ЛАЗЕРНЫЕ СИСТЕМЫ

СЕРВЕРНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ «АЛКОЗАМОК-М СМАРТ»

Руководство по установке

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ 3
1.1 Назначение и основные функции программы 3
1.2 Требования к пользователю 3
1.3 Требования к аппаратному обеспечению 4
1.4 Минимальный состав программных средств 4
2 СТРУКТУРА ПРОГРАММЫ 5
3 НАСТРОЙКА ПРОГРАММЫ 6
3.1 Инструкция по развертыванию 6
3.1.1 Необходимое общее программное обеспечение 6
3.1.2 Настройка операционной системы Ubuntu на СВУ 6
3.1.3 Создание базы данных и настройка СУБД PostgreSQL 7
3.1.4 Установка и настройка прокси-сервера Nginx 8
3.2 Установка СПО 12
3.3 Инструкция по настройке СПО 13
4 ПРОВЕРКА ПРОГРАММЫ 16
5 СООБЩЕНИЯ ПОЛЬЗОВАТЕЛЮ 19

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение и основные функции программы

Серверное программное обеспечение «Алкозамок-М СМАРТ» (СПО) реализует систему управления и контроля доступа пользователей и систему контроля движения транспортных средств на основе взаимодействия с Мобильным программным обеспечением «Алкозамок-М СМАРТ» (МПО).

Система верхнего уровня (СВУ) с установленным СПО является внешним потребителем мониторинговой информации и устанавливается в автотранспортных предприятиях.

СПО реализует следующий функционал:

– систему управления и контроля доступа пользователей;

– авторизацию пользователя в системе при помощи проверки связки логин-пароль;

– возможность создания учетных записей и их изменение;

– в рамках управления доступом для пользователей с доступом к функционалу МПО должна поддерживаться привязка к Google-аккаунту и AppleID;

– систему контроля движения ТС на основе взаимодействия с МПО;

– взаимодействие с зарегистрированным анализатором через МПО с авторизированным водителем, имеющим доступ к указанному комплексу.

1.2 Требования к пользователю

Пользователь, осуществляющий установку СПО, должен иметь уверенные навыки сетевого администрирования, навыки развертывания и управления операционными системами Debian/Ubuntu, навыки администрирования СУБД PostgreSQL.

1.3 Требования к аппаратному обеспечению

Требования к СВУ:

– число процессорных ядер: 4, не менее;

– тактовая частота процессора: 2 ГГц, не менее;

– ОЗУ: 8 Гб, не менее;

– НЖМД или SSD: 1 Тб, не менее;

– пропускная способность сети: 100 Мбит, не менее.

Требования к рабочему месту пользователя:

– число процессорных ядер: 2, не менее;

– тактовая частота процессора: 2 ГГц, не менее;

– ОЗУ: 4 Гб, не менее;

– НЖМД или SSD: 40 Гб, не менее;

– пропускная способность сети: 100 Мбит, не менее.

Требования к каналам связи между рабочим местом пользователя и СВУ:

– время прохождения пакета (ping) от клиента до сервера не больше 100мс;

– скорость передачи данных от 256Кбит/с.

1.4 Минимальный состав программных средств

Требования к СВУ:

– должна быть установлена операционная система Debian 10 или Ubuntu 20.04 LTS x64 или выше;

– должна быть установлена и настроена СУБД PostgreSQL 12 или выше;

– должен быть установлен Oracle Java Development Kit x64 версии 17 или выше.

2 СТРУКТУРА ПРОГРАММЫ

Архитектура серверного программного обеспечения «Алкозамок-М СМАРТ» изображена на рисунке 1.

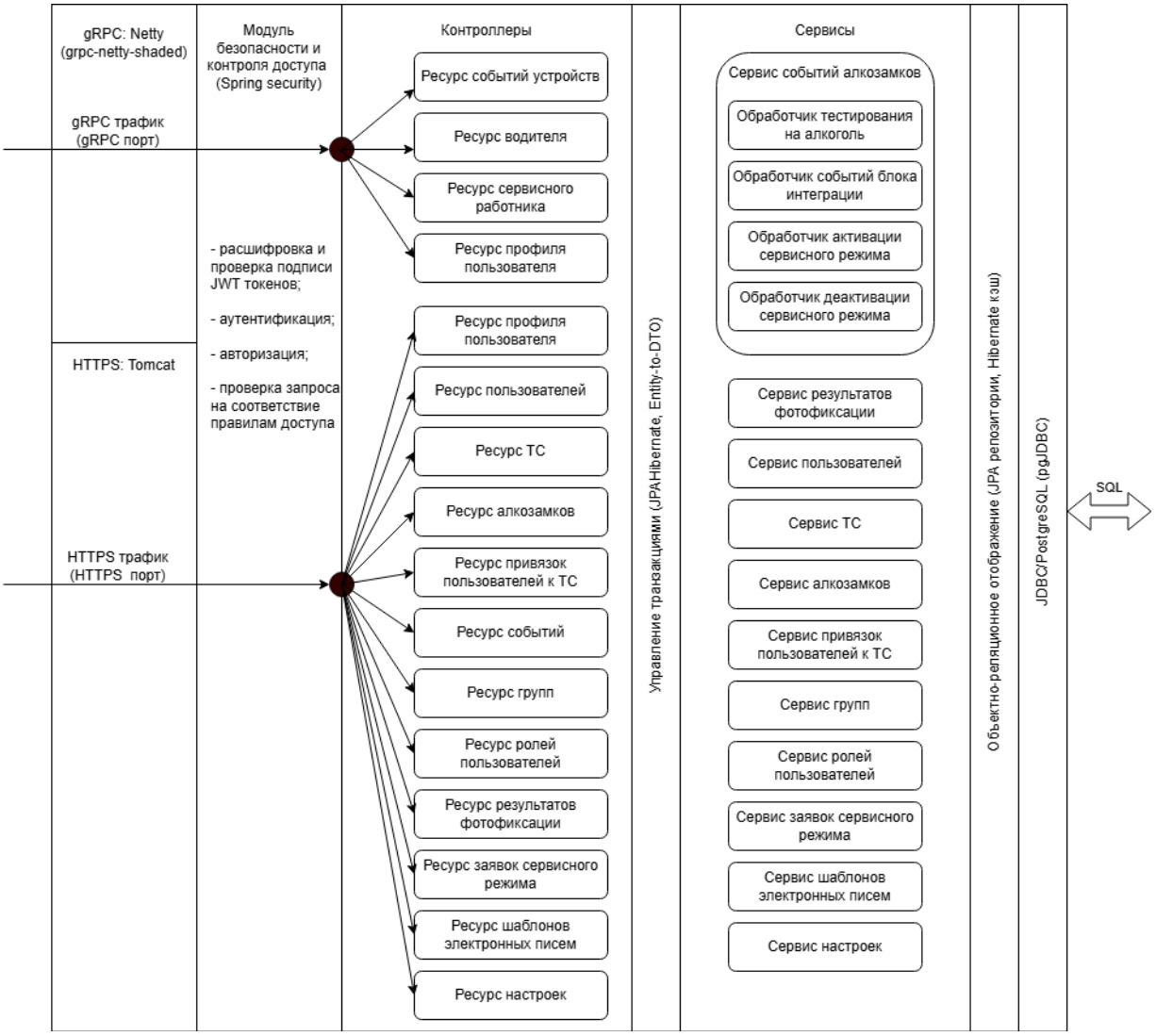

Рисунок 1 – Архитектурная схема СПО

СПО связано с МПО посредством приема и передачи данных с помощью фреймворка gRPC, основанного на протоколе удаленного вызова процедур (RPC).

Взаимодействие веб-интерфейса СПО с серверной частью осуществляется с помощью расширенного протокола HTTP с поддержкой шифрования в целях повышения безопасности (HTTPS) посредством интерфейса RESTful API.

3 НАСТРОЙКА ПРОГРАММЫ

3.1 Инструкция по развертыванию

3.1.1 Необходимое общее программное обеспечение

Для установки и эксплуатации СПО на сервере, отвечающего за работу СВУ, должно быть установлено следующее программное обеспечение:

- ОС Ubuntu 20.04 LTS x64 или выше;
- СУБД PostgreSQL из репозиториев установленной версии ОС;
- веб-сервер Nginx из репозиториев установленной версии ОС.

3.1.2 Настройка операционной системы Ubuntu на СВУ

Для настройки операционной системы Ubuntu на СВУ необходимо перейти в терминале в сессию административного пользователя с помощью команды

sudo su

Все дальнейшие действия выполняются под административным пользователем.

Для работы установочного пакета на СВУ требуются следующие утилиты: sed, adduser, grep, bash. Они должны быть предустановлены в ОС.

В случае отсутствия данных утилит, требуется выполнить их установку с помощью команды

apt install <имя_утилиты>

Например, *apt install sed*

Для установки использовать предоставленный deb-пакет СПО на произвольном носителе или доступный на скачивание по индивидуально предоставленной ссылке.

3.1.3 Создание базы данных и настройка СУБД PostgreSQL

Установить СУБД PostgreSQL можно из репозиториев ОС. Для этого необходимо в терминале выполнить команду

sudo apt-get install postgresql

После установки СУБД PostgreSQL запуститься автоматически, для подтверждения проверьте статус, выполнив команду

sudo systemctl status postgresql (СУБД должна находиться в статусе ‘active’). Далее необходимо создать базу данных и зарегистрировать пользователя для СПО Для этого нужно выполнить команды ниже. В случае смены пароля пользователя базы данных замените «'ls_alcolock'» в команде ниже на ваш пароль в одиночных кавычках.

Создание базы данных:

sudo -u postgres psql -c "CREATE DATABASE ls_alcolock;"

Проверка создания базы данных:

sudo -u postgres psql -c "\l" (в списке доступных БД должна присутствовать s_alcolock)

Создание пользователя:

sudo -u postgres psql -c "CREATE USER ls_alcolock WITH ENCRYPTED PASSWORD 'ls_alcolock';"

Проверка создания пользователя:

sudo -u postgres psql -c "\du" | grep ls_alcolock (Вывод должен содержать “ls_alcolock”)

Выдача прав пользователю:

sudo -u postgres psql -c "GRANT ALL PRIVILEGES ON DATABASE ls_alcolock TO ls_alcolock;"

Проверить что права выданы:

sudo -u postgres psql -c "\l ls_alcolock" (в столбце “Access privileges” должен фигурировать пользователь “ls_alcolock”)

Полная документация по СУБД PostgreSQL 12 доступна по ссылке https://www.postgresql.org/docs/12/index.html.

3.1.4 Установка и настройка прокси-сервера Nginx

Установить Nginx с помощью команды

apt install nginx

Добавить конфигурацию для СПО в файл conf.d, расположенный в директории /etc/nginx/.

К моменту создания конфигурационного файла Nginx, для сервера должны быть получены:

– SSL сертификат в формате «fullchain.pem»;

– приватный ключ SSL в формате «privkey.pem»;

Эти файлы необходимы для указания в конфигурации Nginx.

Создание конфигурационного файла Nginx:

– создать файл «alcolock.conf»;

файл должен содержать три серверных блока (server blocks).

Первый блок должен содержать настройки для перехвата запросов к серверу на порт HTTP (80) и перенаправлению их на порт HTTPS (443): пример конфигурации блока:

```
server {
    listen 80;
    # Замените на ваш домен или IP адрес
    server_name example.com;
    return 301 https://$host$request_uri;
}
```

Второй блок (основной) содержит настройки HTTPS сервера. пример конфигурации блока:

```
server {
 # IPv4: слушает 443 порт с SSL/HTTP
  listen 443 ssl http2;
   # IPv6: слушает 443 порт с SSL/HTTP2
listen [::]:443 ssl http2;
   # Отк. проверку клиентских сертификатов
```

```
        ssl_verify_client off;

        # Домены для сервера или IP адреса
        server_name example.com www.example.com;

        # Путь к SSL-сертификату (укажите путь, где храниться сертифика
        т)
        ssl_certificate /path/to/cert/fullchain.pem;

        # Путь к приватному ключу (укажите путь, где храниться ключ
        ssl_certificate_key /path/to/privkey/privkey.pem;

        # Максимальный размер тела запроса (10 МБ)
    client_max_body_size 10M;

    # Корневая директория статических файлов веб интерфейса
    root /opt/ls-alcolock-spo/static;

    # Файл по умолчанию
    index index.html;

    # CORS заголовки always гарантирует отправку заголовков даже при
ошибках

        add_header 'Access-Control-Allow-Credentials' 'true' always;
        add_header 'Access-Control-Allow-Methods' 'GET, POST, PUT, DELE
        TE, OPTIONS' always;
        add_header 'Access-Control-Allow-Headers' 'Accept...' always;
        add_header 'Access-Control-Expose-Headers' 'Authorization' always;

    location /api {

        # Перенаправление на бэкенд
        proxy_pass https://localhost:8443;

        # Передача оригинального хоста
        proxy_set_header Host $host;

        # Реальный IP клиента
        proxy_set_header X-Real-IP $remote_addr;
```

```
        # Обработка OPTIONS-запросов (preflight)
        if ($request_method = 'OPTIONS') {

      # Разрешенные домены для кросс-доменных запросов (подставляет
ся текущий origin)
            add_header 'Access-Control-Allow-Origin' '$http_origin';

      # Разрешенные HTTP-методы для кросс-доменных запросов
                add_header 'Access-Control-Allow-Methods' 'GET, POST, PUT,
DELETE, OPTIONS';

     # Разрешенные заголовки в запросах
            add_header 'Access-Control-Allow-Headers' 'DNT, User-Agent,
X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range,Auth
orization';

        # Время кеширования preflight-ответа в браузере (в секундах)
            add_header 'Access-Control-Max-Age' 1728000;

        # Тип содержимого ответа (отсутствует тело)
            add_header 'Content-Type' 'text/plain; charset=utf-8';

         # Длина содержимого (0 - пустое тело)
            add_header 'Content-Length' 0;

        # Возвращаем HTTP 204 (No Content) для preflight
            return 204;
         }

            }

         location / {
                # Пытается найти и отдать файлы в следующем порядке:
                # 1. Конкретный файл по запрошенному URI ($uri)
            # 2. Директорию по запрошенному URI ($uri/)
            # 3. Если ничего не найдено - отдает index.html (для SPA)
            try_files $uri $uri/ /index.html;
         }
```

```
    }
```

Третий блок содержит настройки RPC сервера

```
server {

# Слушать порт 9090 с поддержкой SSL и HTTP/2
listen 9090 ssl http2;

# Доменное имя сервера (замените на реальное)
server_name examle.com;

# Разрешенные версии TLS протокола
ssl_protocols TLSv1.2 TLSv1.3;

# Путь к SSL-сертификату (укажите путь, где хранится сертификат)
ssl_certificate /path/to/cert/fullchain.pem;

# Путь к приватному ключу (укажите путь, где хранится ключ)
ssl_certificate_key /path/to/privkey/privkey.pem;

# Отключение проверки клиентских сертификатов
ssl_verify_client off;

# Локация для обработки всех запросов (/)
location / {

    # Проксирование gRPC-трафика на backend-сервер
    grpc_pass grpcs://127.0.0.1:9080;

    # Установка заголовка Content-Type для gRPC
    grpc_set_header Content-Type application/grpc;

    # Включение поддержки трейлеров в gRPC
    grpc_set_header TE trailers;

    # Максимальное время чтения gRPC-ответа (1 час)
    grpc_read_timeout 3600;

    # Максимальное время отправки gRPC-запроса (1 час)
```

```
        grpc_send_timeout 3600;

        # Таймаут установки соединения с backend (60 сек)
        grpc_connect_timeout 60;

    }
}
```

– сохранить конфигурацию
sudo cp /path/to/alcolock.conf /etc/nginx/conf.d/
– проверить с помощью команды
sudo nginx -T

В случае успешного ответа (отсутствие ошибок в конфигурации) перезапустить прокси-сервер Nginx с помощью команды
systemctl restart nginx

Проверьте статус nginx
systemctl status nginx (nginx должен быть в статусе active)

3.2 Установка СПО

Скопируйте deb пакет на сервер.

Для установки СПО необходимо распаковать deb-пакет СПО через утилиту dpkg.

Например, *dpkg –i ls-alcolock-spo_1.0.0_linux-x64.deb*

Пакет устанавливается по пути «/opt/ls-alcolock-spo». При установке создается системный пользователь «ls-alcolock». Сервис регистрируется в «systemd» как «ls-alcolock-spo».

Папка для логов по пути «/var/log/alcolock/».

Для проверки состояния сервиса выполните команду

systemctl status ls-alcolock-spo

В пакете установлены:

– исполняемый jar-файл СПО – «alcolock-spo.jar»;

– среда исполнения Java – «OpenJDK 17.0.2 x86_64» в директории

«runtime»;

– веб-интерфейс СПО – в виде статических файлов в директории «static»;

– файлы конфигурации для СПО в директории «config»;

– основная конфигурация в «application-prod.yml»;

– конфигурация для TLS (HTTPS) в «application-tls.yml»;

– конфигурация логирования в «logback-spring.yml».

– конфигурация для кастомных настроек «application.yml»;

– конфигурация ключа доступа к ключу шифрования фото пользователей и фото результатов тестирования «application-photo-secret.yml»;

– конфигурация ключа подписи JWT токена «application-jwt-secret.yml»;

– папка для хранения фотографий результатов тестирования «alcotest_result_images»;

– папка для хранения фотографии пользователя «images»;

– файл конфигурации запуска приложения «start.sh».

3.3 Инструкция по настройке СПО

Перейти в директорию с установленным СПО с помощью команды

cd /opt/ls-alcolock-spo

Настройка происходит путем изменения файлов конфигурации.

При установке deb пакета были созданы случайные ключи шифрования для фотографий, получаемых в процессе тестирования на алкоголь и фотографий пользователя («application-photo-secret.yml». 256 бит в base64 кодировке), а также для подписи JWT-токенов («application-jwt-secret.yml», 512 бит в base64 кодировке). До первого запуска СПО их можно заменить на собственные ключи.

Идентификатор сервера для аутентификации можно изменить в

файле «application-jwt-secret.yml», по пути «auth:jwt:local:issuer-url». Рекомендуется сменить значение по умолчанию на домен сервера (если доступ планируется по домену *alcolock.domain.ru*, то идентификатор будет https://alcolock.domain.ru).

В файле «application-tls.yml», по пути «ssl: certificate-chain» нужно указать путь к сертификату формата «fullchain.pem», по пути «ssl: private-key» нужно указать путь к ключу формата «privkey.pem» (пути должны совпадать с указанными в конфигурационном файле Nginx).

В файле «application-prod.yml», по пути «spring:datasource» находится настройка адреса, имени пользователя и пароля от базы данных. Рекомендуется заменить пароль по умолчанию на собственный. Так же если был изменен дефолтный пароль на шаге создания базы данных нужно указать сохраненный пароль.

При первом запуске приложения в базе данных будет создано две схемы: «public», «batch_schema». В схеме «public» находятся основные таблицы приложения, в схеме «batch_schema» находятся служебные таблицы, для модуля «Spring Batch».

В файле «application-prod.yml», по путям «server:port» и «gprc:server:port» можно поменять HTTPS и gRPC порты соответственно.

В «application-prod.yml» по пути «spring:mail» можно настроить STM сервер, по средствам, которого будут оправляться письма пользователям при смене пароля.

В приложение используется расширение PostgreSQL "pgcrypto", чтобы его подключить нужно выдать права «супер пользователя», пользователю ls_alcolock до первого запуска приложения, для этого нужно выполнить команду

sudo -u postgres psql -c "ALTER USER ls_alcolock WITH SUPERUSER;"

После того как приложение запустится нужно вернуть права командой

sudo -u postgres psql -c "ALTER USER ls_alcolock WITH NOSUPERUSER;"

После внесения необходимых настроек можно запустить приложение

systemctl start ls-alcolock-spo

После запуска можно проверить статус приложения

systemctl status ls-alcolock-spo

Также можно проверить журнал вывода логов при запуске

journalctl -u ls-alcolock-spo –f.

4 ПРОВЕРКА ПРОГРАММЫ

Для проверки СПО необходимо знать веб-адрес, на обработку которого настроен прокси-сервер Nginx.

Веб-адрес предоставляется пользователю владельцем или распорядителем ПО «Информационная система «Алкозамок-М СМАРТ».

В адресной строке веб-браузера необходимо ввести адрес для доступа к веб-интерфейсу СПО, как показано на рисунке 2.

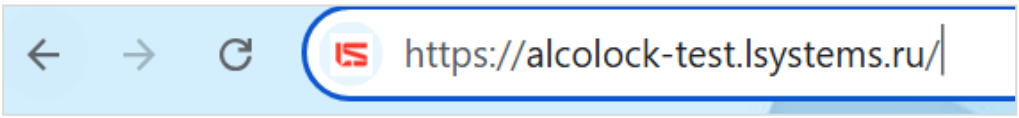

Рисунок 2 – Ввод адреса СПО в адресную строку веб-браузера

После этого происходит загрузка стартовой страницы с формой аутентификации пользователя Информационной системы «Алкозамок-М СМАРТ» (Рисунок 3).

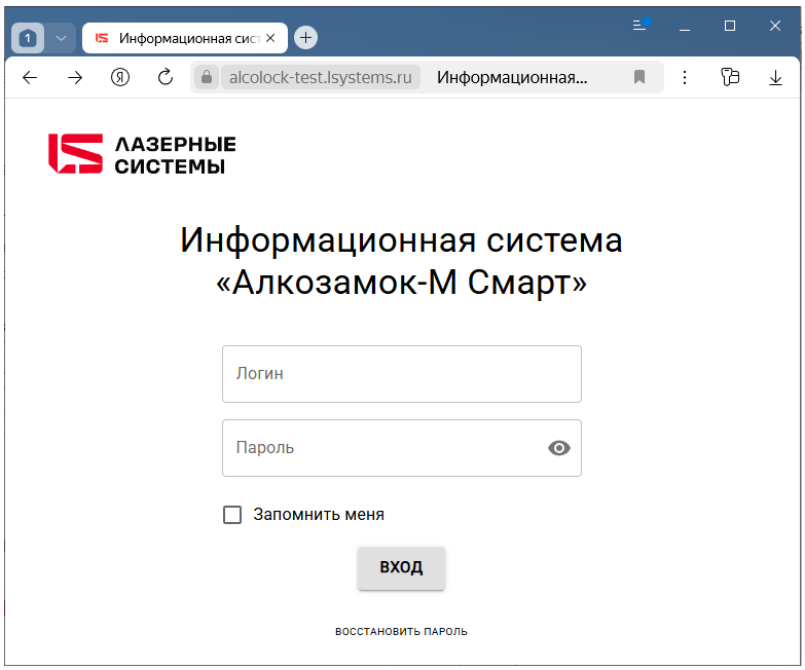

Рисунок 3 – Форма аутентификации пользователя

После предоставления пользователю владельцем или распорядителем Информационной системы «Алкозамок-М СМАРТ» учетных данных для доступа в СПО (логин и пароль) необходимо пройти

процесс аутентификации.

Аутентификация – вход пользователя в программу с проверкой учетных данных. На загруженной странице необходимо ввести логин и пароль и нажать кнопку «Войти».

В качестве логина используется электронный почтовый адрес пользователя, указываемый при создании пользователя в программе.

Для обеспечения конфиденциальности символы пароля не отображаются при вводе.

Если логин и пароль зарегистрированы в программе и введены правильно, то автоматически произойдет перезагрузка рабочей области веб-приложения СПО с перенаправлением на страницу «События», на которой в упорядоченном виде отображаются все события текущего филиала, в котором зарегистрирован данный пользователь (Рисунок 4).

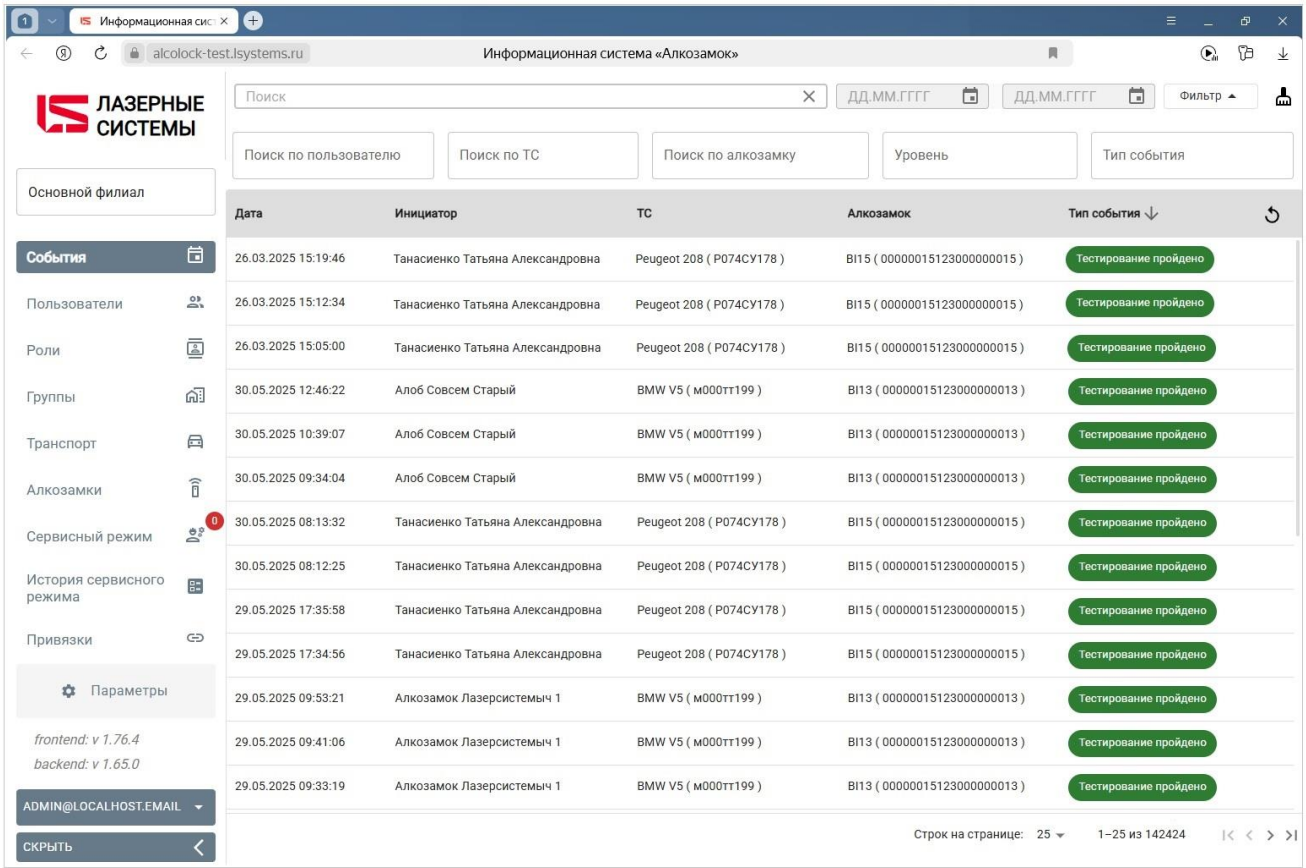

Рисунок 4 – Вкладка «События»

В левом нижнем углу будет отображен логин текущего авторизованного пользователя.

Если имя пользователя или пароль введены неправильно, либо указанное имя пользователя не зарегистрировано в программе, то в правом верхнем углу страницы с формой аутентификации отобразится всплывающее сообщение «Неверные учетные данные пользователя».

Загрузка рабочей области свидетельствует о работоспособности программы.

5 СООБЩЕНИЯ ПОЛЬЗОВАТЕЛЮ

В ходе установки, настройки и проверки программного обеспечения могут выдаваться сообщения. Сообщения могут содержаться в интерфейсе пользователя и в лог-файлах, которые ведутся базисным программным обеспечением. Диагностика сообщений должна проводиться в соответствии с официальной документацией программного обеспечения. Ниже представлен список ресурсов с возможными сообщениями, связанными с событиями СУБД PostgreSQL и работы виртуальной машины Java, в рамках которой исполняется СПО:

– https://www.postgresql.ord/docs/12/errcodes-appendix.html;

– https://docs.oracle.com/cd/E28280_01/core.1111/e10113/chapter_odi_messages.htm#FMERR170.

В ходе выполнения настройки, проверки или в процессе работы с Информационной системой «Алкозамок-М СМАРТ» пользователю могут выдаваться информационные сообщения. В таблице 1 описано содержание таких сообщений, а также действия, которые необходимо предпринять по данным сообщениям.

Таблица 1 – Сообщения пользователю

Текст сообщения	Описание сообщения	Возможные действия
BindException: Permission denied	Доступ к выбранному HTTPS порту заблокирован системой	Смените HTTPS порт в конфигурационном файле «application-prod.yml» на доступный пользователю «ls_alcolock» или запустите СПО с root правами
Web server failed to start. Port … was already in use	Выбранный HTTPS порт уже используется другим приложением	Смените HTTPS порт в конфигурационном файле «application-prod.yml» на свободный или освободите занятый порт
bind(..) failed: Permission denied	Доступ к выбранному gRPC порту заблокирован системой	Смените gRPC порт в конфигурационном файле «application-prod.yml» на доступный пользователю «ls_alcolock» или запустите СПО с root правами
bind(..) failed: Address already in use	Выбранный gRPC порт уже используется другим приложением	Смените gRPC порт в конфигурационном файле «application-prod.yml» на свободный или осво-

Текст сообщения	Описание сообщения	Возможные действия
		бодите занятый порт
PSQLException: Connection to … refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections	Не удалось установить соединение к базе данных согласно конфигурации в файле «application-prod.yml»	Проверьте конфигурацию базы данных в файле «application-prod.yml». Проверьте соединение к базе данных через команду «psql» с параметрами, соответствующими параметрам в файле «application-prod.yml»
PSQLException: FATAL: password authentication failed for user …	Не удалось пройти аутентификацию при соединении с базой данных	Проверьте корректность имени пользователя и пароль для базы данных из конфигурационного файла «application-prod.yml». Создайте пользователя, либо укажите корректный пароль в конфигурационном файле «application-prod.yml».
PSQLEXception: FATAL: database … does not exist	Не удалось установить соединение с базой данных согласно конфигурации в файле «application-prod.yml», так как указанная база данных отсутствует в СУБД	Проверьте имя базы данных в настройках подключения к СУБД в конфигурационном файле «application-prod.yml». Создайте отсутствующую базу данных и выдайте доступ пользователю, указанному в файле «application-prod.yml»
PSQLException: FATAL: permission denied for database	Не удалось установить соединение с базой данных согласно конфигурации в файле «application-prod.yml», так как у пользователя отсутствует доступ к базе данных	Выдайте права доступа пользователю базы данных из конфигурационного файла «application-prod.yml» через команду «psql»