

Общество с ограниченной ответственностью «ТИМ»

**СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ ВРАЧЕБНЫХ РЕШЕНИЙ (СППВР) ПО  
КОНТРОЛЮ И КОРРЕКТИРОВКЕ АНТИГИПЕРТЕНЗИВНОЙ ТЕРАПИИ И ТЕРАПИИ  
ХРОНИЧЕСКОЙ СЕРДЕЧНОЙ НЕДОСТАТОЧНОСТИ (ХСН) НА ОСНОВАНИИ  
ДАННЫХ ПОСТОЯННОГО МОНИТОРИНГА ЭФФЕКТИВНОСТИ, ПЕРЕНОСИМОСТИ  
И ПРИВЕРЖЕННОСТИ ТЕРАПИИ**

УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ПО)

Листов 14

Самара, 2025 г.

### **Аннотация**

Настоящий документ представляет собой руководство по установке программы «Система поддержки принятия врачебных решений (СППВР) по контролю и коррективке антигипертензивной терапии и терапии хронической сердечной недостаточности (ХСН) на основании данных постоянного мониторинга эффективности, переносимости и приверженности терапии» (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 .....	8
3.1. Необходимые настройки .....	8
3.2. Подготовка к установке .....	8
3.3. Развертывание сервиса .....	8
3.4. Подготовка к запуску .....	8
3.5. Запуск и остановка сервиса .....	9
3.6. Заполнение базы данных .....	9
3.7. Просмотр логов .....	9
3.8. Настройка брандмауэра .....	9
4. ПРОВЕРКА РАБОТОСПОСОБНОСТИ ПРОГРАММЫ .....	10
5. СООБЩЕНИЯ ОБ ОШИБКЕ.....	11
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ.....	12
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	13

## **1. ОБЩИЕ СВЕДЕНИЯ**

### **1.1. Область применения**

Программа предназначена для сбора, хранения и обработки жизненных показателей пациента, собираемых посредством беспроводных каналов связи с медицинских приборов (тонометр, весы), а также с использованием ручного ввода информации (результаты анализов пациента, опросник самочувствия пациента).

Программа реализует анализ данных дистанционного мониторинга, результатом которого является набор рекомендаций по коррекции медикаментозного лечения для пациентов с хронической сердечной недостаточностью по фиксированным лекарственным группам.

### **1.2. Краткое описание возможностей**

Сервис позволяет осуществлять следующие функции:

- вести учет пациентов;
- принимать, хранить и анализировать данные показаний тонометра и весов, используемых пациентами;
- предоставлять в текстовом, графическом и табличном виде отчеты по пациентам;
- уведомлять врача о выходе показаний пациента за пределы нормы;
- выводить рекомендации на основе собранных данных при дистанционном мониторинге пациента.

В набор параметров для мониторинга входят следующие показатели:

- артериальное давление (мм рт. ст.);
- пульс (уд/мин);
- вес (кг).

### **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 фреймворка .NET Core на языке программирования C#, клиентская часть на связке HTML 5.0 + React, в качестве внутренней базы данных используется PostgreSQL 12.2.

Для реализации проекта спроектирована модель данных системы в СУБД PostgreSQL 12.2 и настроена система автоматического обновления схемы базы данных. В базе данных должно осуществляться хранение данных таких, как:

- логины и пароли, необходимые для авторизации пользователей в приложении;
- информация о пользователях, заполняемая пользователями на этапе регистрации в приложении, а также добавляемая в приложении после регистрации;
- результаты измерения каждого пользователя;
- дополнительная информация о принимаемых лекарственных аппаратах, дозировке, опросники по самочувствию и при осмотре пациента.

Для возможности взаимодействия с данными, хранящимися в БД, должна быть реализована серверная часть приложения и разработана программная часть приложения: Back-end, Front-end.

Для получения данных с персональных медицинских помощников (весы, тонометр) реализовано использование канала связи Bluetooth. Данное приложение устанавливается на планшет или смартфон пользователя-пациента. Обмен данными между устройством пользователя-пациента и устройством измерения осуществляется через технологию Bluetooth. Помимо этого, пользователю-пациенту реализована возможность вводить данные о приеме лекарственных средств, заполнение опросников вручную с помощью UI приложения.

Взаимодействие мобильного приложения с сервером Back-end (авторизация в приложении, обмен данными) должно осуществляться с помощью формата обмена данными REST API с текстовым форматом JSON на базе интернет-протокола HTTPS. Администрирование и настройку данного сервера должен осуществлять администратор напрямую.

Сервер Back-end должен взаимодействовать с БД по бинарному протоколу. Запросы к БД написаны на стандартном для реляционных баз данных языке SQL.

Данные о измерениях и показателях пациента передаются из сервера Back-end на Front-end по REST API, на базе интернет-протокола HTTPS. Таким образом, для пользователя-врача реализована функция просмотра результатов измерения своего пациента, а также заполнения опросных листов при осмотре, медикаментозной терапии, просмотр рекомендаций.

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

Структурная схема Программы представлена на рисунке 1.

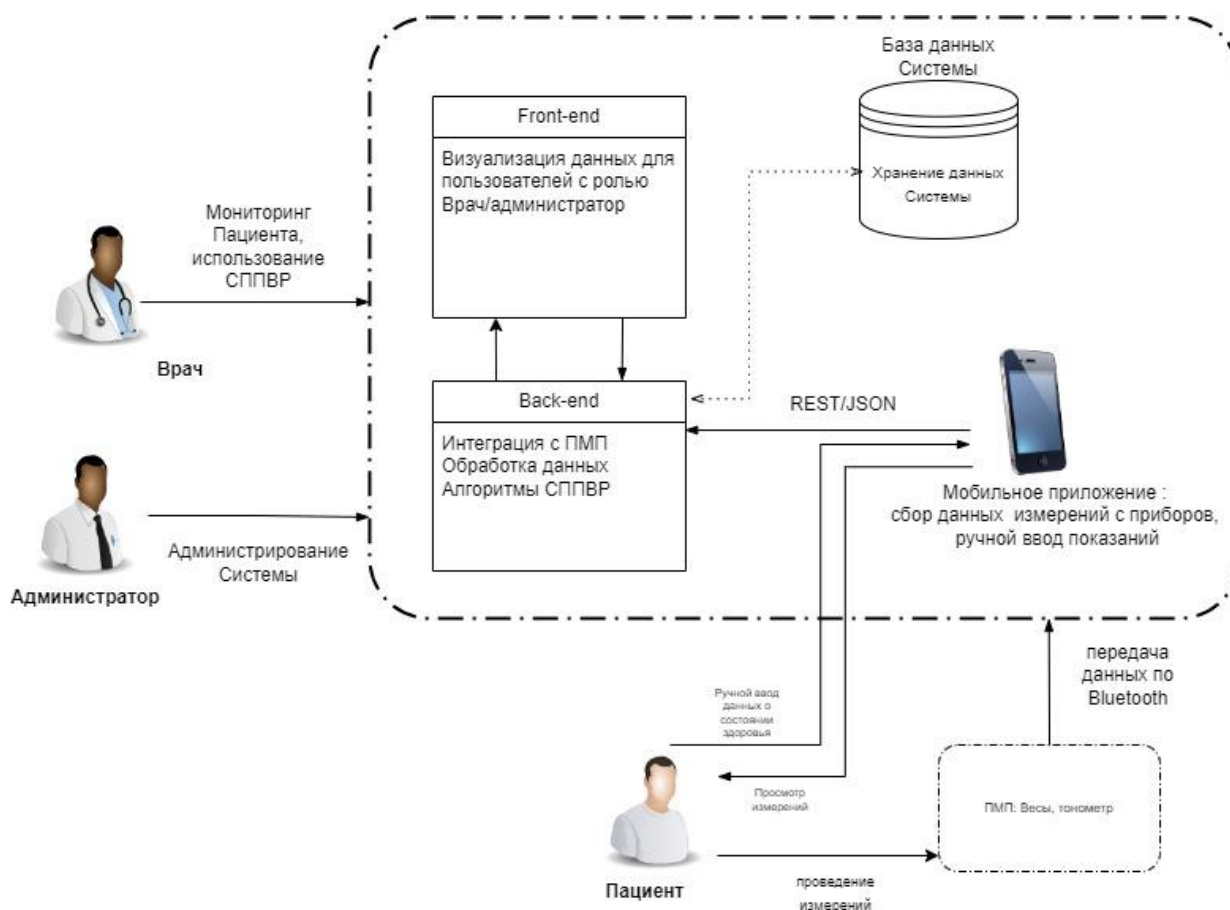


Рисунок 1 — Структурная схема Программы

В состав Программы входят следующие функциональные модули:

- Модуль «Личный кабинет врача — веб-клиент»;
- Модуль «Личный кабинет администратора — веб-клиент»;
- Модуль «Личный кабинет пациента — мобильное приложение»;
- Модуль обработки данных;
- Модуль нотификации;
- Модуль отчетности;
- Модуль авторизации пользователей;
- Модуль визуализации физиологических показателей пациентов с тонометра и весов;
- Модуль самостоятельного ввода данных;
- Модуль СППВР;
- Модуль интеграции медицинских устройств.

## 2.2. Требования к техническому и программному обеспечению

Требования к минимальным характеристикам технических средств для эксплуатации Сервиса представлены в таблицах 1-3.

Таблица 1 – Технические средства для функционирования мобильного приложения

Наименование	Ед. измерения	Допустимые значения
Требования к портативному устройству		
Bluetooth	-	Версия не ниже 4.0

Наименование	Ед. измерения	Допустимые значения
ПО портативного устройства пользователя	-	Android, версия не ниже 10 или HarmonyOS, версия не ниже 3
Объем оперативной памяти	Гб	Не менее 2
Объем свободного дискового пространства	Гб	Не менее 1
Пропускная способность сети интернет	Мбит/сек	Не ниже 1.0

Таблица 2 – Технические средства для серверного оборудования

Требования к серверу приложений		
Количество вычислительных потоков процессоров	Шт.	Не менее 4
Оперативная память	Гб	Не менее 8
Объем свободного дискового пространства	Гб	Не менее 100
Пропускная способность канала к серверу СУБД	Мбит/сек	Не менее 100
Пропускная способность канала в интернет	Мбит/сек	Не менее 200
ПО	-	ОС Linux x64: Docker (не менее) 19.03 Docker-compose (не менее) 1.25
Требования к серверу СУБД		
Количество вычислительных потоков процессоров	Шт.	Не менее 4
Оперативная память	Гб	8
Объем свободного дискового пространства	Гб	100
Пропускная способность канала к серверу приложений	Мбит/сек	Не менее 100
ПО	-	PostgreSQL (не менее) 12.2

Таблица 3 – Технические средства для ПК пользователя

Требования к АРМ врача, пациента и администратора			
ПО	ОС	-	Windows 7 и выше
	Веб-браузер	-	Firefox Яндекс Google Chrome Исключение: консольные браузеры и Internet Explorer версии не ниже 6.0

### 3. УСТАНОВКА И НАСТРОЙКА ПРОГРАММЫ

#### 3.1. Необходимые настройки

Для деплоя и запуска сервиса необходимо получить от заказчика DNS или IP адрес фронта.

#### 3.2. Подготовка к установке

Перед тем, как начать установку программного обеспечения, необходимо скопировать на сервер или компьютер, на котором предполагается развертывание системы, в любую пустую папку архив **Artifacts.zip** с артефактами и распаковать его.

Для распаковки архива подойдет любая программа-архиватор, поддерживающая формат \*.zip.

В распакованном архиве должны присутствовать следующие артефакты:

- telemed-backend-<specific name>.tar
- telemed-frontend-<specific name>.tar
- docker-compose.yml
- SQL-скрипты для инициализации базы данных.

#### 3.3. Развертывание сервиса

Для развертывания Сервиса необходимо выполнить следующие действия:

1. Загрузить на сервер docker-файлы — например, через FTP-клиент.
2. Подключиться по SSH-протоколу к терминалу сервера, на котором будет развернута система, или вызвать командную строку, если предполагается локальное развертывание системы.
3. Выполнить в командной строке команды:

```
docker load -i telemed-backend-<specific name>.tar
```

```
docker load -i telemed-frontend-<specific name>.tar
```

4. Проверить, что контейнер добавился в локальное хранилище docker-образов — для этого выполнить в командной строке команду `docker images`

#### 3.4. Подготовка к запуску

Для подготовки к запуску Сервиса необходимо настроить параметры запуска в файле **docker-compose.yml**, для этого выполнить следующие действия:

- 1) Задать логические тома **volumes** для контейнеров **database** и **backend** — каталоги должны существовать на хостовой машине.
- 2) В секции **db:environment** задать настройки подключения к базе данных:
  - POSTGRES\_USER — имя пользователя;
  - POSTGRES\_PASSWORD — пароль;
  - POSTGRES\_DB — имя базы данных.
- 3) В секции **backend:environment** задать настройки подключения к базе данных:
  - Data:ConnectionStrings:Main:Username — имя пользователя;



- `Data:ConnectionStrings:Main:Password` — пароль;
  - `Data:ConnectionStrings:Main:Database` — имя базы данных.
- 4) При необходимости задать порты для контейнеров **backend**, **frontend** и **database** в секциях **ports**.

### 3.5. Запуск и остановка сервиса

Запуск Сервиса осуществляется командой `docker-compose up -d` из рабочей папки (флаг `-d` позволяет выполнить команду в фоновом режиме).

Проверка работоспособности осуществляется входом в панель администратора системы (в зависимости от настроек URL может отличаться, но по умолчанию это `<IP>` адрес сервера где поднята система).

Остановка сервиса осуществляется командой `docker-compose down`.

### 3.6. Заполнение базы данных

Для заполнения базы данных таблицами и начальными параметрами необходимо подключиться к базе данных, используя параметры из **docker-compose.yml**, и выполнить скрипт миграции SQL, приложенный в архиве.

### 3.7. Просмотр логов

В файле **docker-compose.yml** в секции **backend:volumes** указан путь для хранения данных приложения, формат пути:

```
/srv/hub/<COMMIT_REF>/backend
```

Логи хранятся в поддиректории **logs**, полный путь до логов будет следующим:

```
/srv/hub/<COMMIT_REF>/backend/logs
```

### 3.8. Настройка брандмауэра

Для работы приложения необходимо открыть доступ к следующим ресурсам:

- **dev.launcher.rehab.smu.it.ru** — сервер лицензирования для активации лицензии;
- **elastic-prod.smu.it.ru** — сервер статистики.

#### **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 с артериальной гипертензией 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 4.

Таблица 4

Сокращение	Расшифровка
АРМ	Автоматизированное рабочее место
БД	База данных
ПК	Персональный компьютер
ПО	Программное обеспечение
СУБД	Система управления базами данных
СППВР	Система поддержки принятия врачебных решений
ХСН	Хроническая сердечная недостаточность
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.
SIM, SIM-карта	Subscriber Identification Module, модуль идентификации абонента, который применяется в мобильной связи
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, представлен в таблице 5.

Таблица 5

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
PostgreSQL	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
Весы	медицинский диагностический прибор для измерения веса
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
Интернет	всемирная система объединённых компьютерных сетей для хранения и передачи информации
Сервис	система поддержки принятия врачебных решений по контролю и корректировке антигипертензивной терапии и терапии хронической сердечной недостаточности на основании данных постоянного мониторинга эффективности, переносимости и приверженности терапии
Тонометр	медицинский диагностический прибор для измерения артериального давления

[illegible]