

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Самарский государственный медицинский университет»
Министерства здравоохранения Российской Федерации

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ ДИСТАНЦИОННОГО МОНИТОРИНГА
ПАЦИЕНТОВ, ИМЕЮЩИХ В АНАМНЕЗЕ ХРОНИЧЕСКИЕ НЕИНФЕКЦИОННЫЕ
ЗАБОЛЕВАНИЯ**

УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ПО)

Листов 16

Самара, 2025 г.

Аннотация

Настоящий документ представляет собой руководство по установке программы «Программное обеспечение системы дистанционного мониторинга пациентов, имеющих в анамнезе хронические неинфекционные заболевания» (далее — Программа, Сервис).

В документе приведены следующие сведения:

- назначение и условия применения Программы;
- описание общей структуры Программы;
- установка и настройка;
- проверка работоспособности;
- сообщения администратору.

Документ разработан с учетом рекомендаций ГОСТ 19.101 «Единая система программной документации. Виды программ и программных документов», ГОСТ 19.503 «Единая система программной документации. Руководство системного программиста. Требования к содержанию и оформлению».

СОДЕРЖАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ.....	4
1.1. Область применения	4
1.2. Краткое описание возможностей.....	4
1.3. Требования к персоналу	4
2. СТРУКТУРА ПРОГРАММЫ	5
2.1. Структурное представление	5
2.2. Требования к техническому и программному обеспечению	6
3. УСТАНОВКА И НАСТРОЙКА ПРОГРАММЫ	7
3.1. Получение артефактов	7
3.2. Установка настроек приложения	7
3.3. Установка переменных окружения	10
3.4. Развертывание приложения на сервере.....	10
4. ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОГРАММЫ	12
5. СООБЩЕНИЯ ОБ ОШИБКЕ.....	13
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ.....	14
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	15

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Область применения

Сервис применяется в области здравоохранения частными и государственными организациями. Сервис предназначен для дистанционного мониторинга пациентов, имеющих в анамнезе хронические неинфекционные заболевания.

1.2. Краткое описание возможностей

Сервис в автоматическом режиме принимает данные показаний приборов из медицинской информационной системы (МИС) и заносит их в базу данных. Сервис позволяет осуществлять следующие функции:

- вести учет пациентов;
- принимать, хранить и анализировать данные показаний тонометров, используемых пациентами;
- предоставлять в текстовом, графическом и табличном виде отчеты по пациентам;
- уведомлять врача о выходе показаний пациента за пределы нормы.

1.3. Требования к персоналу

Системный администратор Сервиса должен обладать следующими навыками:

- инсталляция, общесистемное сопровождение и администрирование системы;
- администрирование СУБД (PostgreSQL);
- администрирование баз данных;
- администрирование системного и прикладного ПО;
- администрирование операционных систем;
- мониторинг работоспособности ПО и автоматизированных рабочих мест (АРМ), в том числе с использованием специального прикладного ПО.

2. СТРУКТУРА ПРОГРАММЫ

2.1. Структурное представление

Сервис построен на базе стандартной клиент-серверной архитектуры, где серверная часть реализована на основе фреймвока .NET Core на языке программирования С#, клиентская часть на связке HTML 5.0 + React, в качестве внутренней базы данных используется PostgreSQL 12.2.

Интеграция с сервисом сотового оператора выполнена по протоколу REST API с сообщениями в формате JSON.

Web UI администратора и врачей общается с сервером по REST API с сообщениями в формате JSON.

Структурная схема Программы представлена на рисунке 1.

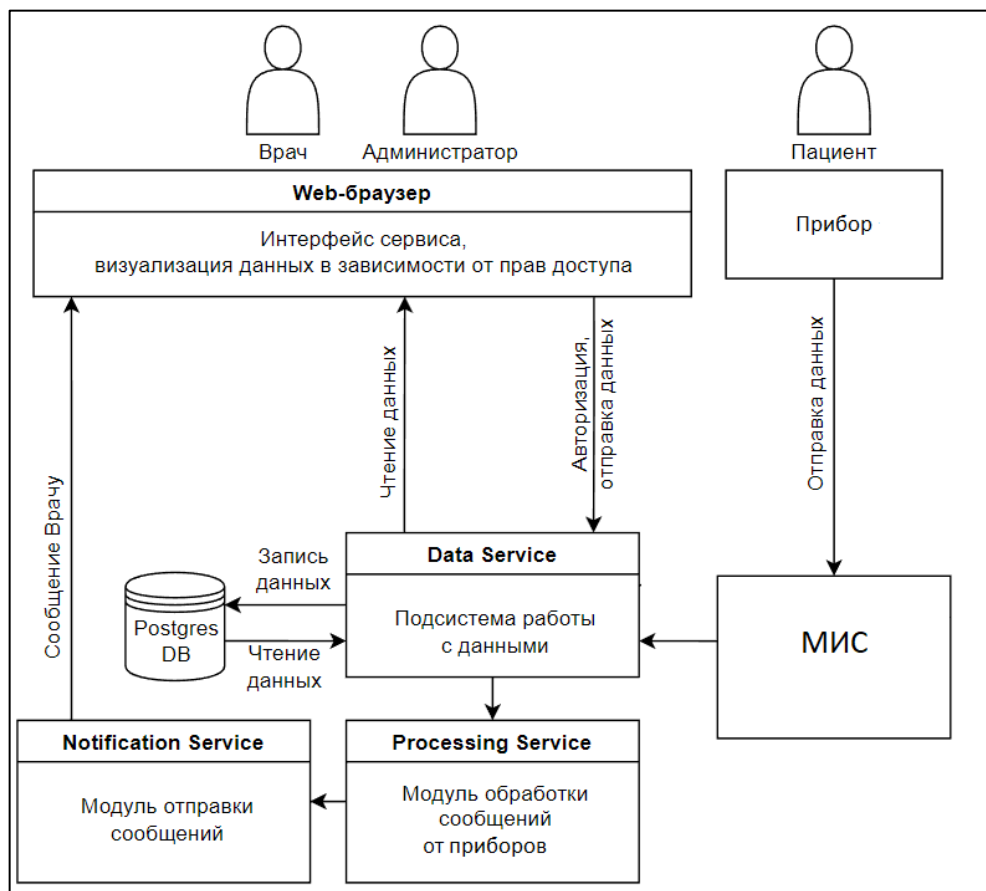


Рисунок 1 — Структурная схема Программы

В состав Программы входят следующие функциональные модули:

- врачей;
- пациентов;
- обработки данных;
- нотификации;
- отчетности;
- авторизации пользователей;
- визуализации физиологических показателей пациентов с приборов;
- интеграции с МИС.

2.2. Требования к техническому и программному обеспечению

Условием применения Сервиса является развернутый комплекс программно-технических средств в ЦОД медицинской организации.

Требования к программному обеспечению на АРМ пользователя:

- операционная система с поддержкой программного обеспечения доступа в интернет;
- браузер (за исключением Internet Explorer версии ниже 6.0).

Требования к минимальным характеристикам технических средств для эксплуатации Сервиса представлены в таблице 1.

Таблица 1 — Минимальные характеристики технических средств

Показатель	Сервер приложений	Сервер баз данных
Тип накопителя		SSD
Процессор	Не менее 4 ядер	Не менее 4 ядер
Оперативная память	Не менее 8 Гб	Не менее 8 Гб
Свободное дисковое пространство	Не менее 50 Гб	Не менее 100 Гб
Пропускная способность сетевого интерфейса	Не менее 50 Мбит/сек	Не менее 50 Мбит/сек

3. УСТАНОВКА И НАСТРОЙКА ПРОГРАММЫ

3.1. Получение артефактов

Артефактами являются образы бэкенда, фронтенда, базы данных, мигратора.

Архив содержит образы:

Бэкенда: **telemed-health-check-up-back.<сгенерированный тэг>.tar**.

Мигратора скриптов БД: **telemed-health-check-up-back-updater.<сгенерированный тэг>.tar**.

Артефакт фронтенда запрашивается у разработчиков фронтенда, обычно называется **frontend.tar**.

Артефактом базы данных является образ postgres версии 12.2: **postgres.12.2.tar**.

3.2. Установка настроек приложения

Скачать файл: **appsettings.Prod.json**.

В файле задаются конфигурируемые настройки приложения.

Описание настроек:

```
{
  "ServiceSettings": {
    "HostId": "HCU.CustomerName",           // Идентификатор
экземпляра приложения
    "WebUrl": "https://customer.web.url.ru"   // URL адрес
фронтенда
  },
  "SecuritySettings": {
    "JwtSchemas": {
      "Bearer": {
        "JwtValidityMinutes": 420,           // Срок
действия токена авторизации (обычно менять не надо)
        "JwtSecretKey": "сгенерировать ключ сюда" // Ключ для
генерации токена
      },
      "MobilePhone": {
        "JwtValidityMinutes": 5,             // Срок
действия токена для мобилки (обычно менять не надо)
        "JwtSecretKey": "сгенерировать ключ сюда" // Ключ для
генерации токена
      }
    }
  },
  "MailSettings": {                          // Настройки
почтового сервиса, если надо отправлять сообщения на почту
    "MailEnabled": false,
    "MailHost": "",
    "MailPort": 587,
    "MailUser": ""
  },
  "JitsiServiceSettings": {
```

```

    "Url": "https://jitsu.web.url.ru",           // URL адрес
сервиса видеоконференций jitsu
    "AppId": "jitsu.app.id",                     // Идентификатор
сервиса jitsu
    "Secret": "jitsu secret key"                 // Секретный ключ
для генерации токена авторизации jitsu
  },
  "LicenseServiceSettings": {                   //
Настройки сервера лицензий
    "LicenseStorePath": "BASE_URL",              //
Путь к директории с файлами, берется из переменной окружения
BASE_URL (менять не надо)
    "ApiUrl": "https://prod.license.mdapp.online", //
URL адрес сервера лицензий (менять не надо)
    "OrganizationName": "customer_name",         //
Название заказчика
    "ProductName": "62e05f36-fca6-4112-9653-de4d3300477d", //
Идентификатор продукта на сервере лицензии, данный ID
соответствует Telemed_hub2 (менять не надо)
    "Credentials": {
      "Email": "customer_name.hcu@smuit.ru"      //
Почтовый адрес аккаунта заказчика на сервере лицензии (см. п.
Активация лицензии)
    }
  },
  "VaultSettings": {
    "Path": "telemed/health-check-up/customer/customer_name" //
Путь для секретов в волте (см. п. Создание секретов в волте)
  },
  "NLog": {                                     //
Настройки логгера (менять обычно не надо)
    "throwConfigExceptions": false,
    "internalLogLevel": "Info",
    "internalLogToConsole": true,
    "extensions": [
      {
        "assembly": "NLog.Extensions.Logging"
      },
      {
        "assembly": "NLog.Web.AspNetCore"
      },
      {
        "assembly": "NLog.Targets.ElasticSearch"
      }
    ],
    "targets": {
      "file": {
        "type": "File",
        "fileName": "${path}/nlog/log.${shortdate}.log",
        "archiveFileName": "${path}/nlog/log.{#}.log",
        "archiveNumbering": "DateAndSequence",
        "archiveAboveSize": "10240",
        "archiveDateFormat": "yyyy-MM-dd",

```



```

        "layout": "${longdate}
(UTC+0)|${level:uppercase=true}|${logger}|${message:withexception
=true}"
    },
    "audit": {
        "type": "File",
        "fileName": "${path}/audit/app_audit.${shortdate}.log",
        "archiveFileName": "${path}/audit/app_audit.{#}.log",
        "archiveNumbering": "DateAndSequence",
        "archiveAboveSize": "10240",
        "archiveDateFormat": "yyyy-MM-dd",
        "layout": "${longdate} (UTC+0)|${message}"
    },
    "console": {
        "type": "Console",
        "layout": "${MicrosoftConsoleLayout}"
    }
},
"variables": {
    "path": "/data/files"
},
"rules": [
    {
        "logger": "Medmon.*",
        "minLevel": "Info",
        "writeTo": "file"
    },
    {
        "logger": "Telemed.*",
        "minLevel": "Info",
        "writeTo": "file"
    },
    {
        "logger": "Microsoft.AspNetCore",
        "minLevel": "Warning",
        "writeTo": "file"
    },
    {
        "logger": "audit",
        "minLevel": "Info",
        "writeTo": "audit"
    },
    {
        "logger": "*",
        "minLevel": "Warning",
        "writeTo": "console"
    }
]
}

```

3.3. Установка переменных окружения

Скачать файл [.env](#) и указать актуальные настройки (файл должен называться именно .env (с точкой в начале)).

В файле задаются параметры запуска приложения:

Уже указанные параметры обычно менять не требуется, только если, например, нужно будет поменять порты если указанные будут заняты, или тэг фронтенда поменяется.

```
VAULT_ADDR=https://telemed.vault.mdapp.online
VAULT_TOKEN=

MAIL_SENDER_PASS=

DOCKER_IMAGE_DATABASE=postgres.12.2
DATABASE_PORT=5432
DATABASE_NAME=postgres_db
DATABASE_USERNAME=postgres
DATABASE_PASS=

DOCKER_IMAGE_UPDATER=telemed-health-check-up-back-updater
DOCKER_IMAGE_BACKEND=telemed-health-check-up-back
BACKEND_TAG=
PORT_BACKEND=5000

DOCKER_IMAGE_FRONTEND=frontend
FRONTEND_TAG=master
PORT_FRONTEND=8080
```

Описание параметров:

- VAULT_ADDR: адрес волта
- VAULT_TOKEN: Временный токен доступа к волту
- MAIL_SENDER_PASS: Пароль для доступа к почтовому серверу если используется почта
- DOCKER_IMAGE_DATABASE: Название образа БД
- DATABASE_PORT: Порт БД
- DATABASE_NAME: Имя БД
- DATABASE_USERNAME: Имя пользователя для БД
- DATABASE_PASS: Пароль для БД
- DOCKER_IMAGE_UPDATER: Название образа мигратора
- DOCKER_IMAGE_BACKEND: Название образа бэкенда
- BACKEND_TAG: Тэг бэкенда, берется из пайплайна сборки (см. п. 3.1, та часть имени образа бэкенда, отмеченная как <сгенерированный тэг>)
- PORT_BACKEND: Порт бэкенда
- FRONTEND_TAG: Тэг фронтенда, обычно это master, если что, можно посмотреть в загруженном образе
- PORT_FRONTEND: Порт фронтенда.

3.4. Развертывание приложения на сервере

Для развертывания приложения на сервере необходимо:

1. Закачать на сервер файлы

Необходимо закачать на сервер в одну директорию следующие файлы:

- образы бэкенда, фронтенда, базы данных, мигратора (см. п. 3.1);
- appsettings.Prod.json (см. п. 3.2);
- .env (см. п. 3.3);
- docker-compose.yml;
- скрипт деплоя: script.sh.

2. Запуск деплоя на сервере

Выполнить команды на сервере в директории с файлами (для выполнения нужны права sudo, поэтому команды указаны уже с ними):

- **sudo chmod +x script.sh** (сделать скрипт исполняемым);
- **sudo ./script.sh** (выполнить скрипт).

4. ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОГРАММЫ

Чтобы проверить работоспособность Сервера необходимо открыть страницу авторизации Сервиса в браузере по ссылке `https://<IP адрес выделенного сервера под Сервис>/`.

В результате должен отобразиться интерфейс Сервиса, подробная информация о работе врача в Сервисе изложена в Руководстве пользователя.

5. СООБЩЕНИЯ ОБ ОШИБКЕ

При работе сервис дистанционного мониторинга пациентов, имеющих в анамнезе хронические неинфекционные заболевания, выдает следующие типы сообщений об ошибках:

– Ошибка **500** (Internal Server Error) — это внутренняя ошибка сервера. Она возникает, когда браузер или другой клиент отправляет серверу запрос, а тот не может его обработать. В случае возникновения данного сообщения необходимо просмотреть логи сервера.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Перечень сокращений, которые используются в настоящем документе, представлен в таблице 2.

Таблица 2

Сокращение	Расшифровка
АРМ	Автоматизированное рабочее место
БД	База данных
МИС	Медицинская информационная система
ПО	Программное обеспечение
ЦОД	Центр обработки данных
API	Application Programming Interface, программный интерфейс приложения, описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой
ID	Identifier, уникальный признак объекта, позволяющий отличать его от других объектов, то есть идентифицировать.
SSD	Solid-State Drive, твердотельный накопитель
URL	Uniform Resource Locator, уникальный указатель ресурса, система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Перечень терминов и определений, которые используются в настоящем документе, представлены в таблице 3.

Таблица 3

Термин	Определение
Docker	свободно распространяемое программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений с открытым исходным кодом
Docker-compose	свободно распространяемое инструментальное средство, входящее в состав Docker, предназначено для решения задач, связанных с развёртыванием проектов с открытым кодом
Postgres	свободно распространяемая объектно-реляционная система управления базами данных с открытым исходным кодом
REST	архитектурный стиль взаимодействия компонентов распределённого приложения в сети
REST-API	предназначен для взаимодействия со сторонними информационными системами
Автоматизированное рабочее место	комплекс компьютерного обеспечения, который предназначен для автоматизации выполняемых пользователем процессов
База данных	представленная в объективной форме совокупность материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины
Браузер	программа для поиска и просмотра на экране компьютера информации из компьютерной сети
Интернет	всемирная система объединённых компьютерных сетей для хранения и передачи информации
Интерфейс	общая граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил взаимодействия между элементами системы
Контейнер	запущенное приложение, совокупность процессов и образа
Образ системы	шаблон, который используется для создания контейнеров. Представляет собой слепок файловой системы, в котором расположен код приложения и его окружение
Операционная система	комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем
Сервис	Программное обеспечение системы дистанционного мониторинга пациентов, имеющих в анамнезе хронические неинфекционные заболевания

[illegible]