

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

**ПЛАТФОРМА ДИСТАНЦИОННОГО МОНИТОРИНГА  
ФИЗИОЛОГИЧЕСКИХ ПОКАЗАТЕЛЕЙ ПАЦИЕНТА  
«HEALTH CHECK-UP»  
ОПИСАНИЕ ТЕХНИЧЕСКОЙ АРХИТЕКТУРЫ  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Листов 31

**Самара 2023 г.**

## АННОТАЦИЯ

Настоящий документ содержит описание технической архитектуры программного обеспечения «Платформа дистанционного мониторинга физиологических показателей пациента «Health check-up» (далее — Сервис).

В документе приведены следующие сведения:

- общие сведения о Сервисе;
- описание логической структуры;
- структура базы данных.

Документ разработан с учетом рекомендаций ГОСТ 2.105 «Единая система конструкторской документации. Общие требования к текстовым документам», ГОСТ 19.101 «Единая система программной документации. Виды программ и программных документов», ГОСТ 19.105 «Единая система программной документации. Общие требования к программным документам».

Термины, применяемые в настоящем документе, и их определения — по ГОСТ Р 59853 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения».

**СОДЕРЖАНИЕ**

<b>1. ОБЩИЕ СВЕДЕНИЯ .....</b>	<b>7</b>
1.1. НАИМЕНОВАНИЕ ПРОГРАММЫ .....	7
1.2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ .....	7
1.3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, НЕОБХОДИМОЕ ДЛЯ ФУНКЦИОНИРОВАНИЯ СЕРВИСА.....	7
1.4. ИСПОЛЬЗУЕМЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ: .....	7
1.5. ЛИНГВИСТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ .....	8
<b>2. СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....</b>	<b>9</b>
2.1. АРХИТЕКТУРА СЕРВИСА .....	9
2.2. ФУНКЦИОНАЛЬНАЯ СТРУКТУРА .....	9
<b>3. МЕТОДЫ API .....</b>	<b>14</b>
3.1. ВХОД В СИСТЕМУ.....	14
3.2. СОЗДАНИЕ ПРОГРАММЫ МОНИТОРИНГА .....	15
3.3. ИЗМЕНЕНИЕ ПРОГРАММЫ МОНИТОРИНГА.....	18
3.4. СНЯТИЕ С МОНИТОРИНГА .....	22
3.5. ПОЛУЧЕНИЕ ПРОГРАММЫ МОНИТОРИНГА .....	23
3.6. ДОБАВЛЕНИЕ ИЗМЕРЕНИЯ .....	24
3.7. ПОЛУЧЕНИЕ ССЫЛКИ НА СТРАНИЦУ ПАЦИЕНТА .....	24
3.8. ИЗМЕНЕНИЕ ПРОФИЛЯ ПАЦИЕНТА.....	25
3.9. ИЗМЕНЕНИЕ ПРОФИЛЯ ДОКТОРА .....	26
<b>4. СТРУКТУРА БАЗЫ ДАННЫХ.....</b>	<b>27</b>

### Перечень сокращений

Перечень сокращений, которые используются в настоящем документе, представлен в Таблица 1.

Таблица 1 — Перечень сокращений

Сокращение	Расшифровка
БД	База данных
МО	Медицинская организация
ЦОД	Центр обработки данных
ФАП	Фельдшерско-акушерский пункт
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 Таблица 2.

Таблица 2 — Перечень терминов и определений

Термин	Определение
Автоматизированная система	Система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций.
Браузер	программа для поиска и просмотра на экране компьютера информации из компьютерной сети
База данных	Совокупность взаимосвязанных данных, организованных в соответствии со схемой базы данных таким образом, чтобы с ними мог работать пользователь.
Веб-приложение	Прикладное программное обеспечение, которое работает на веб-сервере, в отличие от компьютерных программ, которые запускаются локально в операционной системе устройства. Доступ к веб-приложениям осуществляется пользователем через веб-браузер с активным сетевым подключением.
Интерфейс	Общая граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил взаимодействия между элементами системы
Логи	Записи о событиях в хронологическом порядке, простейшее средство обеспечения журналирования
МО	медицинская организация
Операционная система	комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем
Пациент	Пациент с установленной артериальной гипертензией
Прибор	см. тонометр
Интернет	всемирная система объединённых компьютерных сетей для хранения и передачи информации
Сервис	сервис дистанционного мониторинга пациентов с артериальной гипертензией
Docker	Свободно распространяемое программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений с открытым исходным кодом

Термин	Определение
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
Web-интерфейс	Совокупность правил взаимодействия устройств и программ между собой или с пользователем и средств, реализующих это взаимодействие.

## 1. ОБЩИЕ СВЕДЕНИЯ

### 1.1. Наименование программы

Наименование: Платформа дистанционного мониторинга физиологических показателей пациента «Health check-up».

### 1.2. Функциональное назначение

Сервис предназначен для сбора и последующего представления агрегированных данных для мониторинга медицинских показателей пациента, используя приборы с функцией передачи результатов по беспроводному каналу Bluetooth.

В набор параметров для мониторинга входят следующие показатели:

- |                                       |                          |
|---------------------------------------|--------------------------|
| - артериальное давление (мм рт. ст.); | - вес (кг);              |
| - пульс (уд/мин);                     | - биоимпеданс;           |
| - гемоглобин;                         | - сатурация;             |
| - холестерин;                         | - аускультация;          |
| - уровень глюкозы в крови (ммоль/л);  | - физическая активность; |
| - инсулин;                            | - сон;                   |
| - углеводы;                           | - спирометрия;           |
| - температура тела (С°);              | - анализ мочи;           |
|                                       | - ЭКГ,                   |
|                                       | - ВАС.                   |

### 1.3. Программное обеспечение, необходимое для функционирования Сервиса

Для клиентской части:

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

Для серверной части:

- СУБД PostgreSQL 8.3 и выше;
- Контейнеризатор приложений Docker 19.03.
- Пакетный менеджер Docker-compose

### 1.4. Используемые языки программирования:

- язык гипертекстовой разметки HTML;
- каскадные таблицы стилей CSS — формальный язык описания внешнего вида веб-документа, разработанного с применением языка разметки HTML (XHTML);
- встраиваемый язык программного доступа к объектам приложений JavaScript;
- язык разработки веб-приложений Java.

### **1.5. Лингвистическое обеспечение**

Взаимодействие пользователя с Сервисом осуществляется на русском языке. Исключение составляют только системные (администраторские) команды, сообщения на английском языке (системные и технические сообщения, не подлежащие русификации) для персонала, выполняющего функции администрирования Сервиса.

Все документы, производимые инструментами Сервиса, предоставляются пользователю на русском языке.

Графический интерфейс пользователя Сервиса реализован на русском языке.

## 2. СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

### 2.1. Архитектура сервиса

Сервис построен на базе стандартной клиент-серверной архитектуры, где серверная часть реализована на основе фреймворка .NET 6 на языке программирования C#, клиентская часть на связке HTML 5.0 + React, в качестве внутренней базы данных используется PostgreSQL 12.2.

Схема архитектуры Сервиса представлена на рис. **Рисунок 1**.

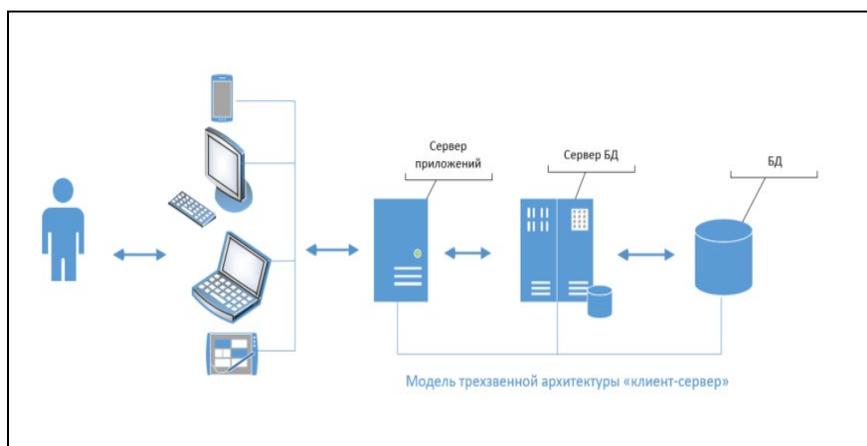


Рисунок 1 — Схема архитектуры Сервиса

Приборы передают данные измерений на мобильное устройство пациента по Bluetooth-соединению. Полученные данные передаются мобильным устройством на сервер приложений по протоколам TCP/IP, где данные обрабатываются, передаются на сервер баз данных и записываются в базу данных.

### 2.2. Функциональная структура

В состав Сервиса входят следующие функциональные модули:

- личный кабинет среднего медицинского персонала — мобильный клиент;
- личный кабинет врача — веб-клиент;
- личный кабинет администратора — веб-клиент;
- обработки данных;
- нотификации;
- отчетности;
- авторизации пользователей;
- визуализации физиологических показателей пациентов с цифровых устройств;
- самостоятельного ввода данных пациента медицинским работником;
- интеграции приборов;
- интеграции с МИС.

Функциональная структура Сервиса представлена на рис. Рисунок 2.

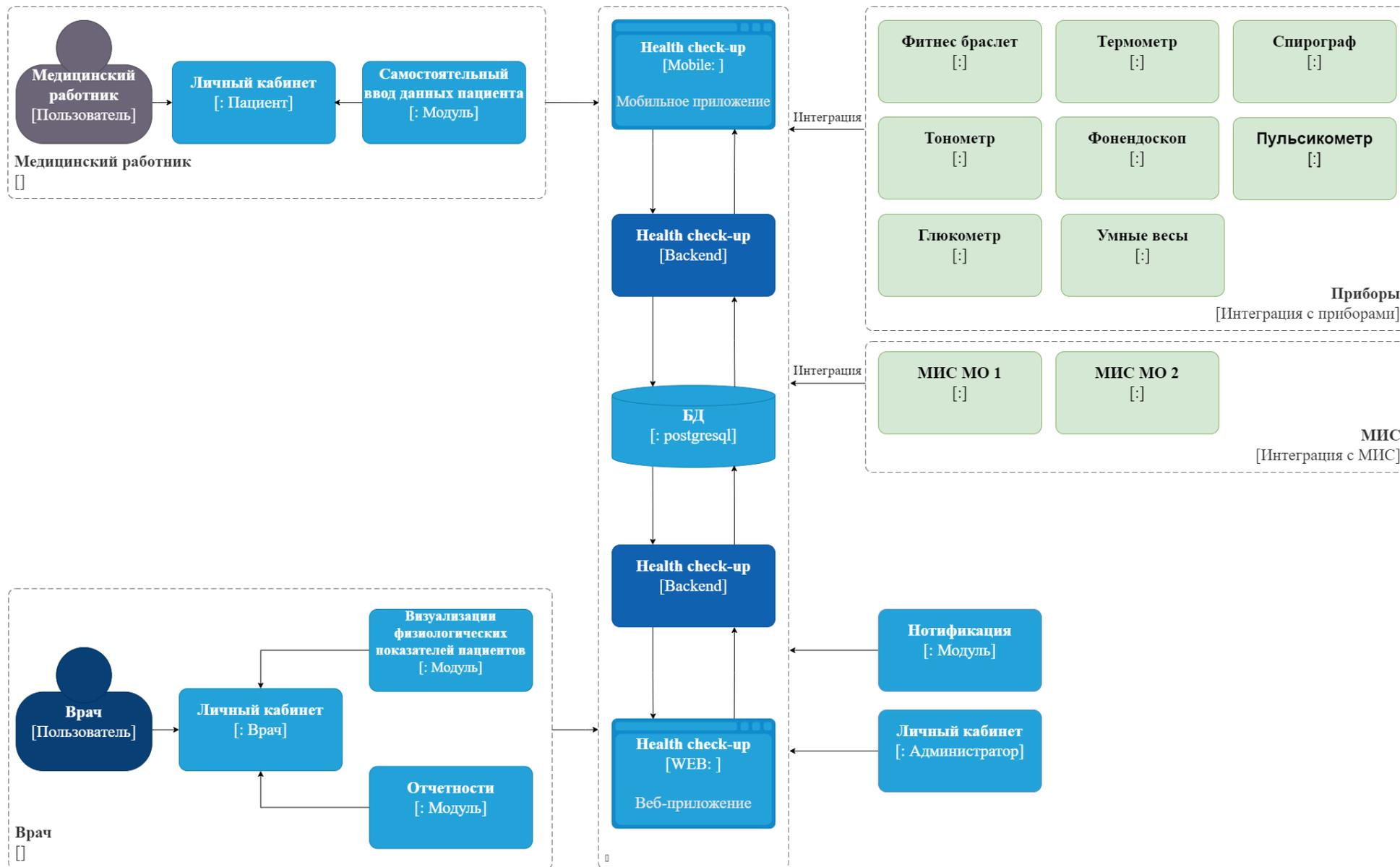


Рисунок 2 — Схема взаимодействия компонентов системы

### **2.2.1. Личный кабинет медицинского работника (мобильный клиент)**

Модуль предназначен для выполнения следующих функций:

- Регистрация пациентов в ФАП;
- Подключение устройства для сбора физиологических показателей пациента;
- Ввод данных в ручном режиме;
- Просмотр измеренных показателей;
- Настройка отображаемых данных;
- Редактирование профиля пациента;
- Фиксации начала приема пациента;
- Проведение анкетирования пациента.

### **2.2.2. Личный кабинет врача (веб-клиент)**

Модуль предназначен для выполнения следующих функций:

- регистрация пользователей с ролью «Пациент»;
- просмотр данных о пациенте;
- редактирование данных о пациентах;
- просмотр результатов измерений пациента;
- настройки отображаемой информации;
- настройки границ норм показателей здоровья;
- настройки нотификации о выходе за границы установленных норм;
- просмотр и получение уведомлений;
- формирование комментариев к измерениям;
- редактирование профиля врача;
- восстановление пароля;
- смена пароля.

### **2.2.3. Личный кабинет администратора (веб клиент)**

Модуль предназначен для выполнения следующих функций:

- регистрация пользователей с ролью «Врач», «Медицинская сестра»;
- просмотр списка пользователей;
- редактирования данных о пользователях;
- деактивация пользователей;
- настройка референсных значений по измеряемым показателям;
- управление списками диагнозов;
- регистрация медицинских организаций;
- редактирования медицинских организаций.

В Сервисе реализована четырехуровневая структура подчиненности медицинских организаций согласно схеме, представленной на рис. Рисунок 3.

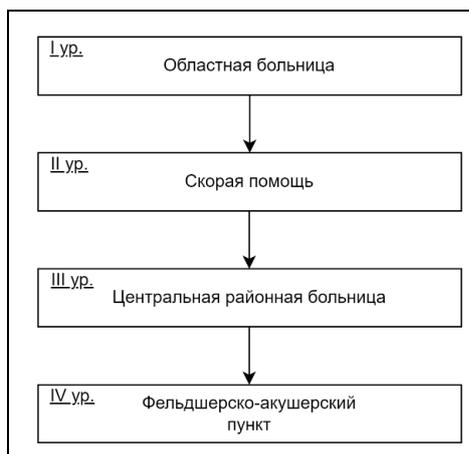


Рисунок 3 — Структура подчиненности организаций по уровням

Администраторам организаций первого и второго уровней не доступны функции добавления и редактирования организаций.

Администраторы организаций третьего уровня могут добавлять организации четвертого уровня.

#### 2.2.4. Модуль обработки данных

Модуль обработки данных предназначен для выполнения следующих функций:

- сбор измеряемых показателей;
- хранение измеряемых показателей;
- агрегация данных;
- анализ данных;
- интерпретирование значений собранных показателей;
- предоставление внешних интеграционных интерфейсов для доступа к функции сбора и интерпретации данных.

#### 2.2.5. Модуль визуализации физиологических показателей пациентов

Модуль предназначен для построения и вывода на экран графика из данных замеров показателей здоровья пациента.

#### 2.2.6. Модуль отчетности

Модуль отчетности предназначен для выполнения следующих функций:

- хранение шаблонов отчетов;
- формирование отчетов по измеряемым параметрам;
- формирование отчетов по логам;
- формирование протоколов инструментальных исследований;
- предоставление внешних интеграционных интерфейсов для загрузки новых/изменения существующих шаблонов отчетов
- предоставление внешних интеграционных интерфейсов для формирования отчетов;
- предоставление внешних интеграционных интерфейсов для получения ранее сформированных отчетов.

### **2.2.7. Модуль нотификации**

Модуль предназначен для выполнения следующих функций:

- отправка уведомлений на адрес электронной почты пользователя с ролью «Врач» о выходе измеренных показателей за установленные границы норм;
- отправка уведомлений о срочном осмотре в мобильное приложение;
- отправка уведомлений посредством внешних интеграционных интерфейсов о выходе измеренных показателей за установленные границы норм.

### **2.2.8. Модуль авторизации пользователей**

Модуль предназначен для выполнения следующих функций:

- авторизация пользователей;
- идентификация пользователей;
- аутентификация пользователей;
- разграничении прав пользователей по ролям в зависимости от уровня организации.

## 3. МЕТОДЫ API

### 3.1. Вход в систему

Авторизация осуществляется в два этапа:

1. Получение временного токена.
2. Получение сессионного токена с использованием временного токена. Временный токен необходимо отправить в заголовке **Key**.

#### Методы

1. Получить временный токен, время жизни 30 сек:

#### GET /api/SingleUsingToken

Ответ: 200 Ok

```
{
  "@odata.context":
"http://localhost:7010/api/$metadata#Smuit.TokenApi",
  "token": "eyJ0eXAiOiJKV1..."
}
```

2. Получить постоянный токен:

#### POST /api/SingleUsingTokenLogin

Headers:

- Key:<временный токен>

При этом создается пользователь если его не существует по **externalId**.

#### Тело запроса:

```
{
  "externalId": "string", // Идентификатор пользователя из
внешней системы
  "organizationId": "string" // Идентификатор организации из
внешней системы
}
```

Ответ: 200 Ok

```
{
  "@odata.context":
"http://localhost:7010/api/$metadata#Smuit.TokenApi",
  "token": "eyJ0eXAi...",
  "id": 0, // Внутренний идентификатор пользователя
  "externalId": "123" // Идентификатор пользователя внешней
системы
}
```

## 3.2. Создание программы мониторинга

Существует два метода для добавления программ мониторинга:

1. Базовый метод.
2. Расширенный метод, был создан для обеспечения доп. данных по мониторингу.

### 3.2.1. Базовый метод

#### POST <base>/api/Inspections

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

**Тело запроса:**

```
{
  "id": "string", // Внешний идентификатор мониторинга
  "patientId": "string", // Внешний идентификатор пациента, если
  пациента не существует, он будет создан
  "doctorId": "string", // Внешний идентификатор доктора
  "organizationId": "string", // Внешний идентификатор
  организации, если организации не существует, будет создана
  "organizationName": "string", // Необязательный параметр т.к.
  если организация уже существует, будет привязана она
  "deviceId": "string", // Внешний идентификатор устройства, если
  устройства не существует, будет создан
  "deviceType": 1, // Тип устройства, 1 - тонометр, 2 - глюкометр
  "startDate": "2023-03-13T11:02:48.580Z",
  "endDate": "2023-03-23T11:02:48.580Z" // Необязательный атрибут
}
```

**Ответ 200 Ok:**

```
{
  "@odata.context":
  "http://localhost:7010/api/$metadata#Inspections/$entity",
  "id": "string",
  "patientId": "string",
  "doctorId": "string",
  "organizationId": "string",
  "organizationName": "string",
  "deviceId": "string",
  "deviceType": "string",
  "startDate": "2023-03-13T11:02:48.580Z",
  "endDate": "2023-03-23T11:02:48.580Z"
}
```

Идентификатор запроса приходит в заголовке **Request-Id**.

### 3.2.2. Расширенный метод

#### POST <base>/api/v2/Inspections

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

**Тело запроса:**

```
{
  "id": "string",
  "startDate": "2022-05-18T12:21:15.000Z",
  "endDate": "2022-06-18T12:21:15.000Z",
  "requiredMeasurementsTimes":
  [
    {
      "dayPart": "morning",
      "measurementsNumber": 0,
      "repetitionsNumber": 0
    },
    {
      "dayPart": "afternoon",
      "measurementsNumber": 0,
      "repetitionsNumber": 0
    },
    {
      "dayPart": "evening",
      "measurementsNumber": 0,
      "repetitionsNumber": 0
    }
  ],
  "device": {
    "id": "string",
    "type": 1 //1 - тонометр, 2 - глюкометр
  },
  "patient": {
    "id": "string",
    "firstName": "string",
    "lastName": "string",
    "middleName": "string",
    "birthDate": "2000-04-27",
    "height": 0,
    "weight": 0,
    "sex": "male",
    "anamnesis": "string",
    "diagnosis": "string",
  }
}
```

```

    "phone": "string"
  },
  "doctor": {
    "id": "string",
    "lastName": "string",
    "firstName": "string",
    "middleName": "string"
  },
  "organization": {
    "id": "string",
    "name": "string",
    "phone": "string",
    "email": "string"
  },
  "individualMeasurementLevels": [
    {
      "measurementType": "BloodPressureSystolic",
      "standardValue": "110",
      "minValue": "90",
      "maxValue": "140"
    },
    {
      "measurementType": "BloodPressureDiastolic",
      "standardValue": "80",
      "minValue": "60",
      "maxValue": "90"
    },
    {
      "measurementType": "Pulse",
      "standardValue": "80",
      "minValue": "60",
      "maxValue": "90"
    }
  ]
}

```

**ОТВЕТ 200 Ok:**

```

{
  "@odata.context": "/api/$metadata#Inspections/$entity",
  "id": "string",
  "patientId": "string",
  "doctorId": "string",
  "organizationId": "string",
  "organizationName": "string",
  "deviceId": "string",
  "deviceType": "string",
  "status": "string",
  "startDate": "2022-05-18T12:21:15.000Z",
  "endDate": "2025-06-18T12:21:15.000Z"
}

```

### 3.3. Изменение программы мониторинга

Методы для изменения программы мониторинга присутствуют в двух вариантах:

- Метод **PATCH** `<base>/api/Inspections(<id>)` — можно изменить все необходимые данные через один запрос. Модель позволяет указывать неполные данные.
- **POST** `<base>/api/Inspections(<id>)/<Название экшена>` — можно изменить определенную часть данных - доктора, лимиты, и пр.

#### 3.3.1. Изменение программы мониторинга

##### **PATCH** `<base>/api/Inspections(<id>)`

Например:

**PATCH** `<base>/api/Inspections('11f47ad5-7b73-42c0-abae-878b1e16adee')`

Headers:

- Authorization:Bearer `<token>`
- Request-Id:`<идентификатор>`

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

Данные в теле опциональны — будут заменены только те, что указаны.

**Тело запроса:**

```
{
  "startDate": "2022-05-18T12:21:15.000Z",
  "endDate": "2022-06-18T12:21:15.000Z",
  "requiredMeasurementsTimes":
  [
    {
      "dayPart": "morning",
      "measurementsNumber": 0,
      "repetitionsNumber": 0
    },
    {
      "dayPart": "afternoon",
      "measurementsNumber": 0,
      "repetitionsNumber": 0
    },
    {
      "dayPart": "evening",
      "measurementsNumber": 0,
      "repetitionsNumber": 0
    }
  ],
  "device": {
    "id": "string",
    "type": 1
  }
}
```

```

    },
    "patient": {
      "id": "string",
      "firstName": "string",
      "lastName": "string",
      "middleName": "string",
      "birthDate": "2000-04-27",
      "height": 0,
      "weight": 0,
      "sex": "male",
      "anamnesis": "string",
      "diagnosis": "string",
      "phone": "string"
    },
    "doctor": {
      "id": "string",
      "lastName": "string",
      "firstName": "string",
      "middleName": "string"
    },
    "organization": {
      "id": "string",
      "name": "string",
      "phone": "string",
      "email": "string"
    },
    "individualMeasurementLevels": [
      {
        "measurementType": "BloodPressureSystolic",
        "standardValue": "110",
        "minValue": "90",
        "maxValue": "140"
      },
      {
        "measurementType": "BloodPressureDiastolic",
        "standardValue": "80",
        "minValue": "60",
        "maxValue": "90"
      },
      {
        "measurementType": "Pulse",
        "standardValue": "80",
        "minValue": "60",
        "maxValue": "90"
      }
    ]
  }
}

```

**Ответ 200 Ok:**

```

{
  "@odata.context": "/api/$metadata#Inspections/$entity",

```

```

    "id": "string",
    "patientId": "string",
    "doctorId": "string",
    "organizationId": "string",
    "organizationName": "string",
    "deviceId": "string",
    "deviceType": "string",
    "status": "string",
    "startDate": "2022-05-18T12:21:15.000Z",
    "endDate": "2025-06-18T12:21:15.000Z"
  }

```

### 3.3.2. Изменение даты окончания мониторинга

#### POST <base>/api/Inspections(<id>)/UpdateEndDate

Например:

POST <base>/api/Inspections('11f47ad5-7b73-42c0-abae-878b1e16adee')/UpdateEndDate

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

#### Тело запроса:

```

{
  "endDate": "2023-07-01T15:00:00.000Z"
}

```

#### Ответ 200 Ok:

```

{
  "id": "string",
  "patientId": "string",
  "doctorId": "string",
  "organizationId": "string",
  "organizationName": "string",
  "deviceId": "string",
  "deviceType": "string",
  "status": "string",
  "startDate": "2022-05-18T12:21:15.000Z",
  "endDate": "2025-06-18T12:21:15.000Z"
}

```

### 3.3.3. Изменение данных о количестве необходимых измерений за период

#### POST <base>/api/Inspections(<id>)/UpdateRequiredMeasurementTime

Например:

POST <base>/api/Inspections('11f47ad5-7b73-42c0-abae-878b1e16adee')/UpdateRequiredMeasurementTime

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

**Тело запроса:**

```
{
  "dayPart": "morning",
  "measurementsNumber": 0,
  "repetitionsNumber": 0
}
```

**Ответ 200 Ok:**

```
{
  "dayPart": "morning",
  "measurementsNumber": 0,
  "repetitionsNumber": 0
}
```

### 3.3.4. Изменение граничных значений измерений пользователя

**POST <base>/api/Inspections(<id>)/UpdateMeasurementLevel**

Например:

**POST <base>/api/Inspections('11f47ad5-7b73-42c0-abae-878b1e16adee')/UpdateMeasurementLevel**

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

**Тело запроса:**

```
{
  "measurementType": "BloodPressureSystolic",
  "standardValue": "110",
  "minValue": "90",
  "maxValue": "140"
}
```

**Ответ 200 Ok:**

```
{
  "measurementType": "BloodPressureSystolic",
  "standardValue": "110",
  "minValue": "90",
  "maxValue": "140"
}
```

### 3.3.5. Изменение параметров доктора

#### POST <base>/api/Inspections(<id>)/UpdateDoctor

Например:

POST <base>/api/Inspections('11f47ad5-7b73-42c0-abae-878b1e16adee')/UpdateDoctor

#### Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

#### Тело запроса:

```
{
  "id": "string", //Идентификатор доктора
  "firstName": "string",
  "lastName": "string",
  "middleName": "string"
}
```

#### Ответ 200 Ok

### 3.4. Снятие с мониторинга

Статус мониторинга меняется с **Active** на **Completed**

#### POST <base>/api/Inspections(ExternalId=<id>)/Complete

где id — идентификатор мониторинга.

Например:

#### POST <base>/api/Inspections(ExternalId=123)/Complete

#### Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

#### Ответ 200 OK

```
{
  "@odata.context":
"http://localhost:7010/api/$metadata#Inspections/$entity",
  "id": "string",
  "patientId": "string",
  "doctorId": "string",
  "organizationId": "string",
  "organizationName": "string",
}
```

```
"deviceId": "string",
"deviceType": "string",
"startDate": "2023-03-13T11:02:48.580Z",
"endDate": "2023-03-23T11:02:48.580Z"
}
```

Идентификатор запроса приходит в заголовке **Request-Id**.

### Ответ 404 NotFound

Если не найден мониторинг по данному Id.

### 3.5. Получение программы мониторинга

Запрос поддерживает OData, т.о. можно добавлять в запрос фильтры, пагинацию, сортировку, вывести счетчик.

Например:

**GET /api/Inspections?\$filter=doctorId eq '123'&top=10&skip=20&count=true**

**GET /api/Inspections**

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

### Ответ 200 ОК:

```
{
  "@odata.context": "/api/$metadata#Inspections",
  "value": [
    {
      "id": "string",
      "patientId": "string",
      "doctorId": "string",
      "organizationId": "string",
      "organizationName": "string",
      "deviceId": "string",
      "deviceType": "tonometer",
      "status": "active",
      "startDate": "2023-03-12T20:00:00Z",
      "endDate": "2023-03-28T20:00:00Z"
    }
  ]
}
```

### 3.6. Добавление измерения

#### POST <base>/api/AddMeasurements

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

**Тело запроса:**

```
{
  "inspectionId": "string",
  "dateTime": "2023-04-05T10:30:45.000Z",
  "measurements": [
    {
      "type": "BloodPressure",
      "value": "120/80"
    },
    {
      "type": "Pulse",
      "value": "79"
    }
  ]
}
```

**Типы измерений:**

- BloodPressure

В этом случае систолическое и диастолическое давления должны приходить в формате <сист>/<диаст>

- Pulse

**Ответ 200 Ok**

Идентификатор запроса приходит в заголовке **Request-Id**.

### 3.7. Получение ссылки на страницу пациента

#### GET api/Patients(<id пациента>)/GetCardLink

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

### Ответ 200 ОК:

```
{
  "patientCardLink": "<web url>/patient/<id пациента>"
}
```

### 3.8. Изменение профиля пациента

#### PATCH <base>/api/Patients

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

#### Тело запроса:

```
{
  "id": "string",
  "firstName": "string",
  "lastName": "string",
  "middleName": "string",
  "birthDate": "2000-04-27",
  "height": 0,
  "weight": 0,
  "sex": "male",
  "anamnesis": "string",
  "diagnosis": "string",
  "phone": "string"
}
```

### Ответ 200 ОК:

```
{
  "id": "string",
  "firstName": "string",
  "lastName": "string",
  "middleName": "string",
  "birthDate": "2000-04-27",
  "height": 0,
  "weight": 0,
  "sex": "male",
  "anamnesis": "string",
  "diagnosis": "string",
  "phone": "string"
}
```

### Ответ 400 BadRequest:

- Если id не указан.

**Ответ 404 NotFound:** Если пользователь по идентификатору не найден.

### 3.9. Изменение профиля доктора

#### PATCH <base>/api/Doctors

Headers:

- Authorization:Bearer <token>
- Request-Id:<идентификатор>

Метод может использоваться для синхронизации запроса и ответа, отправленный заголовок так же возвращается в ответе в заголовке **Request-Id**.

**Тело запроса:**

```
{
  "id": "string",
  "firstName": "string",
  "lastName": "string",
  "middleName": "string",
}
```

**Ответ 200 ОК:**

```
{
  "id": "string",
  "firstName": "string",
  "lastName": "string",
  "middleName": "string",
}
```

**Ответ 400 BadRequest:**

- Если id не указан.

#### 4. СТРУКТУРА БАЗЫ ДАННЫХ

Логическая структура базы данных спроектирована и реализована для СУБД PostgreSQL.

В базе данных содержатся следующие таблицы:

1) user — данные пользователей

Название поля	Описание	Тип поля	Параметры
id	Первичный ключ, уникальный идентификационный номер	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
created	Время создания записи	TIMESTAMP	NOT NULL
username	Учетное имя пользователя; при регистрации в поле пишется email, если это доктор, если это пациент с мобильного приложения, то в это поле пишется сгенерированное значение из его ФИО и других данных	VARCHAR(50)	NULL UNIQUE
email	Электронный адрес пользователя	VARCHAR(50)	NULL UNIQUE
phone	Номер телефона пользователя	VARCHAR(20)	
phone_trustee	Номер телефона доверенного лица пациента	VARCHAR(20)	
snils	СНИЛС	VARCHAR(20)	NULL UNIQUE
org_id	Уникальный идентификационный номер организации, к которой относится пользователь	BIGINT	NULL
first_name	Имя	VARCHAR(50)	
middle_name	Отчество	VARCHAR(50)	
last_name	Фамилия	VARCHAR(50)	
sex	Пол	VARCHAR(10)	DEFAULT 'male'
birthdate	Дата рождения	TIMESTAMP	
avatar	Фотография	VARCHAR(255)	
height	Рост	SMALLINT	
weight	Вес	NUMERIC(10, 2)	
status	Статус, может быть активным или в архиве	VARCHAR(30)	DEFAULT 'base.status.active'
password_hash	Хэш пароля	VARCHAR(255)	NOT NULL
role_name	Роль в системе	VARCHAR(30)	NOT NULL REFERENCES roles (role_name)

2) organizations — медицинские организации

Название поля	Описание	Тип поля	Параметры
id	Первичный ключ, уникальный идентификационный номер	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
created	Дата и время создания записи	TIMESTAMP	NOT NULL
id_ext	Идентификатор сторонней организации, к которой привязана текущая, которая находится не в нашей системе (сейчас есть возможность указать при создании, но никак не используется)	VARCHAR(50)	NULL UNIQUE
name	Название организации	VARCHAR(255)	

address	Адрес организации	VARCHAR(255)	
email	Электронный адрес	VARCHAR(50)	NULL UNIQUE
phone	Номер телефона	VARCHAR(50)	
license	Номера договора	VARCHAR(255)	
status	По умолчанию всегда «active»	VARCHAR(50)	
parent_id	Уникальный идентификационный ключ вышестоящей в иерархии организации	BIGINT	NULL REFERENCES organizations (id) DEFAULT NULL
level	Уровень в структуре подчиненности медицинских организаций: 0 — областная больница, 1 — скорая помощь, 2 — центральная районная больница, 3 — фельдшерско-акушерский пункт	INT	NOT NULL DEFAULT 0

### 3) measurements — измерения

Название поля	Описание	Тип поля	Параметры
id	Первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
created	дата и время создания записи	TIMESTAMP	NOT NULL
user_id	Уникальный идентификационный номер пользователя (пациента)	BIGINT	REFERENCES users(id)
source	Прибор, «источник» измерения (glucosemeter, thermometer, tonometer и др.)	VARCHAR(30)	NOT NULL
type	Тип измерения	VARCHAR(255)	NOT NULL
datetime	Дата и время выполнения измерения. Приходит вместе с измерением с прибора.	TIMESTAMP	NOT NULL
value	значение измерения	VARCHAR(255)	NOT NULL
deleted	Статус удаления измерения. При удалении, запись не пропадает из БД, а помечается данным флагом как “true”	BOOLEAN	NOT NULL DEFAULT false
need_inspection	Статус «срочного осмотра». Устанавливается врачом, если он решит, что пациенту владельцу измерения требуется срочный осмотр	BOOLEAN	NOT NULL DEFAULT false

### 4) measurement\_params — параметры измерений

Название поля	Описание	Тип поля	Параметры
id	Первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
created	Дата и время создания записи	TIMESTAMP	NOT NULL
user_id	Уникальный идентификационный номер пользователя (пациента)	BIGINT	REFERENCES users(id)
measurement_id	Уникальный идентификационный номер измерения	BIGINT	REFERENCES measurements(id)
type	Тип измерения	VARCHAR(30)	NOT NULL
value	Значение измерения	VARCHAR(255)	NOT NULL

5) `recovered_passwords` — восстановленные пароли

Название поля	Описание	Тип поля	Параметры
<code>id</code>	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
<code>user_id</code>	уникальный идентификационный номер пользователя	BIGINT	REFERENCES users (id) UNIQUE NOT NULL
<code>password_hash</code>	хэш пароля	VARCHAR(255)	NOT NULL
<code>expiration_time</code>	Дата и время истечения срока, в течение которого восстановленный пароль валиден и может быть использован для первого входа в систему после чего будет установлен в качестве основного	TIMESTAMP	NOT NULL

6) `users_params` — используется для задания дополнительных параметров пользователей

Название поля	Описание	Тип поля	Параметры
<code>id</code>	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
<code>created</code>	дата и время создания записи	TIMESTAMP	NOT NULL
<code>user_id</code>	уникальный идентификационный номер пользователя	BIGINT	REFERENCES users (id)
<code>type</code>	Тип параметра (сейчас 3 распространенных: UrgentInspection, mod.meals.tags.enable, heartInterval)	VARCHAR(50)	NOT NULL
<code>value</code>	Значение параметра	VARCHAR(255)	NOT NULL
<code>source</code>	Источник, сейчас задается как «app»	VARCHAR(30)	default " NOT NULL

7) `diagnoses` — диагнозы

Название поля	Описание	Тип поля	Параметры
<code>id</code>	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
<code>created</code>	дата и время создания записи	TIMESTAMP	NOT NULL
<code>code</code>	код болезни по МКБ-10	VARCHAR(10)	UNIQUE NOT NULL
<code>description</code>	Описание (например: «Болезни нервной системы»)	VARCHAR(255)	DEFAULT "

8) `measurements_comments` — комментарии к измерениям

Название поля	Описание	Тип поля	Параметры
<code>id</code>	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
<code>created</code>	дата и время создания записи	TIMESTAMP	NOT NULL
<code>user_id</code>	уникальный идентификационный номер пользователя (пациента)	BIGINT	REFERENCES users (id)
<code>measurement_id</code>	Идентификтор измерения для которого был оставлен комментарий	BIGINT	REFERENCES measurements (id)
<code>value</code>	Сам комментарий	VARCHAR(255)	NOT NULL
<code>updated</code>	Дата-время изменения(обновления) комментария	TIMESTAMP	NULL

9) `notifications` — уведомления

Название поля	Описание	Тип поля	Параметры
---------------	----------	----------	-----------

id	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
created	дата и время создания записи	TIMESTAMP	NOT NULL
is_read	Статус прочтения уведомления (true/false)	BOOLEAN	NOT NULL DEFAULT false
recipient_id	уникальный идентификационный номер пользователя (врач, получатель уведомления)	BIGINT	REFERENCES users (id)
message	Текст уведомления	VARCHAR(500)	NOT NULL
deleted	Аналогично с измерениями, уведомления не удаляются из базы данных, а помечаются данным флагом как true и больше не отображаются	BOOLEAN	NOT NULL DEFAULT false
title	Заголовок уведомления, в отображении идет перед текстом уведомления	VARCHAR(50)	NOT NULL DEFAULT "
type	Событие, инициировавшее отправку уведомления. Например: Выход измерения за лимиты, активация срочного осмотра.	VARCHAR(50)	NOT NULL DEFAULT "

10) **urgent\_inspections** — отображает историю постановки и снятия срочных осмотров

Название поля	Описание	Тип поля	Параметры
id	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
created	дата и время создания записи	TIMESTAMP	NOT NULL
initiator_id	уникальный идентификационный номер пользователя (врача)	BIGINT	REFERENCES users (id)
patient_id	уникальный идентификационный номер пользователя (пациента)	BIGINT	REFERENCES users (id)
measurement_id	уникальный идентификационный номер измерения	BIGINT	REFERENCES measurements (id)
action	Действие относительно срочного осмотра (постановка/снятие)	VARCHAR(10)	NOT NULL

11) **measurements\_tracking** — просмотр измерений

Название поля	Описание	Тип поля	Параметры
id	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
created	дата и время создания записи	TIMESTAMP	NOT NULL
initiator_id	уникальный идентификационный номер пользователя (врача)	BIGINT	REFERENCES users (id)
initiator_org_id	Идентификатор организации в которой зарегистрирован инициатор	BIGINT	REFERENCES organizations (id)
measurement_id	уникальный идентификационный номер измерения	BIGINT	REFERENCES measurements (id)

12) **measurements\_limits** — границы нормы измерений

Название поля	Описание	Тип поля	Параметры
id	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY

created	дата и время создания записи	TIMESTAMP	NOT NULL
min	значение верхней границы	NUMERIC(10, 2)	NOT NULL
max	значение нижней границы	NUMERIC(10, 2)	NOT NULL
measurement_type	тип измерения	VARCHAR(255)	NOT NULL
user_id	уникальный идентификационный номер пользователя (пациента)	BIGINT	REFERENCES users (id)
initiator_id	уникальный идентификационный номер пользователя (врача)	BIGINT	REFERENCES users (id)

### 13) default\_limits — границы измерений по умолчанию

Название поля	Описание	Тип поля	Параметры
id	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
created	дата и время создания записи	TIMESTAMP	NOT NULL
min	значение верхней границы	NUMERIC(10, 2)	NOT NULL
max	значение нижней границы	NUMERIC(10, 2)	NOT NULL
measurement_type	тип измерения	VARCHAR(255)	NOT NULL

### 14) subscriptions — таблица для хранения подписок докторов на нотифицируемые события

Название поля	Описание	Тип поля	Параметры
id	первичный ключ, уникальный идентификационный номер записи	BIGINT	UNIQUE GENERATED BY DEFAULT AS IDENTITY
created	дата и время создания записи	TIMESTAMP	NOT NULL
publisher_id	Идентификатор пользователя (пациента), на события которого установлена подписка (события такие как выход за лимиты измерений)	BIGINT	NOT NULL REFERENCES users (id)
subscriber_id	Идентификатор пользователя (доктора), который подписался на получение уведомлений от publisher_id и получит уведомление в случае наступления события	BIGINT	NOT NULL REFERENCES users (id)
event_type	Тип события. Здесь указывается тип измерений, на уведомления о выходе за нормы которых была подписка	VARCHAR(50)	NOT NULL
subscribed	Статус подписки (true/false). Когда врач подписывается на событие становится true	BOOLEAN	NOT NULL DEFAULT true );

### 15) roles — роли пользователей в системе

Название поля	Описание	Тип поля	Параметры
created	дата и время создания записи	TIMESTAMP	NOT NULL
role_name	название роли	VARCHAR(30)	NOT NULL