

Московский ордена Ленина, ордена Октябрьской Революции и ордена Трудового

Красного Знамени государственный технический университет имени Н. Э. Баумана

Факультет «Робототехника и комплексная автоматизация»

Кафедра «Компьютерные системы автоматизации производства»

«Документация по творческому заданию (этап№4)»

ID: 254187039

05.02.1999

[dmasny@yandex.ru](mailto:dmasny@yandex.ru)

+79778781228

Бакалавриат 4 курс

Студент: Масный Дмитрий Игоревич

Группа: РК9-84Б

**Аннотация**

В данной работе рассматривается процесс создания чат-бота telegram для управления системой мониторинга физического состояния шахтера на основе облачной платформы Rightech IoT Cloud. Приведены модели функционирования системы, разработаны алгоритмы работы чат-бота, а также проведено тестирование и продемонстрирована работа созданного бота. Приведены ссылки на видеоматериалы и репозиторий с исходным кодом.

**Ключевые слова:** чат-бот, Rightech IoT Cloud, телеметрия, система мониторинга физического состояния человека, сбор данных, matplotlib, numpy, шахта.

**Оглавление**

[**Введение 4**](#_Toc69860635)

[**1. Обзор и анализ предметной области 6**](#_Toc69860636)

[**2. Теоретическая часть 7**](#_Toc69860637)

[**2.1 Разработка функциональной модели системы управления мониторингом физического состояния рабочего 7**](#_Toc69860638)

[**2.2 Взаимодействие чат-бота с облачной платформой Rightech IoT Cloud 9**](#_Toc69860639)

[**2.3 Проектирование алгоритма подпрограммы сбора данных 9**](#_Toc69860640)

[**2.4 Проектирование алгоритма работы 10**](#_Toc69860641)

[**3. Практическая часть 12**](#_Toc69860642)

[**3.1 Создание модели системы мониторинга здоровья на платформе RIC 12**](#_Toc69860643)

[**3.2 Создание чат-бота Telegram 13**](#_Toc69860644)

[**3.3 Разработка исходного кода программ 13**](#_Toc69860645)

[**3.4 Тестирование разработанного чат-бота 13**](#_Toc69860646)

[**3.5 Перспективы развития проекта 17**](#_Toc69860647)

[**Заключение 18**](#_Toc69860648)

[**Список литературы 19**](#_Toc69860649)

# **Введение**

Одна из проблем, которая существует в горнодобывающей промышленности – низкая продолжительной жизни шахтеров. В настоящее время продолжительность жизни шахтера составляет в среднем 53 года [10], что непозволительно в 21 веке. Низкая продолжительность жизни связана с тем, что рабочие очень часто страдают хроническими заболеваниями лёгочно-дыхательной группы, заболеваниями сердечно-сосудистой системы, о наличии которых они порой не подозревают и которые трудно диагностировать при обычном медицинском осмотре перед сменой [11]. Своевременная диагностика заболевания позволяет раньше начать лечение, что часто может спасти жизнь человеку.

Существующее диагностическое медицинское оборудование не может решить данную проблему, потому что диагностика рабочего проводится стационарно либо до, либо после смены, что не дает информации о том, как ведет себя организм рабочего при нагрузках. Из-за отсутствия данных о состоянии человека во время нагрузки, зачастую невозможно дать заключение о том, страдает ли он каким-либо заболеванием на ранних этапах, на которых лечение будет максимально эффективным и вред здоровью будет минимальным, или нет. В связи с этим я предлагаю разработать систему мониторинга здоровья шахтеров в течение рабочей смены с применением технологий IoT.

В настоящее время устройства носимой электроники становятся все более популярными среди потребителей. Это связано, во-первых, с их доступностью, во-вторых, с ростом их функциональных возможностей. Современные умные-браслеты могут не только измерять пульс и сатурацию, но и давление, а иногда даже и делать ЭКГ. Кроме того, увеличение площади покрытия интернет сетями, а также увеличение скорости передачи данных по сети, позволяет собирать и обрабатывать удаленно все большие объемы данных. Объединив эти составляющие IoT и добавив возможность обработки полученных данных, можно создать систему мониторинга физического состояния рабочего в шахте. Система мониторинга включает в себя устройство сбора телеметрии (например, умный браслет), облачную платформу, для взаимодействия с данным устройством, систему управления мониторингом, которая позволяет визуализировать полученные данные и анализировать состояние здоровья человека.

Практическая значимость данной работы состоит в том, что внедрение данной системы позволит выявлять опасные заболевания среди шахтеров на ранних этапах, оказывать своевременную медицинскую помощь в случае чрезвычайных ситуаций (ухудшение состояния здоровья во время смены) на основании полученной телеметрии. Эти меры будут способствовать увеличению продолжительности жизни шахтеров.

**Объект исследования:** системы мониторинга физического состояния рабочих шахты.

**Предмет исследования:** система управления мониторингом состояния шахтера с помощью чат-бота.

В данной работе рассматривается только процесс создания системы управления мониторингом с использованием чат-бота telegram.

**Цель работы:** разработать чат-бот на платформе telegram для управления системой мониторинга физического состояния шахтера во время смены с использованием платформы Rightech IoT Cloud.

В ходе работы над проектом были поставлены следующие **задачи**:

1. Проанализировать существующие системы мониторинга физического состояния человека.
2. Выявить недостатки существующих систем.
3. Разработать функциональную модель системы управления на основе чат-бота.
4. Разработать алгоритмы проектируемых программ.
5. Разработать исходный код программ для работы чат-бота.

# **Обзор и анализ предметной области**

В настоящее время шахтеры проходят медицинский осмотр до и после смены. На некоторых предприятиях внедряются системы автоматической диагностики рабочих, как на пример на предприятии «Южный Кузбас» [11]. Данная система включает в себя терминал, манжет для измерения давления, алкотестер, тепловизор для изменения температуры тела. Внедрение данной системы позволяет сократить штат медицинский работников и ускорить процедуру медицинского осмотра. Однако, данное решение имеет ряд недостатков: описание состояния шахтера только в момент диагностирования (до, либо после смены), информация о диагностике вносится в единую БД с задержкой. Кроме этого, система требует постоянного подключения к электросети.

Предлагаемая система мониторинга позволит собирать информацию постоянно в течение рабочей смены. Устройство сбора данных представляет из себя предмет умной носимой электроники (наподобие умного-браслета), которое осуществляет сбор таких показателе, как давление, пульс, сатурация. Полученные данные будут отправляться по сети интернет на облачную платформу Rightech IoT Cloud по протоколу MQTT. Медицинский персонал шахты будет взаимодействовать с платформой с помощью чат-бота в мессенджере telegram. Система управления мониторингом разработана на основе чат-бота telegram по нескольким причинам, главными из которых являются: кроссплатформенность – чат-бот не привязывается к определенной операционной системе (на данный момент существуют клиенты как и для мобильных устройств, так и для ПК), интерфейс взаимодействия с пользователем прост и удобен («user friendly») – чат-бот имеет подсказки, которые позволяют пользователю выполнить требуемую задачу. Для разработки бота был выбран мессенджер telegram, потому что существует документация и множество библиотек для работы с telegram API [6].

# **2. Теоретическая часть**

## **2.1 Разработка функциональной модели системы управления мониторингом физического состояния рабочего**

В ходе работ по проектированию чат-бота для управления системой мониторинга физического состояния рабочего была разработана функциональная модель с помощью методологии проектирования IDEF0 (рисунок 1).

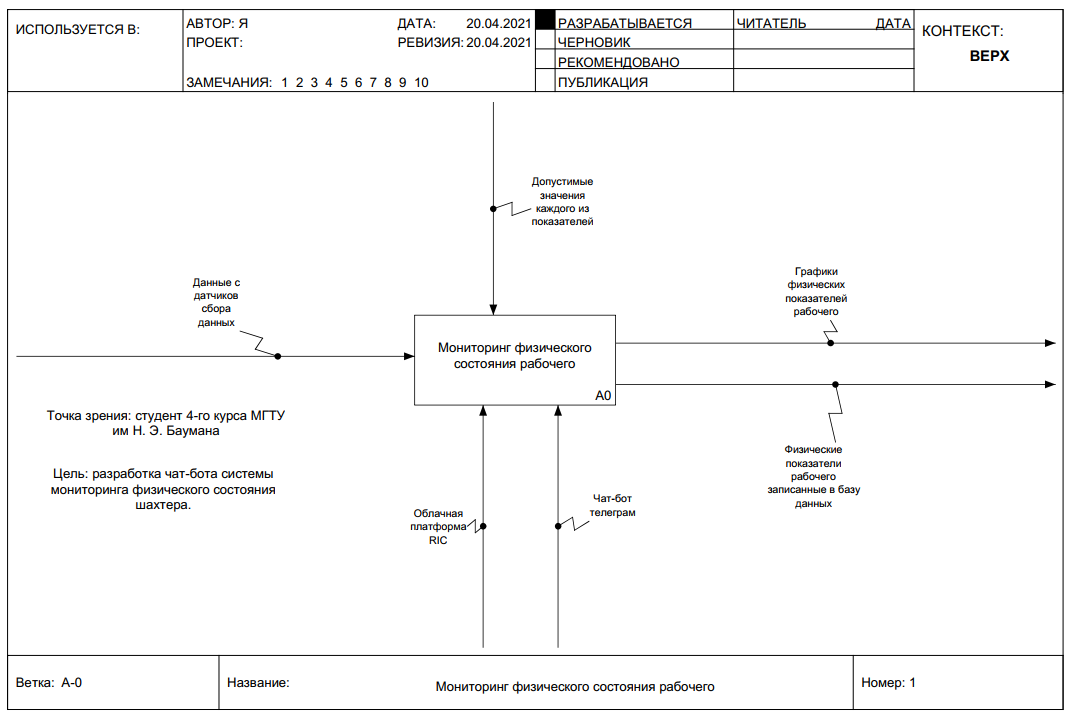


Рисунок 1

Функциональная модель отражает этапы сбора телеметрии и обработки полученных данных о состоянии здоровья шахтера, начиная с отправки get запроса на платформу RIC и заканчивая отправкой графиков показателей физического состояния рабочего в телеграмм чат врачу (рисунок 2).

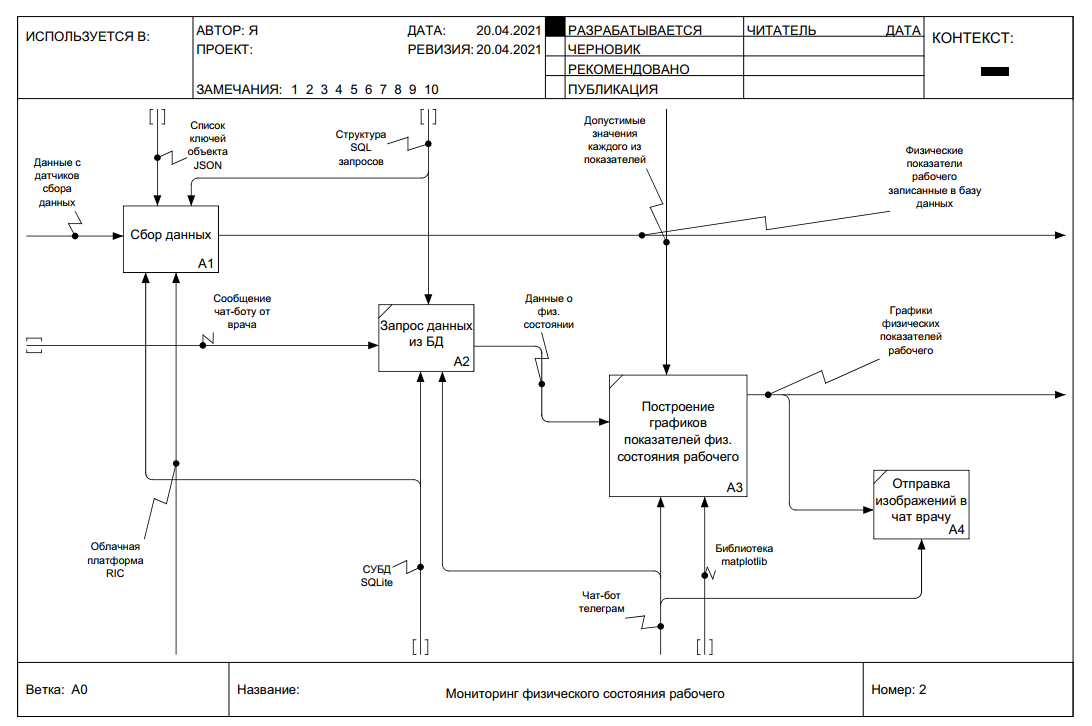


Рисунок 2

Процесс мониторинга начинается с этапа сбора данных. Устройство сбора телеметрии отправляет показания на облачную платформу RIC. Затем врач инициирует взаимодействие с чат-ботом путем отправки ему сообщения. Далее чат-бот делает запрос к базе данных, в которой хранятся данные телеметрии, полученные после этапа сбора данных. После этого строятся графики физических показателей (давления, пульса, сатурации) шахтера. Следующим шагом эти данные отправляются в чат врачу, который инициировал запрос.

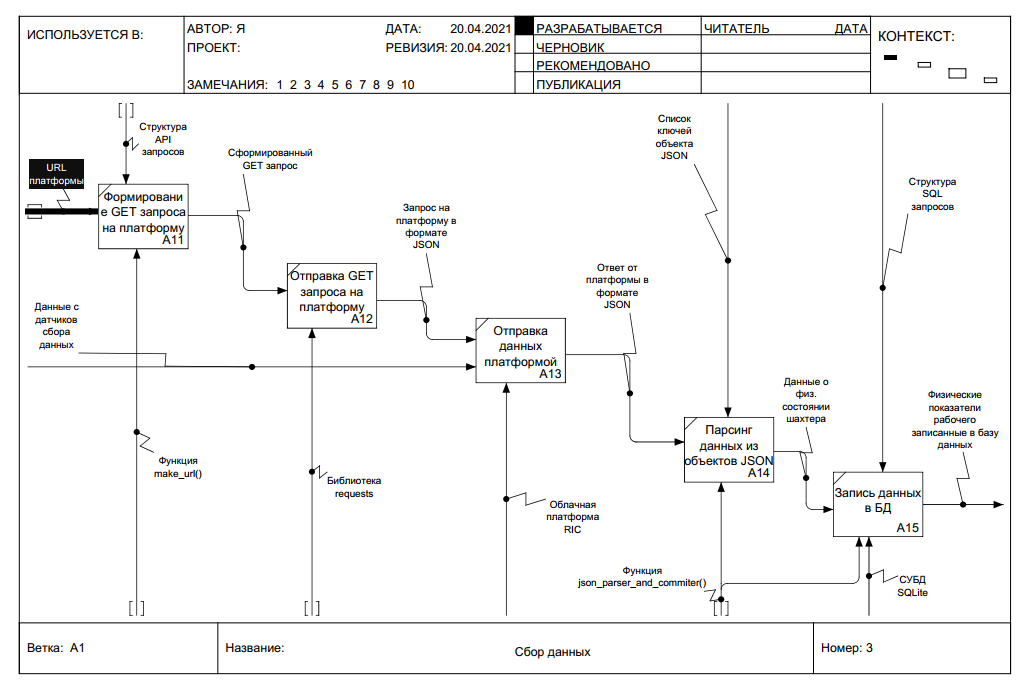


Рисунок 3

Этап сбора данных был разобран более подробно. Функциональная схема представлена на рисунке 3. Сначала формируется GET запрос на платформу. На вход поступает URL облачной платформы, в которую дописываются необходимые параметры согласно структуре API запросов доступных на платформе [8]. После этого сформированный запрос отправляется на платформу с использованием библиотеки requests [7]. После этого платформа RIC отправляет требуемые данные в формате json. Далее бот парсит эти данные и достает необходимые показатели, которые записывает с базу данных с помощью СУБД SQLite [2].

## **2.2 Взаимодействие чат-бота с облачной платформой Rightech IoT Cloud**

Взаимодействие чат-бота с облачной платформой RIC осуществляется через REST API. REST API представляет собой программный интерфейс, который позволяет использовать функционал платформы при обращении к ней из внешних сред. В данном проекте API платформы предоставляет данные об объекте «Устройство мониторинга здоровья».

Схема взаимодействия чат-бота и облачной платформы, а также облачной платформы и устройства сбора телеметрии представлена на рисунке 4.

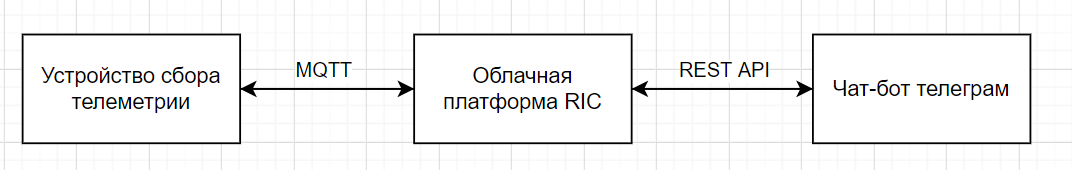
****

Рисунок 4

Стоит отметить, что в данной работе рассматривается только создание чат-бота, поэтому устройство, собирающее данные о физическом состоянии рабочего, эмулируется с помощью платформы RIC путем создания объекта, используемого в качестве бота.

## **2.3 Проектирование алгоритма подпрограммы сбора данных**

Для загрузки данных с платформы на локальный сервер (в роли которого выступает на данный момент ПК) используется программа сбора данных, алгоритм которой представлен на рисунке 5.

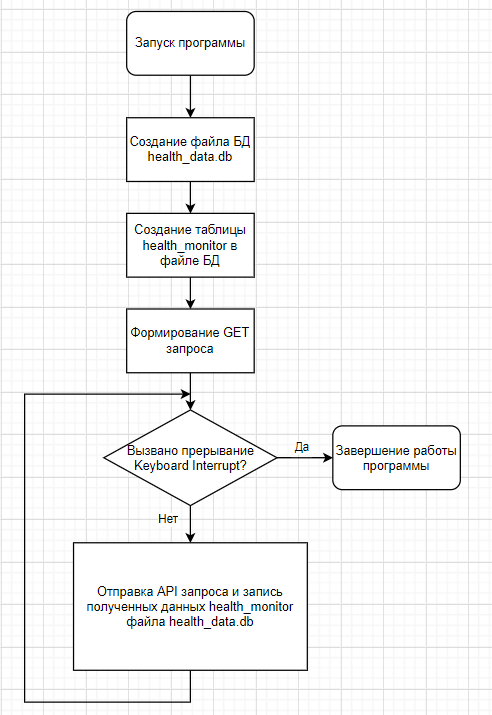


Рисунок 5

Исходный код программы разрабатывается на языке программирования Python 3 с использованием библиотек time [9], requests [7], json [3], sqlite3[2]. Данный язык программирования был выбран по ряду преимуществ, в числе которых можно отметить следующие: динамическая типизация, качественная документация. Исходный код разработанной программы находится в файле backend.py расположенном в репозитории по ссылке: <https://github.com/dmasny99/IoTpit/tree/main/case№4>.

## **2.4 Проектирование алгоритма работы**

Для выполнения требуемых функций по построению и отправке графиков в чат используется программа работы бота, алгоритм которой представлен на рисунке 6.

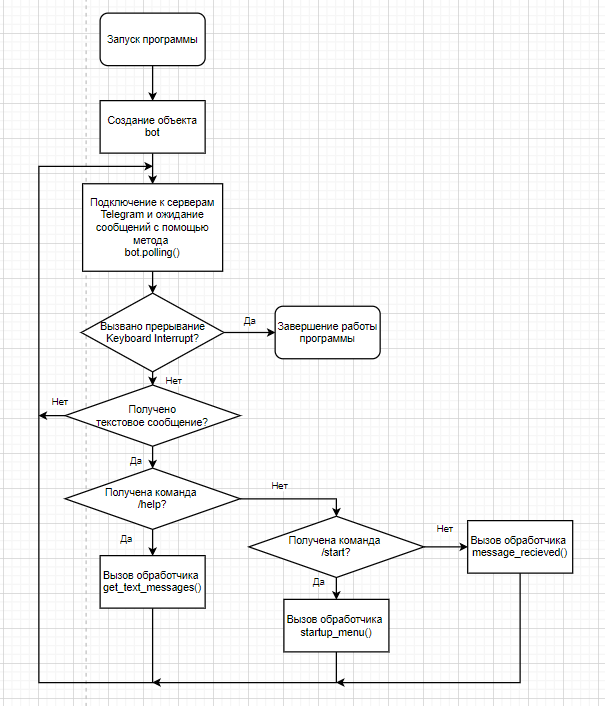


Рисунок 6

Исходный код программы разрабатывается на языке программирования Python 3 с использованием библиотек telebot [6], datetime [1], matplotlib [4], numpy [5], sqlite3 [2]. Исходный код разработанной программы находится в файле bot.py расположенном в репозитории по ссылке: <https://github.com/dmasny99/IoTpit/tree/main/case№4>.

# **3. Практическая часть**

## **3.1 Создание модели системы мониторинга здоровья на платформе RIC**

Для эмуляции устройства сбора телеметрии создается модель системы мониторинга здоровья на платформе RIC (рисунок 7). Аргументы «Пульс», «Давление», «Сатурация» типа number.

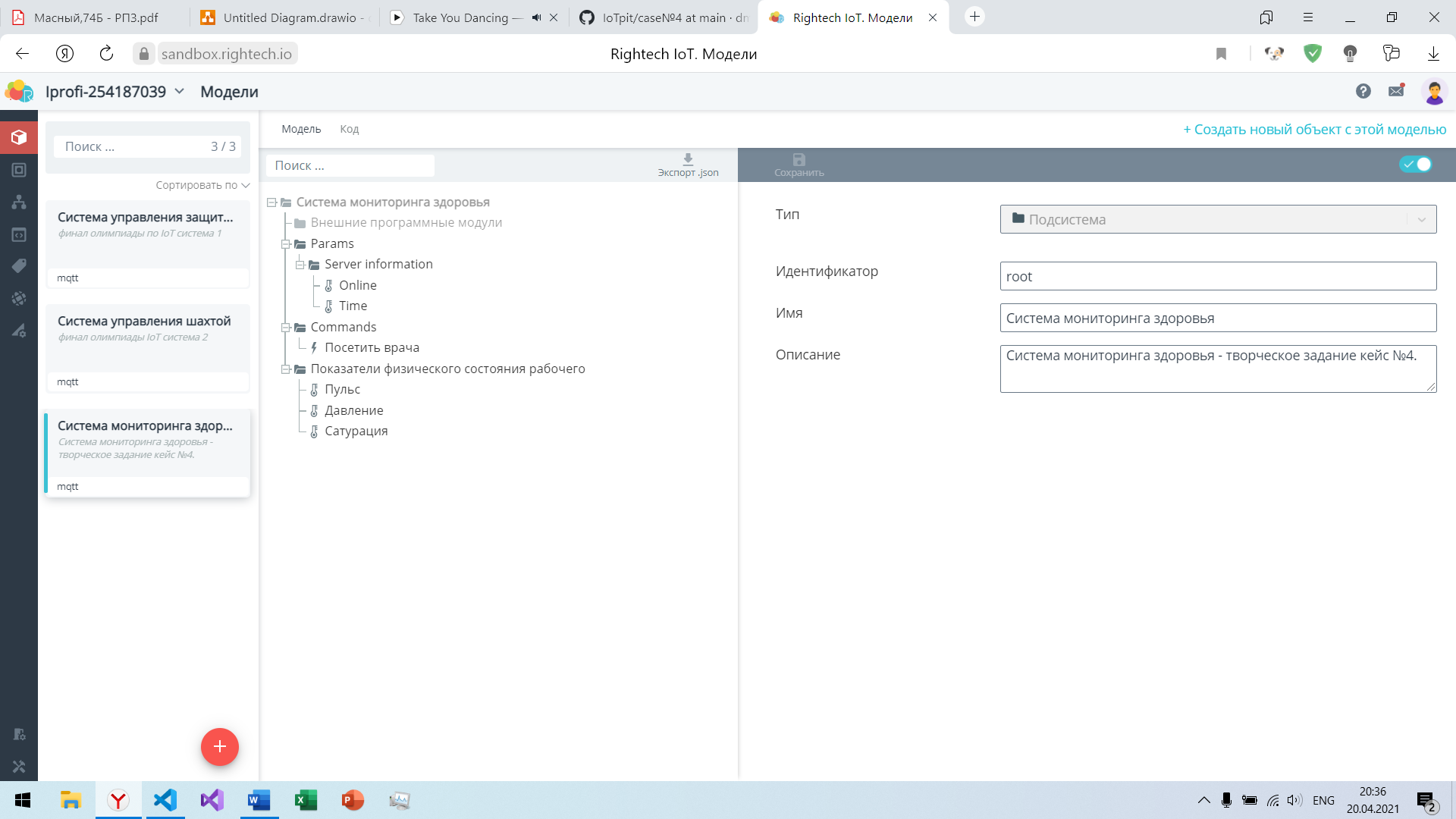
****

Рисунок 7

Далее создается объект «Устройство мониторинга здоровья» на основе ранее созданной модели и настраивается на работу в качестве бота-эмулятора. Далее настраивается автоматическая генерация параметров здоровья (рисунок 8).

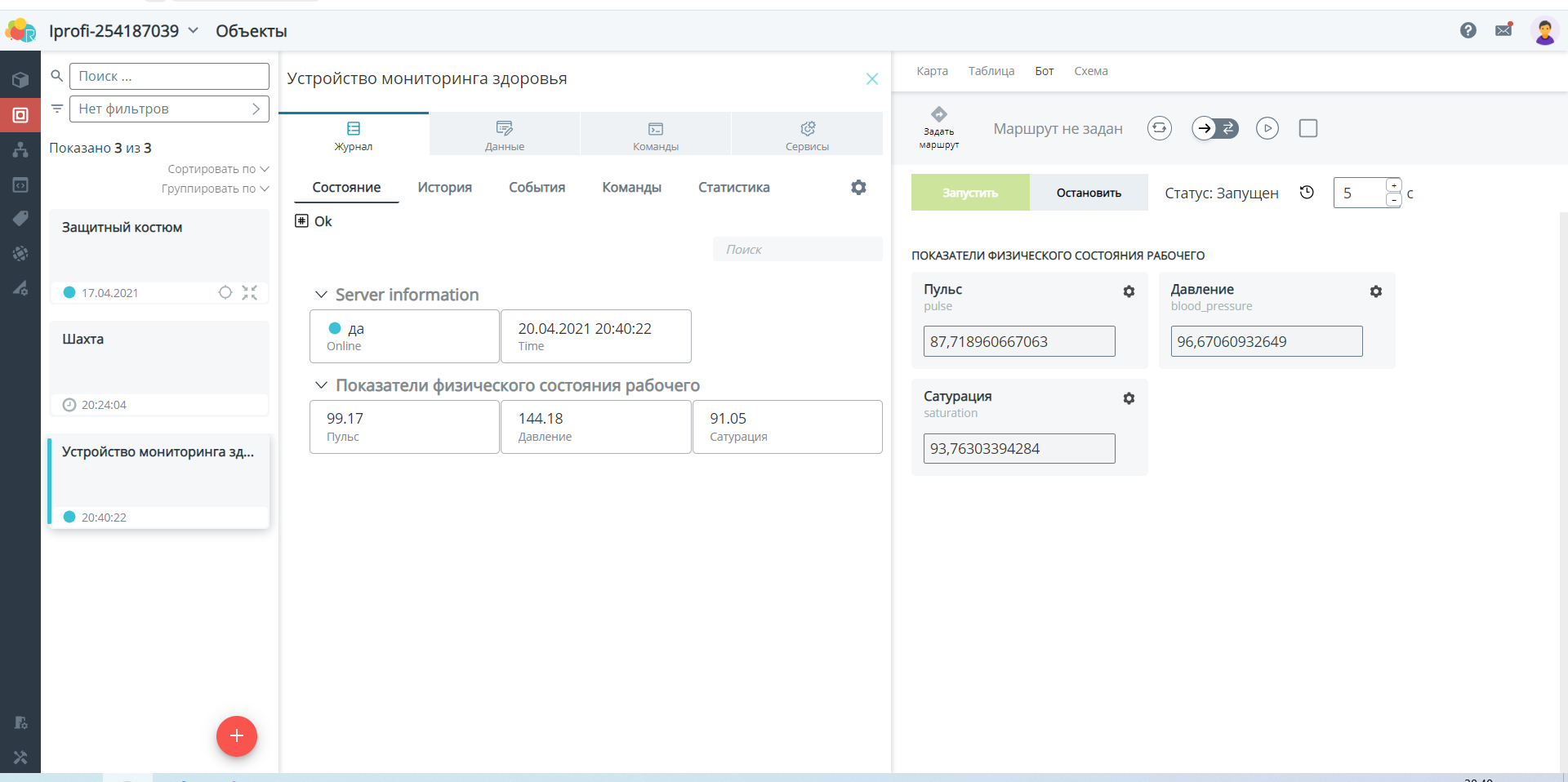


Рисунок 8

## **3.2 Создание чат-бота Telegram**

Сначала создается телеграм-бот, для этого пишется запрос боту BotFather, который помогает в настройке и запуске чат-бота (рисунок 9).

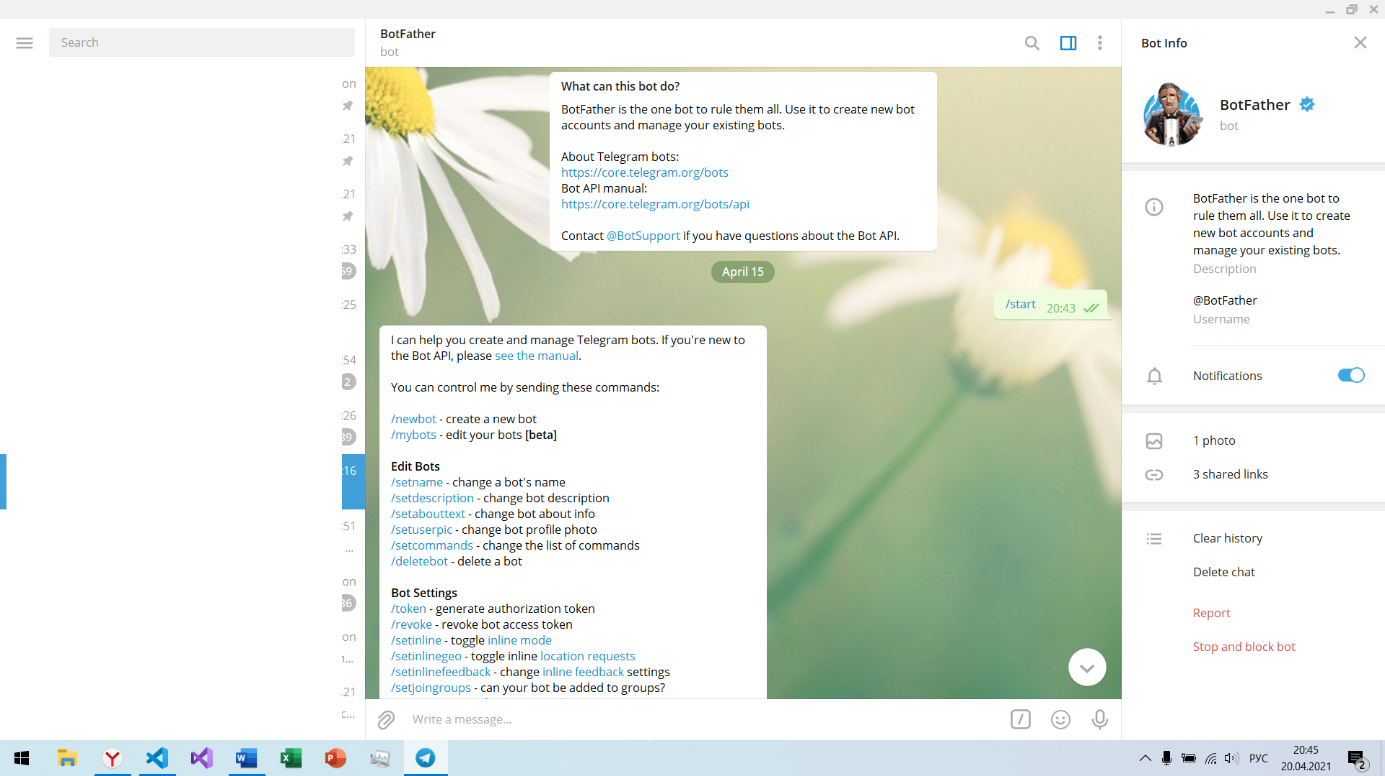


Рисунок 9

## **3.3 Разработка исходного кода программ**

Разработка исходного кода программ управления ботом ведется на основе алгоритмов с использованием библиотек, описанных в теоретической части. В результате разработки ПО были созданы две программы bot.py, backend.py , исходный код которых расположен в репозитории по ссылке: <https://github.com/dmasny99/IoTpit/tree/main/case№4>. Кроме этого, более подробно принцип работы каждой программы описывается в видеоматериалах, расположенных в следующем разделе, посвященном тестированию бота.

## **3.4 Тестирование разработанного чат-бота**

Сначала запускается созданный бот-эмулятор на платформе, в котором настраиваются значения параметров физического состояния для эмуляции (рисунок 8). После этого запускается файл backend.py. Далее запускается файл bot.py. После этого открывается Telegram Desktop, и отправляется команда «/help» боту в чате (рисунок 10).

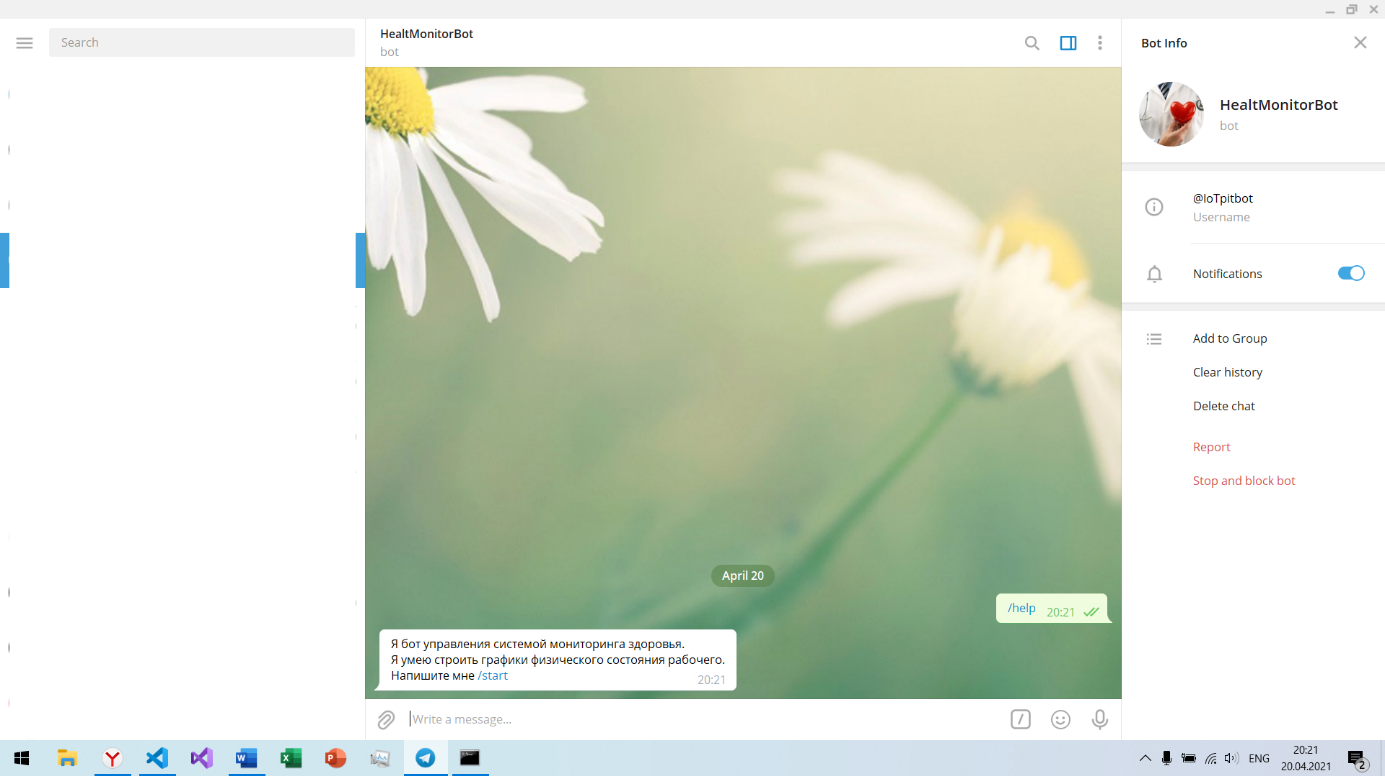


Рисунок 10

Далее отправляется команда «/start» (рисунок 11).

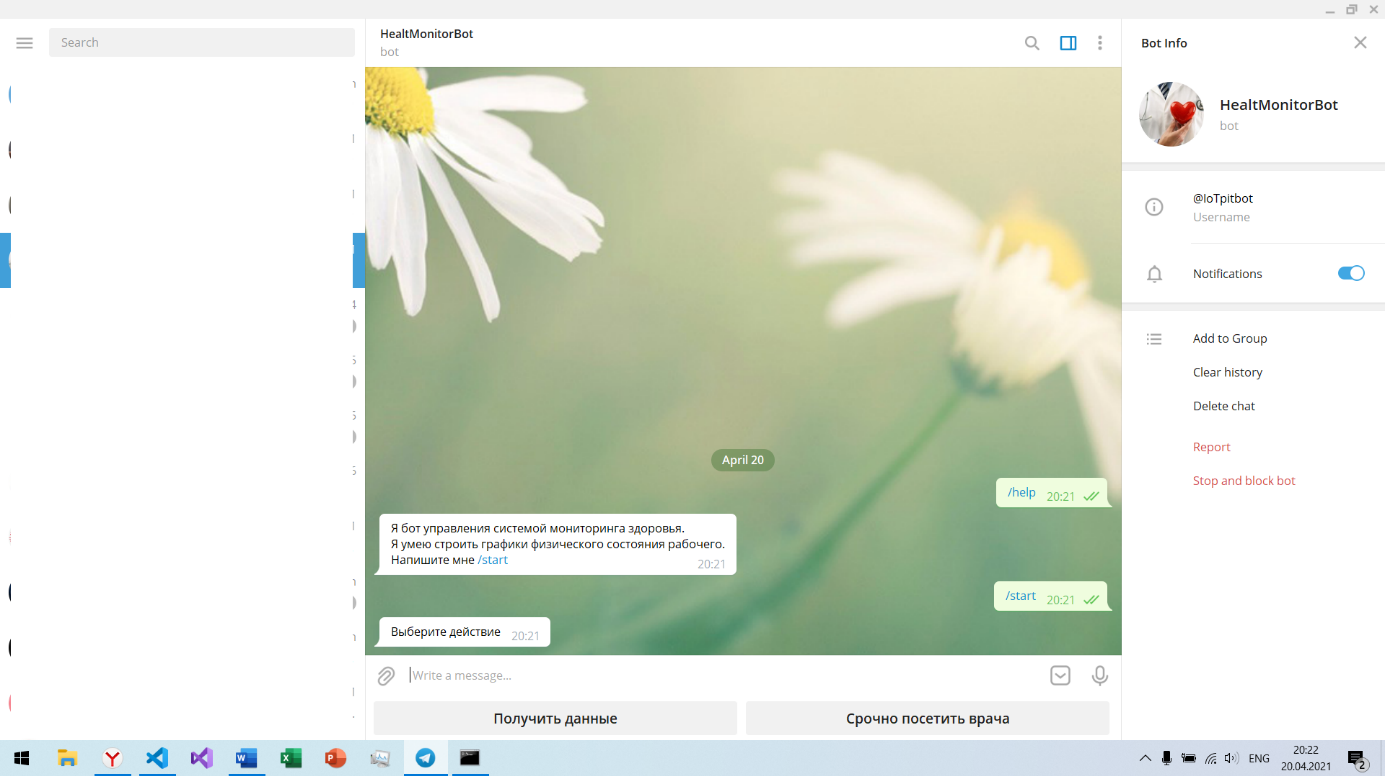


Рисунок 11

Далее нажимается кнопка «Получить данные» (рисунок 12).

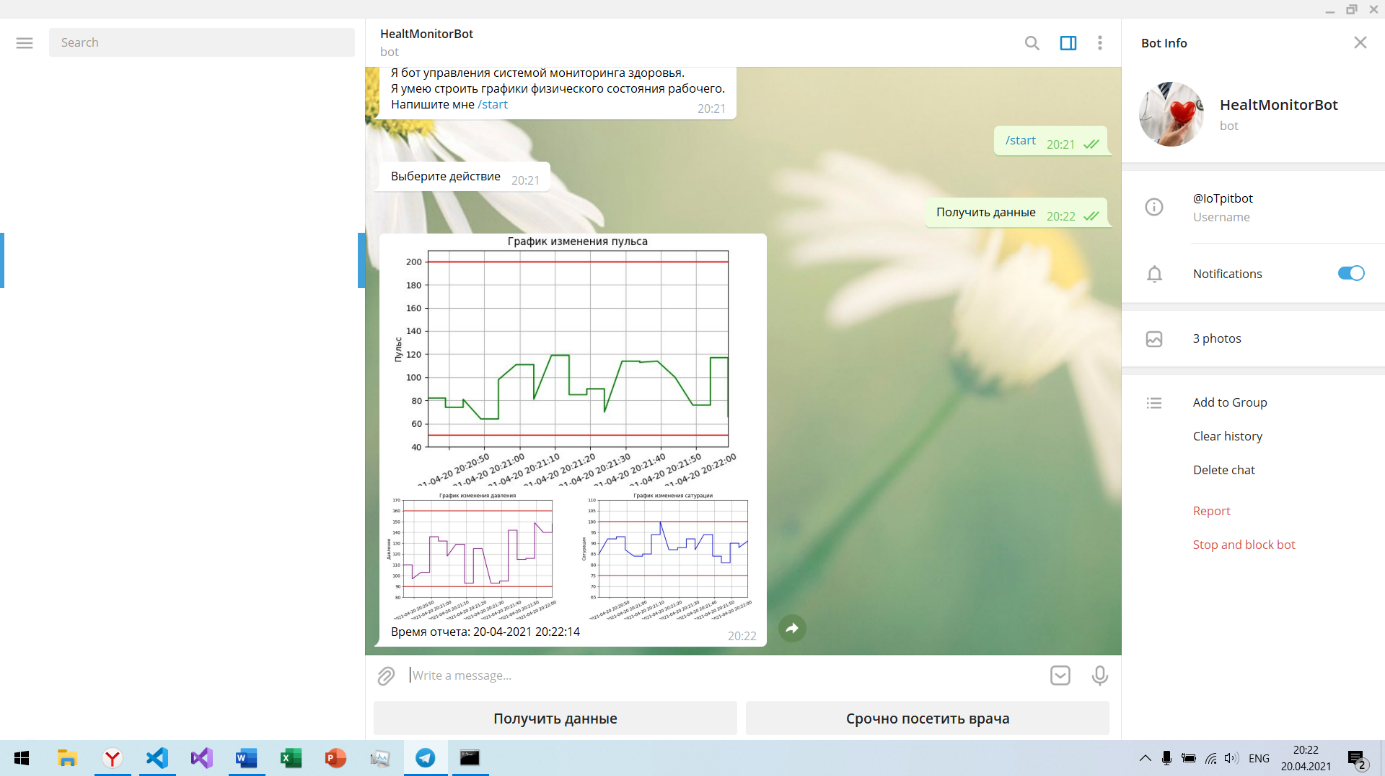


Рисунок 12

Для того, чтобы показать, что данные обновляются и графики строятся повторно, еще раз нажимается кнопка «Получить данные» (рисунок 13).

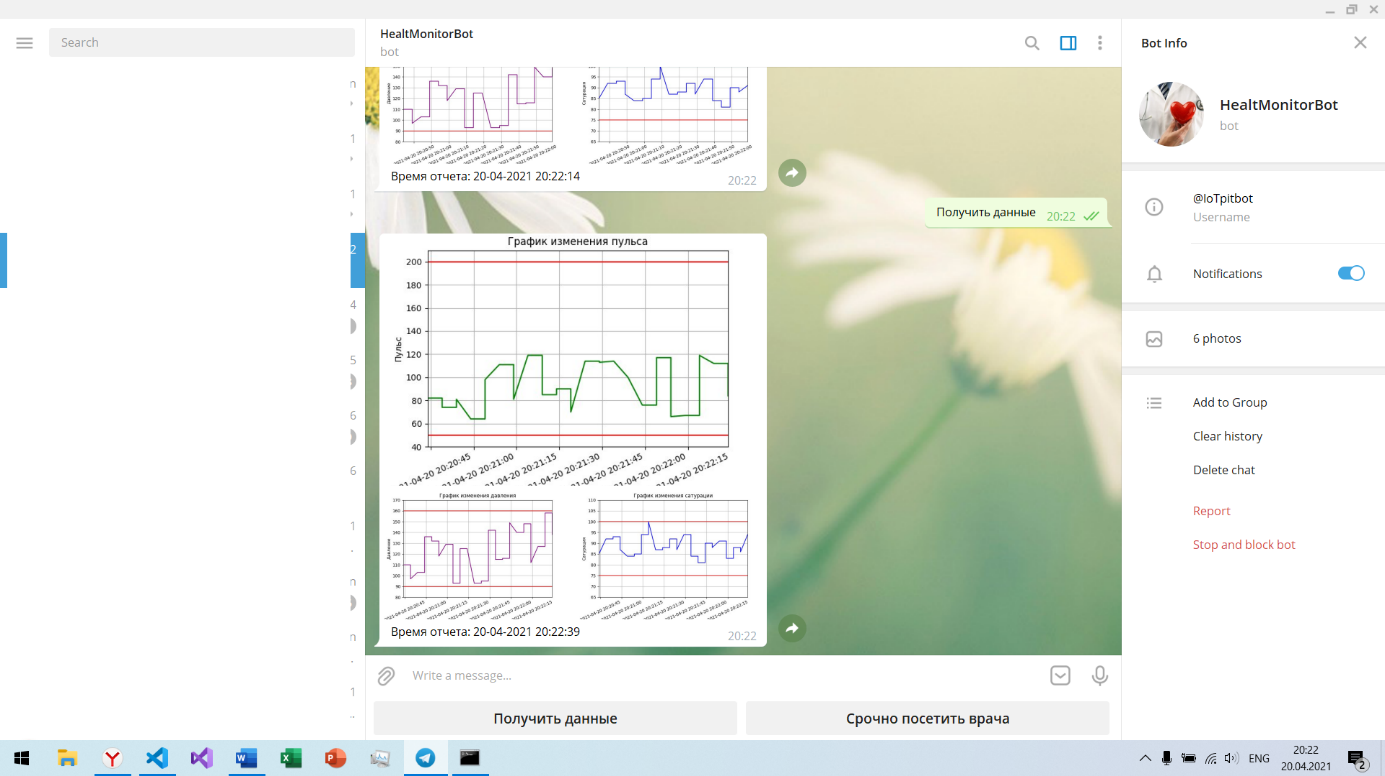


Рисунок 13

Видно, что графики были перестроены, и число точек на графиках больше, чем было на рисунке 12. Действительно, это происходит потому, что данные в БД постоянно обновляются, потому что работает программа backend.py. Далее нажимается кнопка «Срочно посетить врача», чтобы продемонстрировать, как чат-бот делает POST запросы на платформу (рисунок14, рисунок 15).

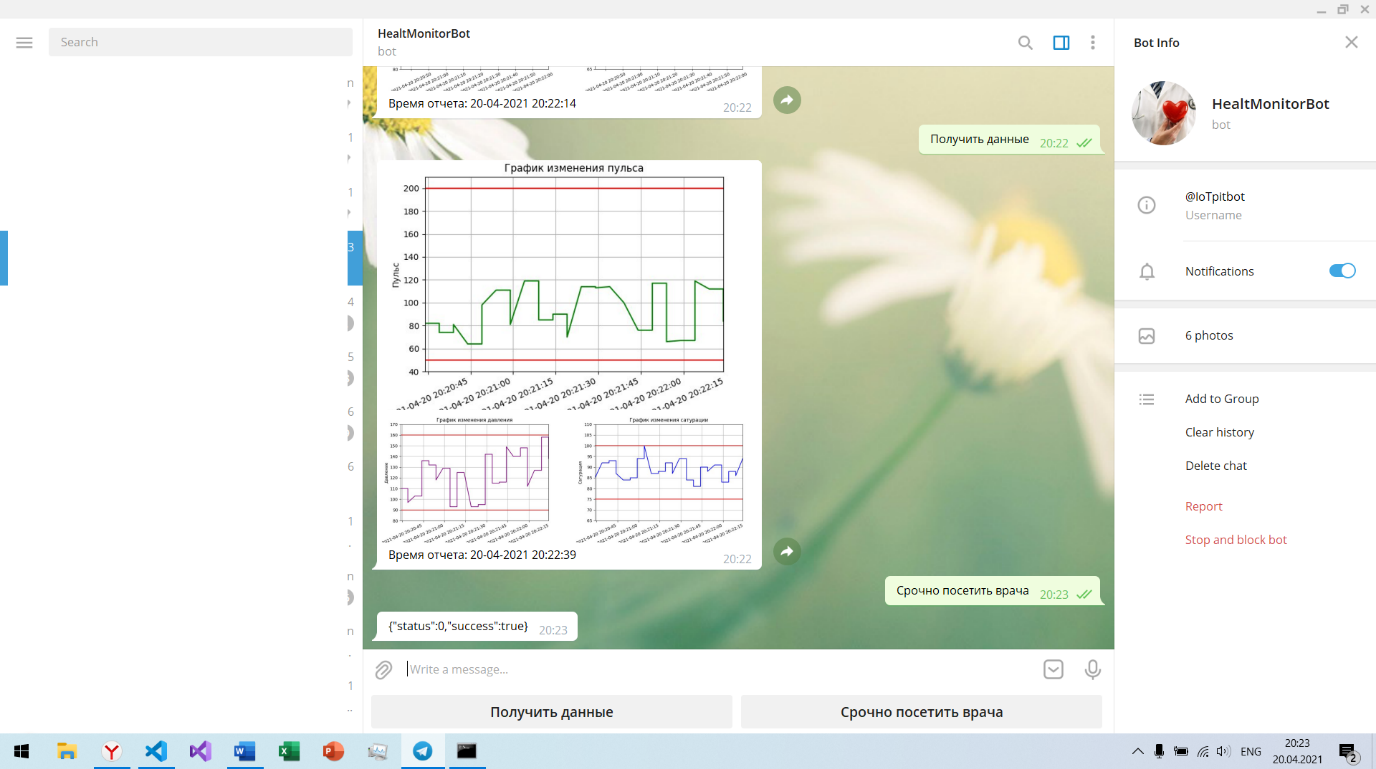


Рисунок 14

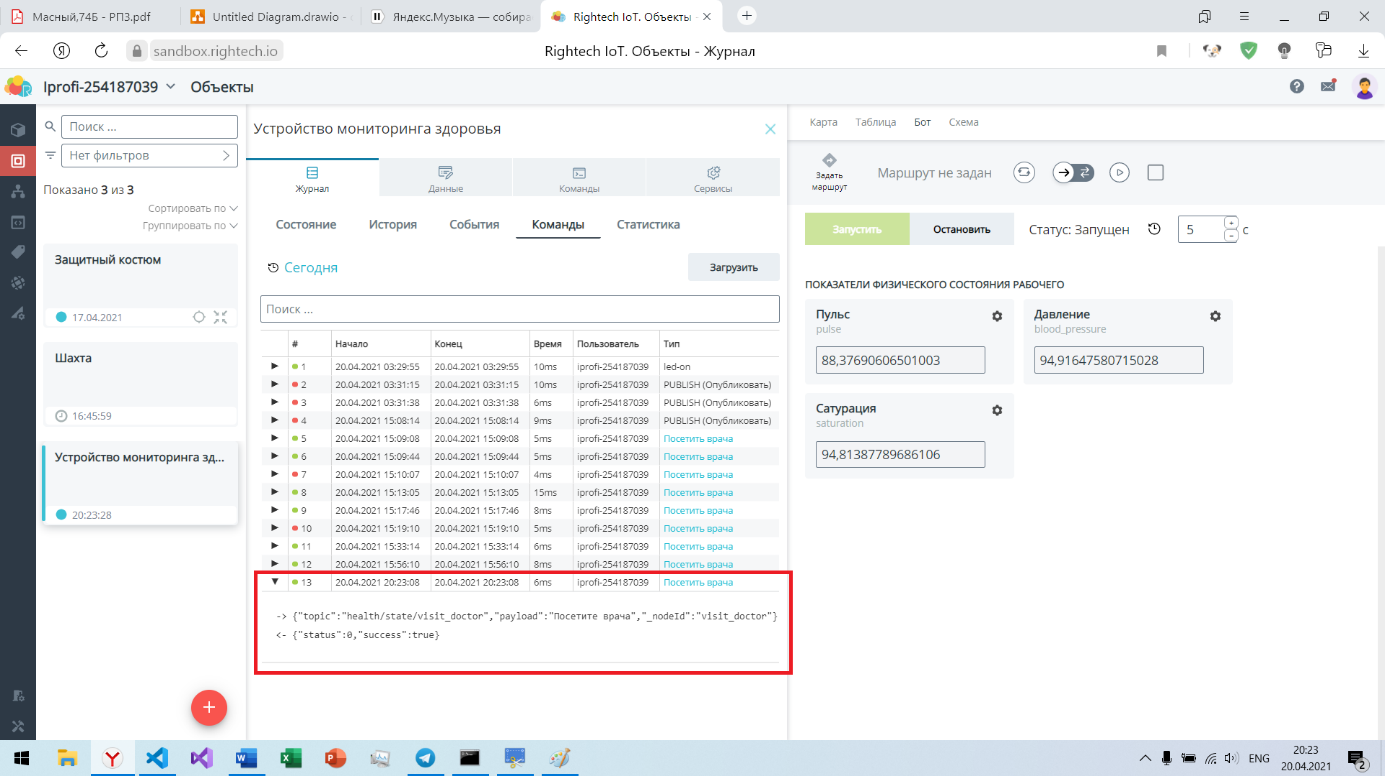


Рисунок 15

Далее проверяется реакция бота на некорректные сообщения (рисунок 16).

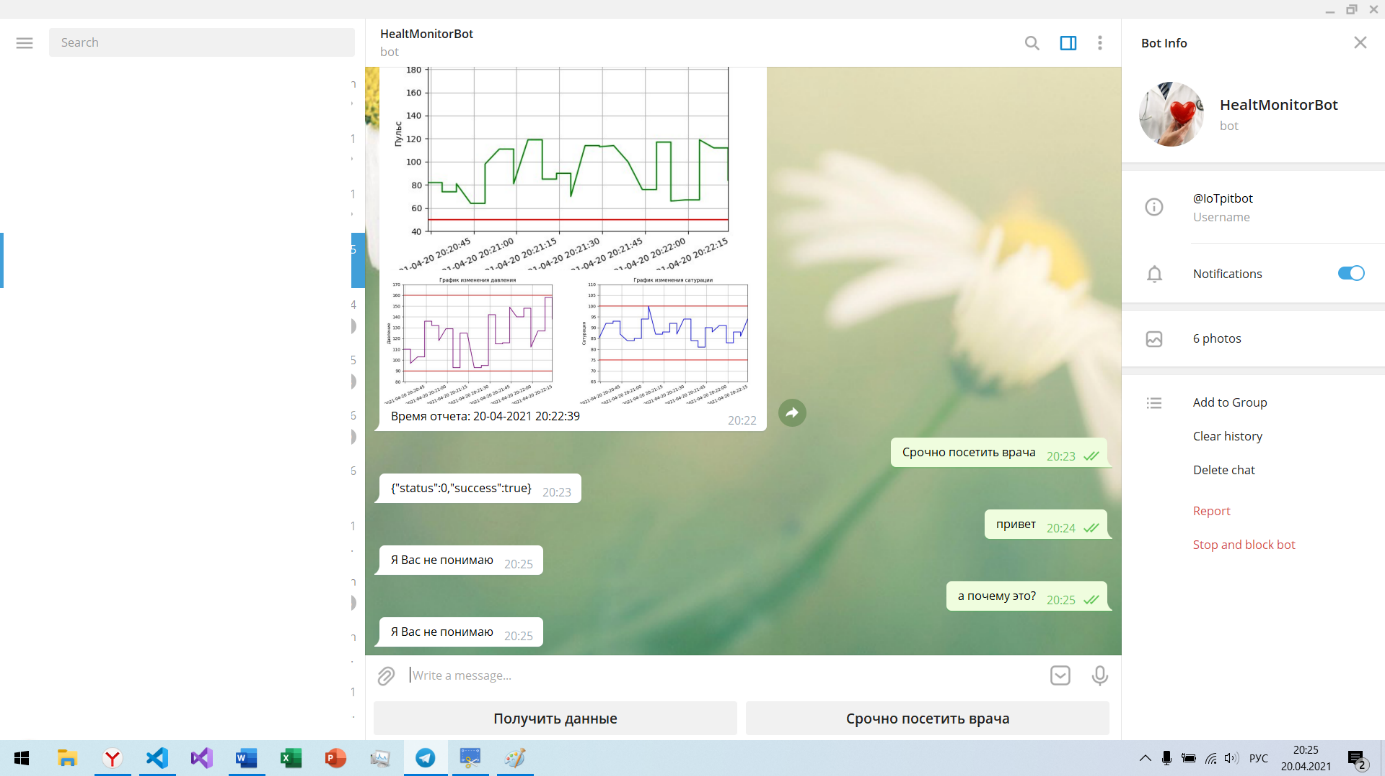


Рисунок 16

Кроме этого, видеозапись тестирования и подробного описания каждой программы можно найти по ссылке: <https://www.youtube.com/watch?v=RUYQomA0iLA> .

## **3.5 Перспективы развития проекта**

Дальнейшее развитие проекта может быть следующим: проектирование асинхронного исходного кода чат-бота для повышения производительности его работы, размещение чат-бота на сервере, добавление дополнительного функционала исходя из требований медицинских работников (данные требования можно получить при консультации специалистов), разработка модели машинного обучения (задача классификации), для предсказания критического состояния шахтера.

# **Заключение**

В ходе работы над проектом были получены следующие результаты:

1. В настоящее время на рынке отсутствуют системы непрерывного мониторинга физических показателей шахтеров, которые могли бы в режиме реального времени отправлять данные для обработки в облачной платформе.
2. Существующие разработки в области носимых устройств и беспроводной передачи данных позволяют создать устройство, способное измерять основные показатели физического состояния человека на протяжении всей смены и отправлять эти данные на облачную платформу.
3. Была разработана функциональная модель системы управления системой мониторинга физического состояния шахтеров.
4. Были разработаны алгоритмы программ сбора и визуализации данных.
5. Был разработан исходный код чат-бота на языке python 3.
6. Были протестированы функциональные возможности чат-бота и предложены дальнейшие сценарии развития системы.

Разработанный чат-бот позволяет работнику медицинской части горнодобывающего предприятия получать информацию о физическом состоянии определенного шахтера по запросу. В ответ на свой запрос врач получает три графика с показателями пульса, давления, сатурации за время от начала смены до времени запроса. На графике отмечена полоса допустимых значений показателей, что позволяет врачу быстрее интерпретировать полученные данные. В случае возникновения опасности для здоровья рабочего, врач отправляет информацию на устройство рабочему о необходимости посещения больницы.

# **Список литературы**

1. Basic date and time types [Электронный ресурс] // docs.python [сайт]. [2021] URL: <https://docs.python.org/3/library/datetime.html> (дата обращения 20.04.2021)
2. Documentation [Электронный ресурс] // sqlite [сайт]. [2021] URL: <https://www.sqlite.org/docs.html> (дата обращения 20.04.2021)
3. JSON encoder and decoder [Электронный ресурс] // docs.python [сайт]. [2021] URL: <https://docs.python.org/3/library/json.html> (дата обращения 20.04.2021)
4. Matplotlib: Visualization with Python [Электронный ресурс] // matplotlib [сайт]. [2021] URL: <https://matplotlib.org/> (дата обращения 20.04.2021)
5. NumPy Documentation [Электронный ресурс] // numpy [сайт]. [2021] URL: <https://numpy.org/doc/> (дата обращения 20.04.2021)
6. pyTelegramBotAPI [Электронный ресурс] // pypi [сайт]. [2021] URL: <https://pypi.org/project/pyTelegramBotAPI/> (дата обращения 20.04.2021)
7. Requests: HTTP for Humans [Электронный ресурс] // docs.python-requests [сайт]. [2020] URL: <https://docs.python-requests.org/en/master/> (дата обращения 20.04.2021)
8. Rest API [Электронный ресурс] // Rightech [сайт]. [2021] URL: <https://rightech.io/developers/http-api/> (дата обращения 20.04.2021)
9. Time access and conversions [Электронный ресурс] // docs.python [сайт]. [2021] URL: <https://docs.python.org/3/library/time.html> (дата обращения 20.04.2021)
10. Шахтеры: Были элитой - стали простыми смертниками [Электронный ресурс] // newsland [сайт]. [2010] URL: <https://newsland.com/user/4296711370/content/shakhtery-byli-elitoi-stali-prostymi-smertnikami/4041386#:~:text=Выяснилось%2C%20что%20средняя%20заработная%20плата,жизни%20горняка%20—%2053%20года> (дата обращения 20.04.2021)
11. «Южный Кузбасс» внедряет электронную систему медицинских осмотров работников [Электронный ресурс] // Press Release [сайт]. [2016] URL: <http://www.press-release.ru/branches/coal/c8e8a77531b6b/> (дата обращения 20.04.2021)