Рис 3. Создание/редактирование дежурства

6.3.3 Функционал работы с датчиками

При переходе на экран «датчики» (рис. 4), начнётся загрузка доступных пользователю датчиков и отображение их в списке. На каждый датчик можно нажать, чтобы посмотреть и редактировать информацию о нём.

Экраны просмотра остальных сущностей работают и выглядят идентично уже продемонстрированным, поэтому в дальнейшем их демонстрация отсутствует.

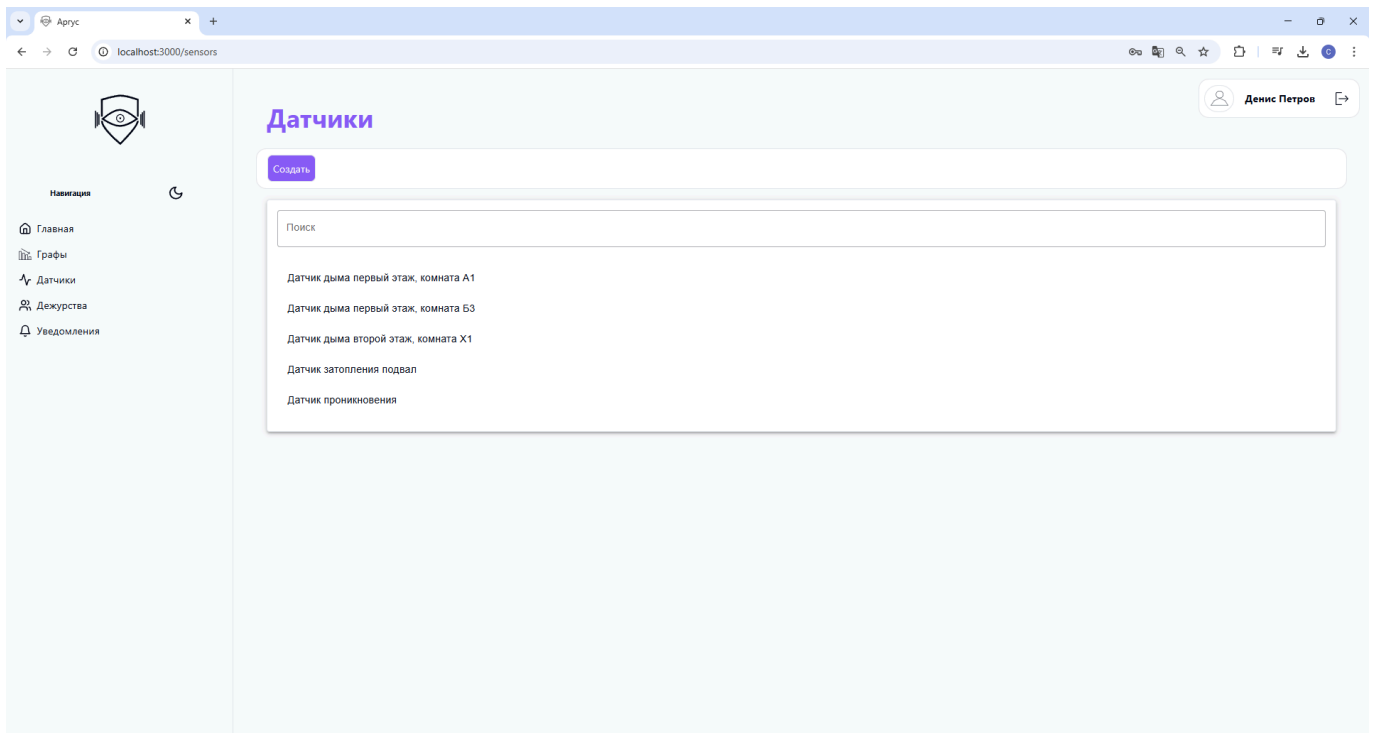


Рис 4. Экран датчиков

Создание датчика проверяется следующим образом:

1. Перейти в раздел "Датчики"
2. Нажать кнопку "Создать датчик"
3. Заполнить вкладку "Основная информация":
 - a. Ввести название
 - b. Ввести описание
4. Перейти на вкладку "Шаблон тикета":
 - a. Заполнить все поля
 - b. Выбрать дату дедлайна
 - c. Выбрать приоритет
5. Перейти на вкладку "Настройки бизнес-процесса":
 - a. Выбрать процесс
 - b. Настроить параметры
6. Нажать кнопку "Сохранить"

Ожидаемый результат: Датчик создан и отображается в списке.

Редактирование датчика проверяется аналогичным образом, но нужно редактировать поля в уже существующем датчике.

Создание датчика

Название

Описание

Приоритет
Средний

Срок решения (дней)
7

Бизнес-процесс

ОТМЕНА СОЗДАТЬ

Рис 5. Создание/редактирование датчика

6.3.4 Функционал работы со статусами

При переходе на экран «статусы», начнётся загрузка доступных пользователю статусов и отображение их в списке. На каждый статус можно нажать, чтобы посмотреть и редактировать информацию о нём.

Создание статуса проверяется следующим образом:

1. Перейти в раздел "Статусы"
2. Нажать кнопку "Создать статус"
3. В открывшейся форме:
 - a. Ввести название (менее 3 символов)
 - b. Проверить появление ошибки
 - c. Ввести корректное название
 - d. Заполнить описание
 - e. Выбрать тип уведомления
 - f. Выбрать дежурство
4. Нажать кнопку "Сохранить"

Ожидаемый результат: Статус создан и отображается в таблице.

Редактирование статуса проверяется аналогичным образом, но нужно редактировать поля в уже существующем статусе.

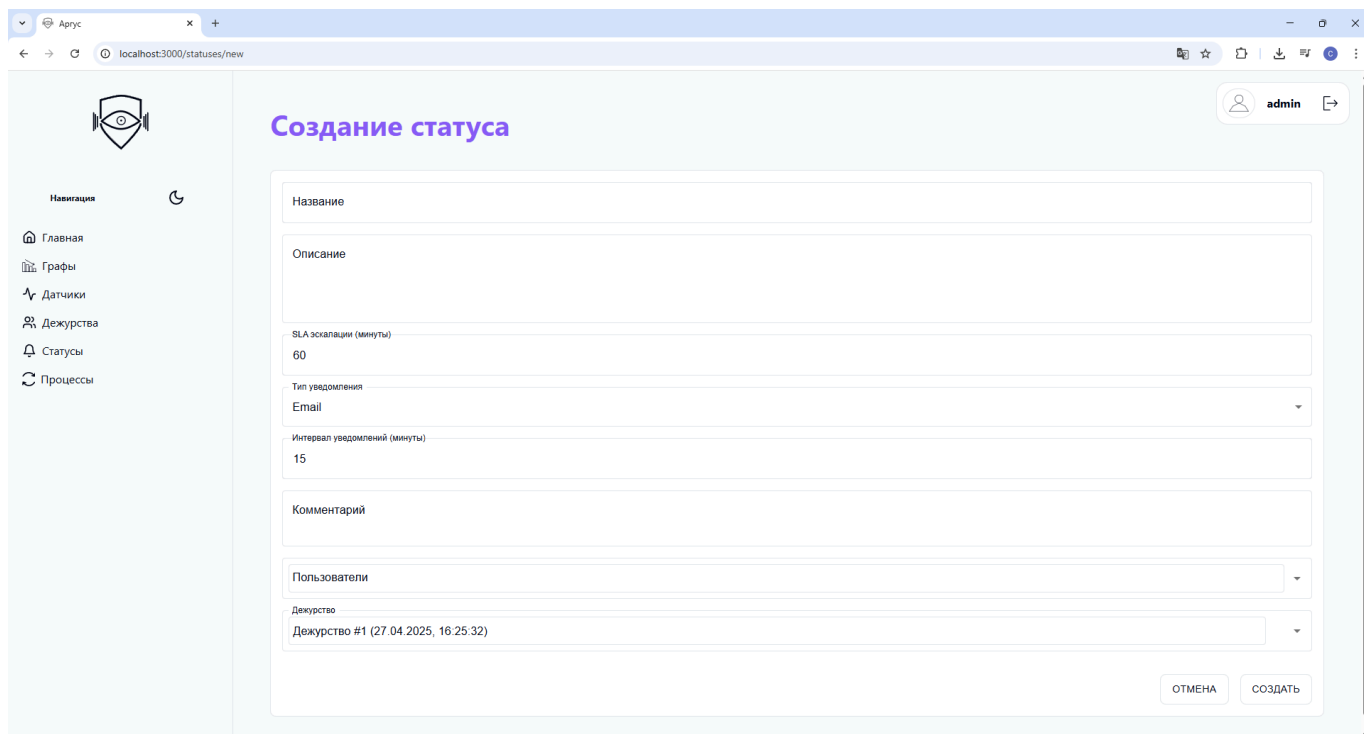


Рис 6. Создание/редактирование статуса

6.3.5 Функционал работы с графами статусов

При переходе на экран «графы статусов», начнётся загрузка доступных пользователю графов статусов и отображение их в списке. На каждый граф можно нажать, чтобы посмотреть и редактировать информацию о нём.

Создание графа статусов проверяется следующим образом:

1. Перейти в раздел "Графы статусов"
2. Нажать кнопку "Создать граф"
3. В открывшемся редакторе:
 - a. Ввести название графа
 - b. Ввести описание графа
 - c. Нажать кнопку "Добавить узел"
 - d. Кликнуть на рабочую область
 - e. Выбрать существующий статус из списка
 - f. Повторить для создания нескольких узлов
4. Для создания связей:

- a. Нажать кнопку "Добавить связь"
- b. Зажать левую кнопку мыши на исходном узле
- c. Протянуть линию к целевому узлу
- d. Отпустить кнопку мыши

5. Нажать кнопку "Сохранить"

Ожидаемый результат: Граф создан и отображается в списке графов.

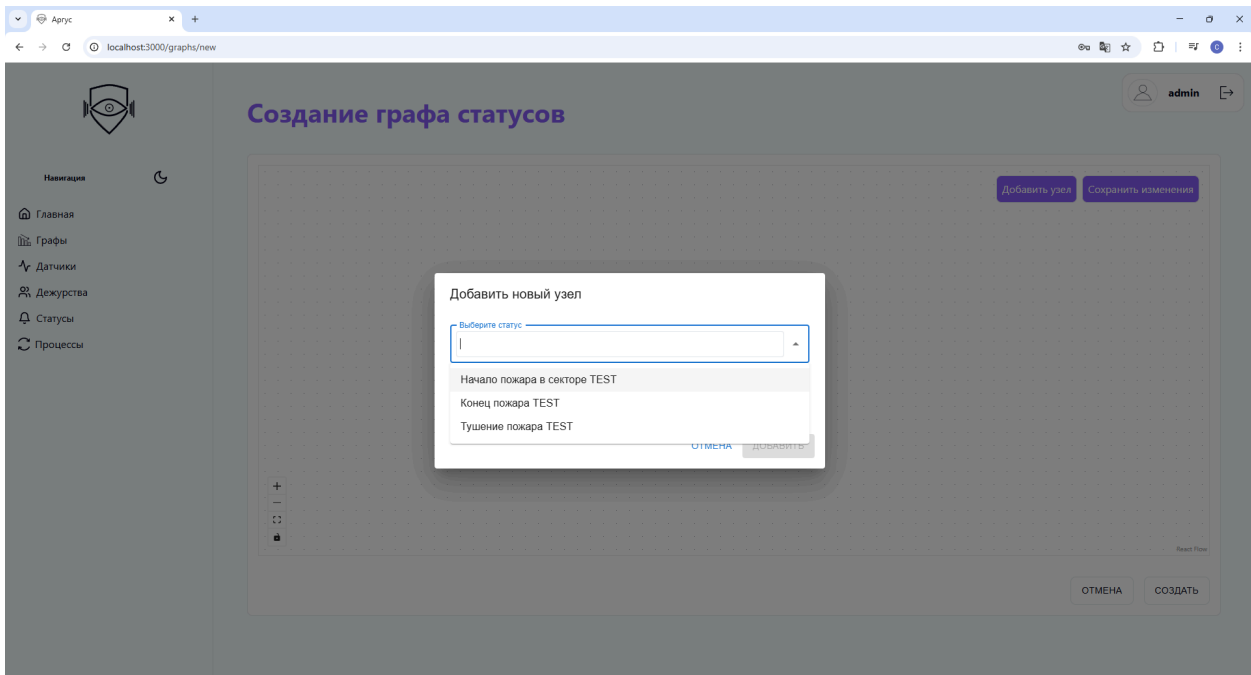


Рис. 7 Создание нового узла

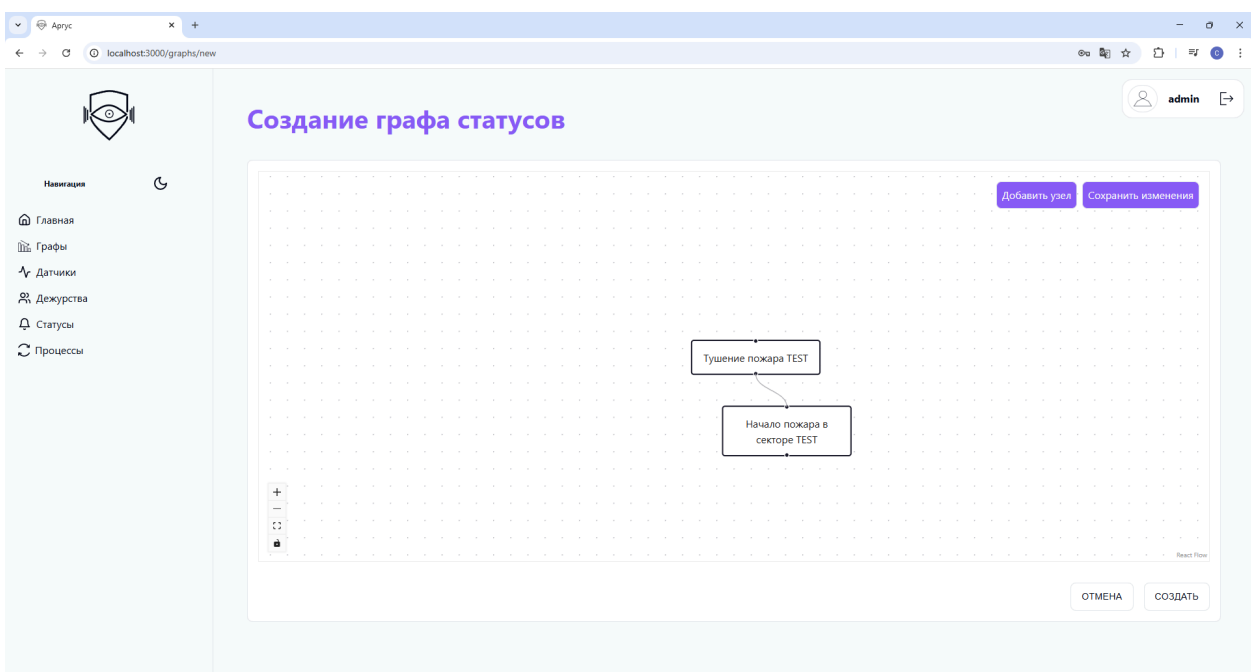


Рис. 8 Соединение двух узлов ребром

Просмотр графа статусов проверяется следующим образом:

1. Открыть существующий граф из списка
2. Проверить работу масштабирования:
 - a. Использовать колесико мыши
 - b. Использовать кнопки "+" и "-"
3. Проверить перемещение по графу:
 - a. Перетащить граф в разные стороны
 - b. Нажать кнопку "Центр"
 - c. Навести курсор на узлы и связи

Ожидаемый результат: Все элементы управления работают корректно, при наведении отображается информация.

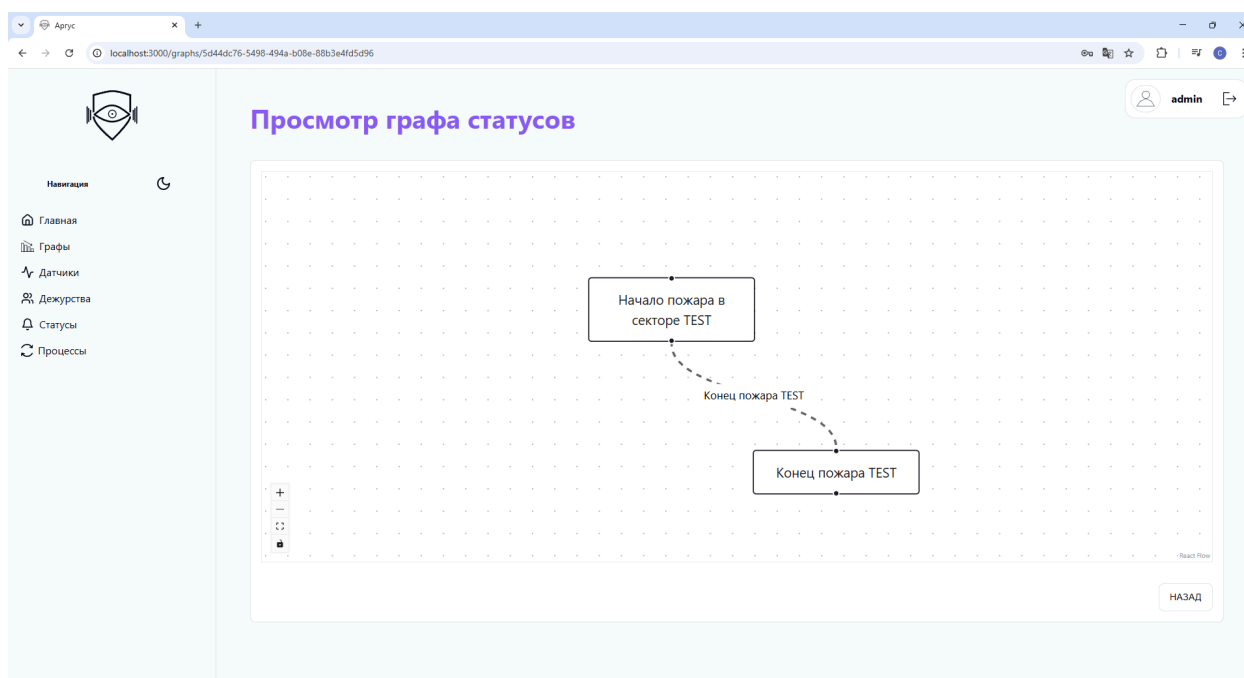


Рис. 9 Просмотр графа статусов

6.3.6 Функционал работы с очередями (процессами) и тикетами

При переходе на экран «процессы» (рис. 3), начнётся загрузка доступных пользователю процессов и отображение их в списке. На каждый процесс можно нажать, чтобы перейти на экран с этим процессом.

Создание процесса проверяется следующим образом:

1. Перейти в раздел "Процессы"
2. Нажать кнопку "Создать процесс"
3. Заполнить раздел "Основная информация":
 - a. Ввести название (проверить валидацию на пустое значение)
 - b. Ввести описание
4. Заполнить раздел "Настройки процесса":
 - a. Выбрать тип процесса
 - b. Установить приоритет по умолчанию
 - c. Настроить сроки выполнения
5. В разделе "Участники процесса":
 - a. Добавить ответственных через поиск
6. В разделе "Настройки уведомлений":
 - a. Выбрать тип уведомлений
 - b. Настроить частоту
 - c. Выбрать шаблоны
7. Нажать кнопку "Сохранить"

Ожидаемый результат: процесс создан и отображается в списке процессов

Создание процесса

Название

Описание

Граф статусов
Реагирование на пожар

ОТМЕНА СОЗДАТЬ

Рис.10 Создание процесса

Работа процесса в виде kanban-доски проверяется следующим образом:

1. Перейти в раздел "Очереди"
2. Проверить работу Kanban-доски:
 - a. Перетащить тикет между колонками
 - b. Проверить что все колонки отображаются согласно графу статусов, привязанному к процессу
 - c. Проверить, что тикет нельзя перетащить в колонки не по логике графа статусов
 - d. Навести курсор на тикет и выбрать его

Ожидаемый результат: Тикеты перемещаются между колонками, отображается информация при наведении, фильтры работают корректно.

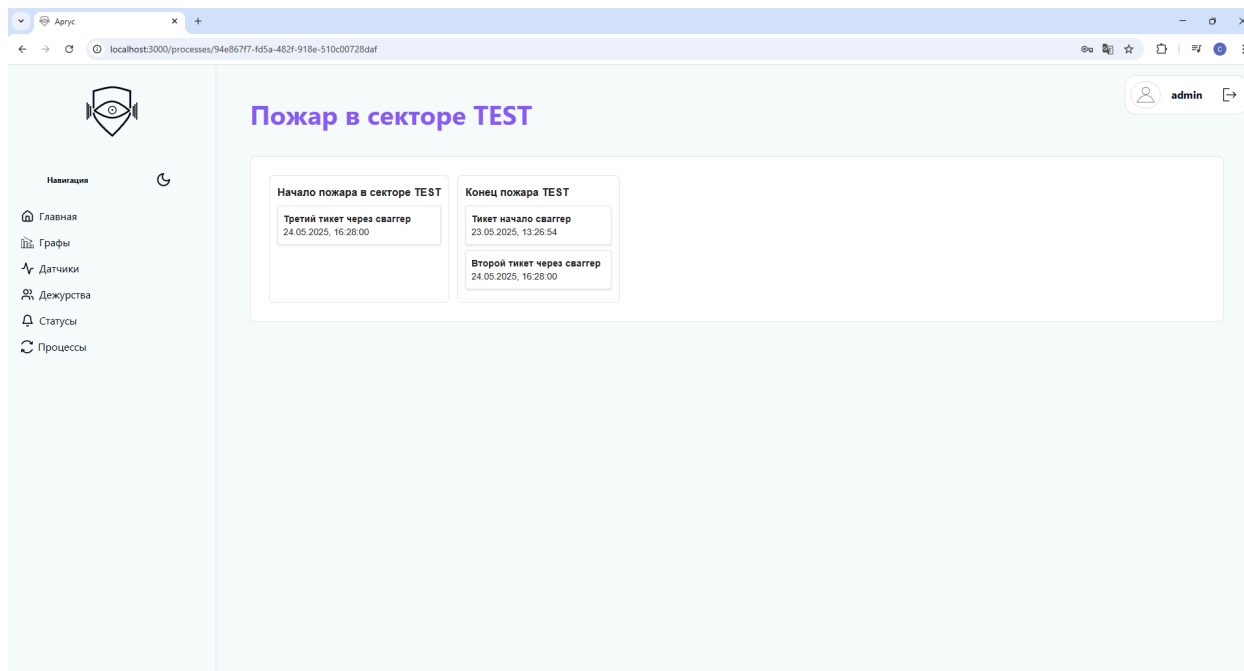


Рис.11 Движение тикета по процессу

Управление тикетом проверяется следующим образом:

1. При открытии тикета отображается детальная информация и доступны следующие действия:
 - a. Редактирование основной информации
 - b. Изменение статуса (через kanban-доску)

- с. Добавление комментариев
- d. Назначение ответственного (post-mvp)
- e. Установка дедлайна (post-mvp)

2. Указанные действия выполняются корректно

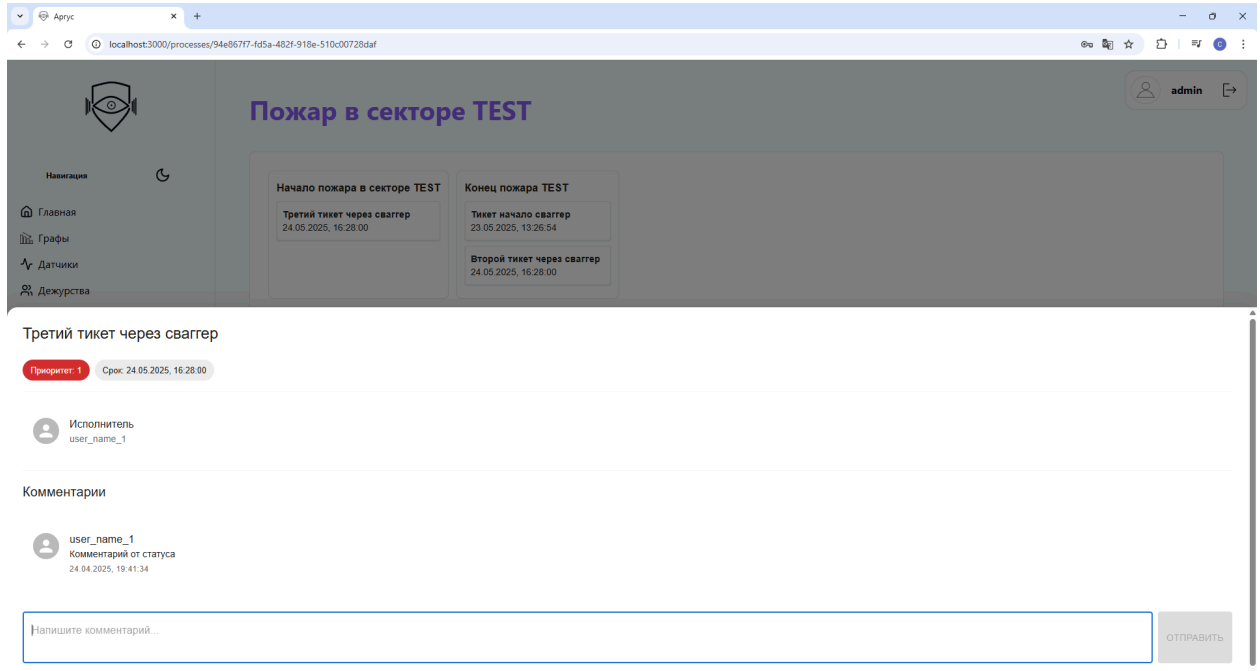


Рис.12 Управление тикетом

СПИСОК ИСТОЧНИКОВ

- 1) ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 10) ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 11) Документация React [Электронный ресурс] / Режим доступа: <https://react.dev/learn>, свободный
- 12) Atlassian Jira - <https://www.atlassian.com/software/jira> [электронный ресурс, дата обращения 21.11.2024]
- 13) Figma - <https://www.figma.com/> [электронный ресурс, дата обращения 21.11.2024]
- 14) Webstorm - <https://www.jetbrains.com/webstorm/download/> [электронный ресурс, дата обращения 21.11.2024]
- 15) Typescript - <https://www.typescriptlang.org/docs/> [электронный ресурс, дата обращения 21.11.2024]

- 16) Git - <https://git-scm.com/doc> [электронный ресурс, дата обращения 21.11.2024]
- 17) Docker - <https://docs.docker.com> [электронный ресурс, дата обращения 21.11.2024]
- 18) Material UI - <https://mui.com/material-ui/getting-started/> [электронный ресурс, дата обращения 21.11.2024]
- 19) Bootstrap - <https://getbootstrap.com/docs/5.3/getting-started/introduction/> [электронный ресурс, дата обращения 21.11.2024]
- 20) Документация ASP .NET Core [Электронный ресурс] / Режим доступа: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0>, свободный
- 21) Документация PostgreSQL [Электронный ресурс] / Режим доступа: <https://postgrespro.ru/docs/postgresql>, свободный
- 22) Документация Entity Framework Core [Электронный ресурс] / Режим доступа: <https://learn.microsoft.com/ru-ru/ef/>, свободный
- 23) Документация платформы Ignition SCADA [Электронный ресурс] / Режим доступа: <https://inductiveautomation.com/scada>, свободный.
- 24) Atlassian. JIRA Service Management [Электронный ресурс] / Режим доступа: <https://www.atlassian.com/software/jira/service-management>, свободный.
- 25) Документация PagerDuty [Электронный ресурс] / Режим доступа: <https://www.pagerduty.com>, свободный.
- 26) Документация OpsGenie [Электронный ресурс] / Режим доступа: <https://www.atlassian.com/software/opsgenie>, свободный.
- 27) OnPage. Лучшие практики управления инцидентами [Электронный ресурс] / Режим доступа: <https://www.onpage.com/incident-management-best-practices/>, свободный.
- 28) Мэ́ртин Фа́улер. Архитектурные шаблоны корпоративных приложений. Addison-Wesley Professional, 2002.
- 29) Сэм Ньюман. Создание микросервисов: проектирование детализированных систем. O'Reilly Media, Inc., 2022.
- 30) Документация Docker [Электронный ресурс] / Режим доступа: <https://docs.docker.com/>, свободный.
- 31) Документация Yandex Cloud Managed Kubernetes [Электронный ресурс] / Режим доступа: <https://cloud.yandex.com/docs/managed-kubernetes>, свободный.

Лист регистрации изменений

[illegible]