

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ Н.Э. БАУМАНА  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

# Серверное приложение для сбора статистики использования периферии АРМ

Студент: Дмитрий Васильевич Якуба, ИУ7-73Б

Руководитель курсового проекта: Николай Олегович Рогозин

МОСКВА, 2021 ГОД

# Цель и задачи

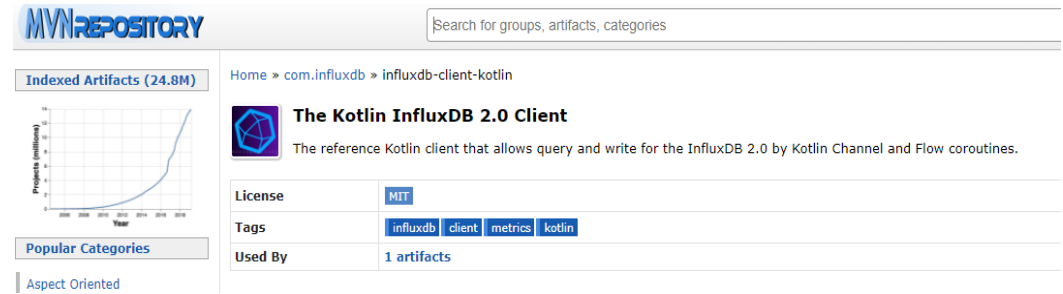
Спроектировать и реализовать серверное приложение для доступа к базе данных, предназначенной для хранения информации о действиях и характеристиках, необходимых для определения усталости пользователей автоматизированного рабочего места (АРМ).

Для достижения поставленной цели потребуется:

- 1) формализовать цель;
- 2) определить требуемую функциональность;
- 3) проанализировать существующие решения;
- 4) описать протокол взаимодействия клиента и сервера;
- 5) разработать приложение для решения поставленной цели.

# Существующие решения

The Kotlin InfluxDB 2.0 Client – это клиент, который предоставляет возможность производить запросы и запись в InfluxDV 2.0 с использованием языка программирования Kotlin.



Поддерживаемый на текущий момент функционал	Отсутствующий, но требующийся функционал
Запись и чтение с использованием InfluxQL	Проверка существования именного хранилища
Запись и чтение с используем Flux	Создание именного хранилища

# Существующие решения

InfluxDB API предоставляет способ взаимодействия с базой данных с использованием протокола HTTP.



Предоставляемый интерфейс функционал полон и непосредственно используется в реализации Web-клиента СУБД.

Недостаток: формирование множественных HTTP-запросов и их обработка.

# Протокол НТТР

Реализует клиент-серверную технологию, которая предполагает наличие множества клиентов, а также множества серверов.

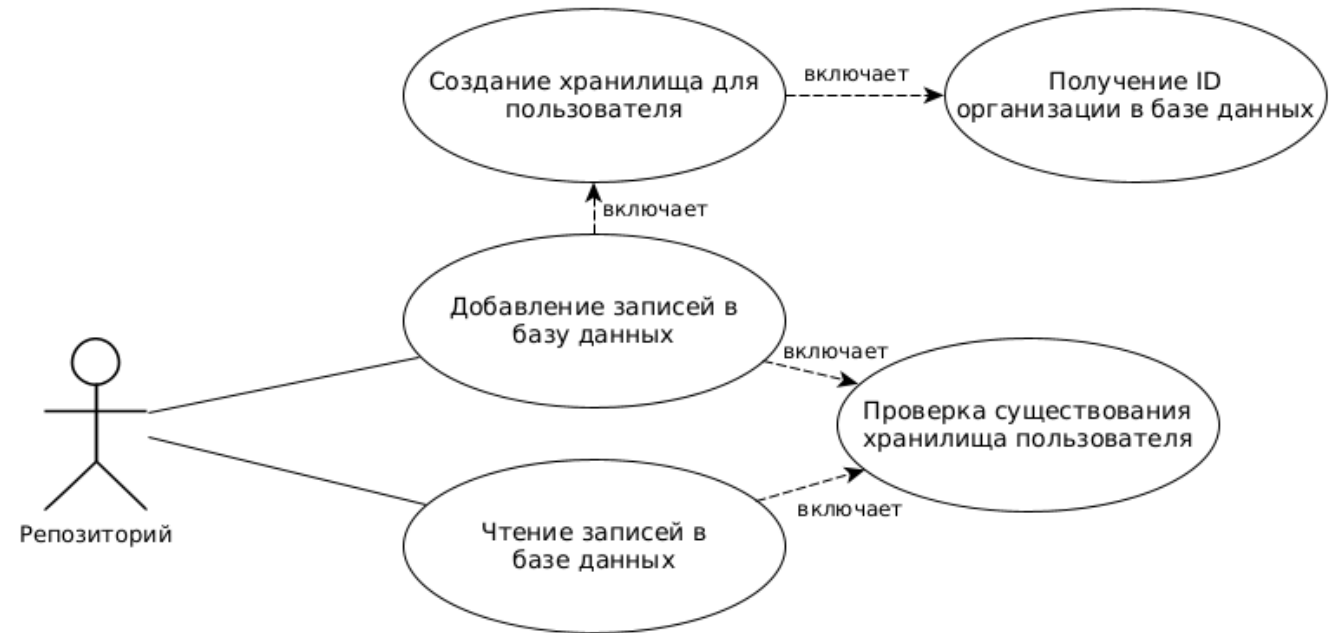
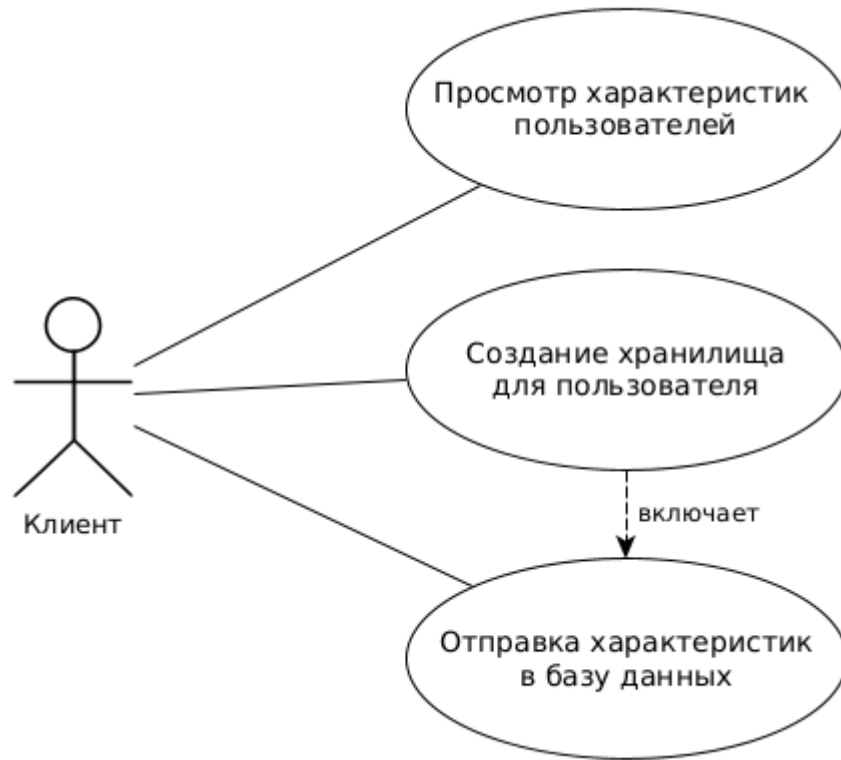
Главный объект обработки — это ресурс, который записан в форме URI. В качестве ресурса выступают файлы.

Каждое сообщение состоит из стартовой строки, заголовков и тела.

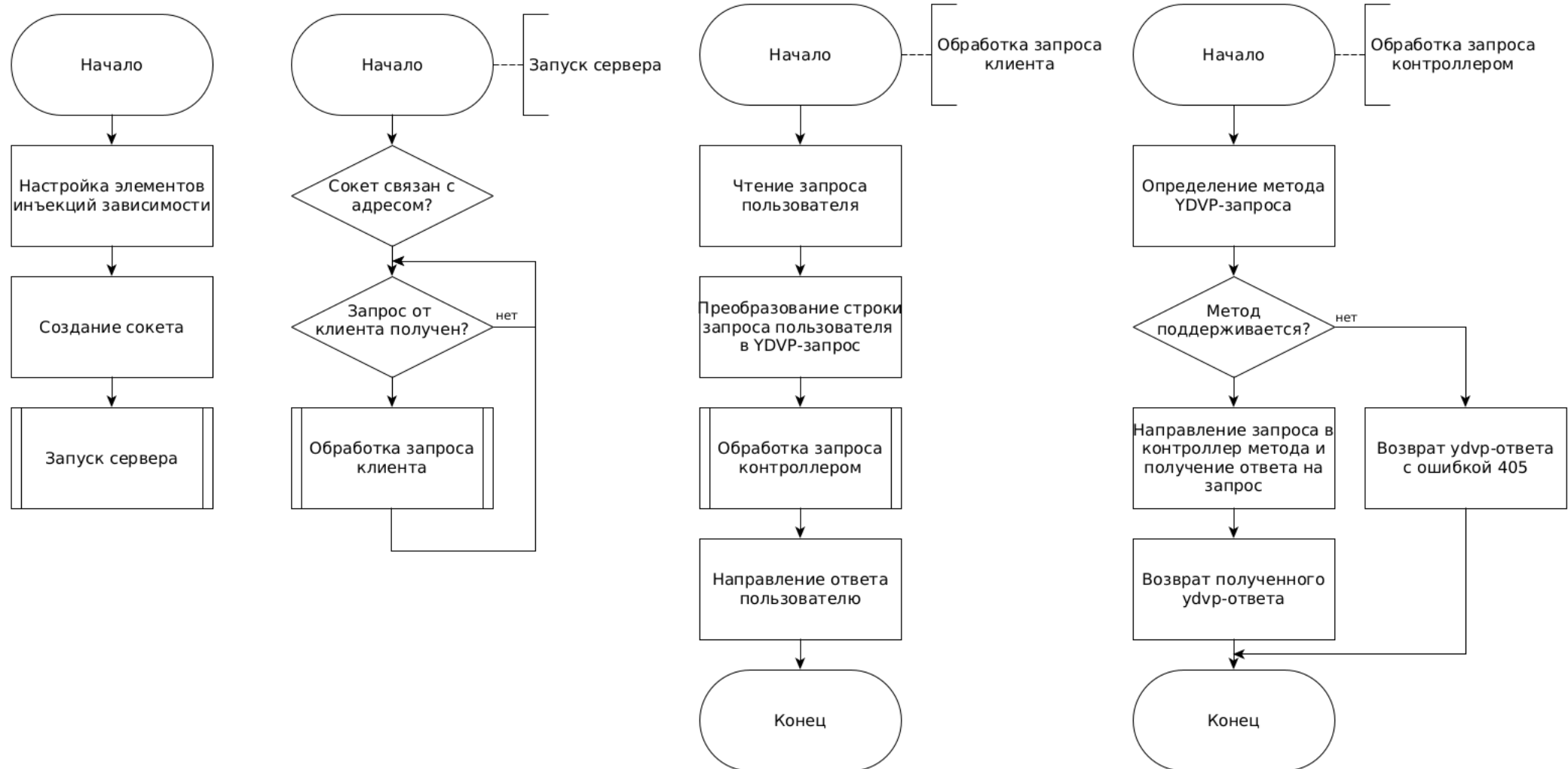
Стартовая строка включает в себя метод, путь и версию используемого протокола.

В ответе сервера на запрос присутствует код состояния, по которому клиент узнает о результатах выполнения запроса.

# Участники и компоненты системы



# Схема работы сервера



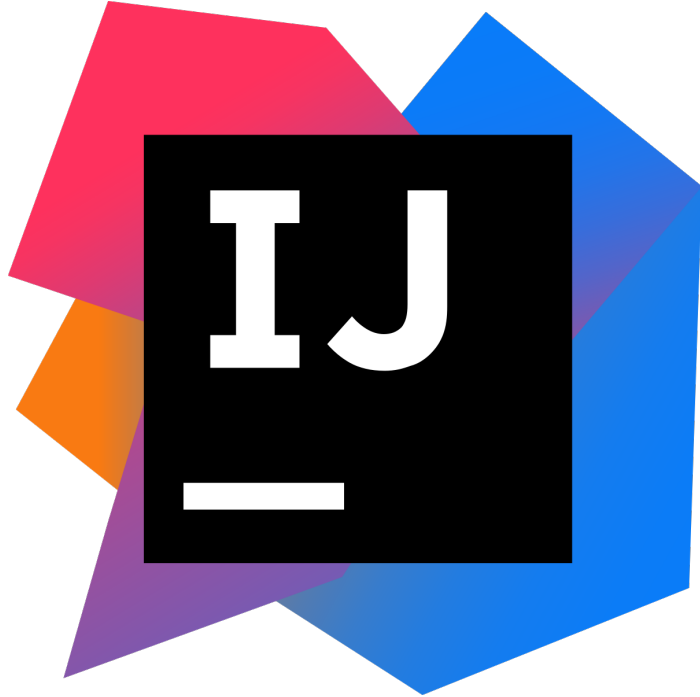
# Протокол YDVP

Протокол YDVP – это протокол для взаимодействия пользователя и сервера в реализуемой системе. Данный протокол основан на существующем стандарте HTTP.

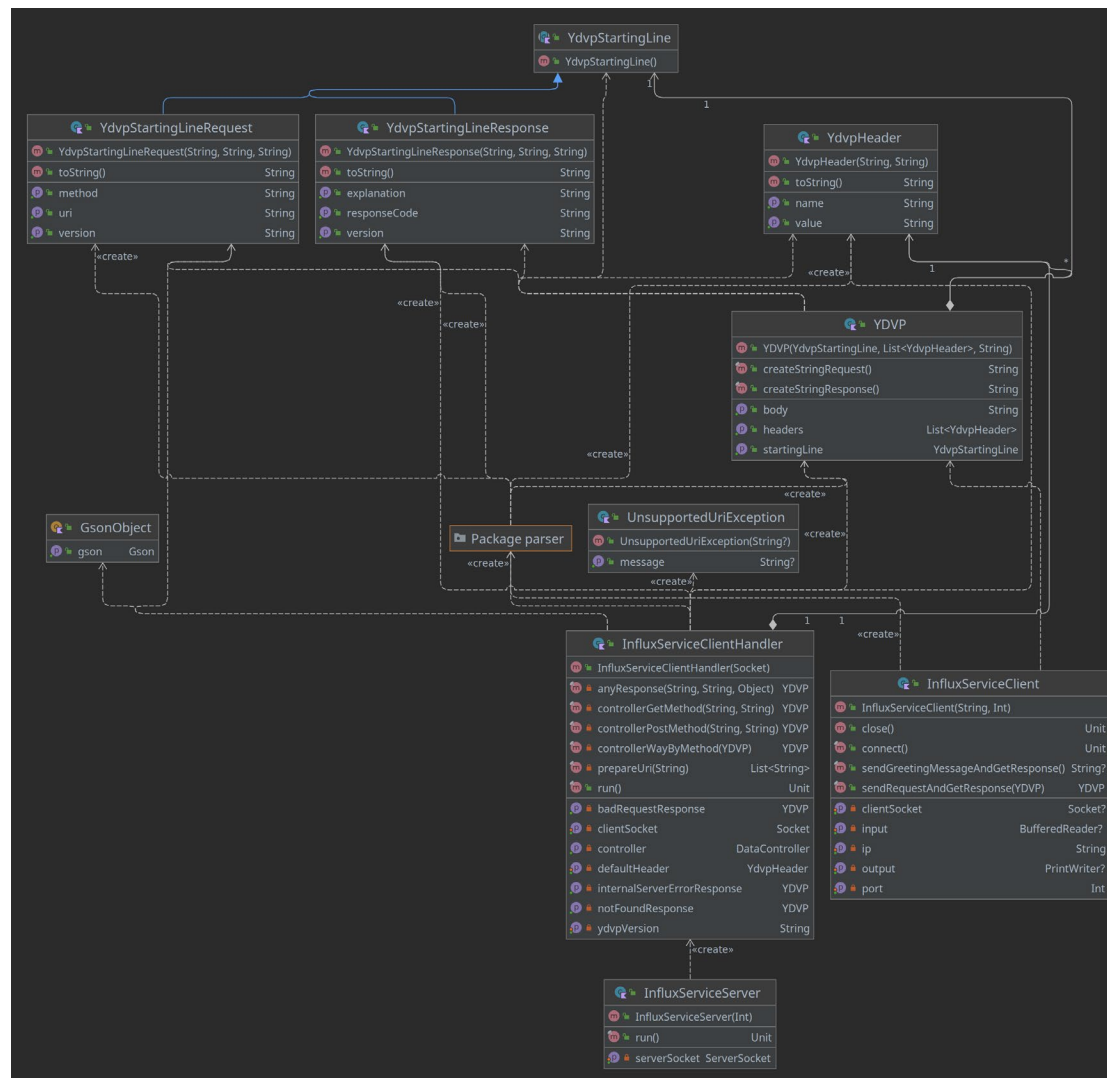
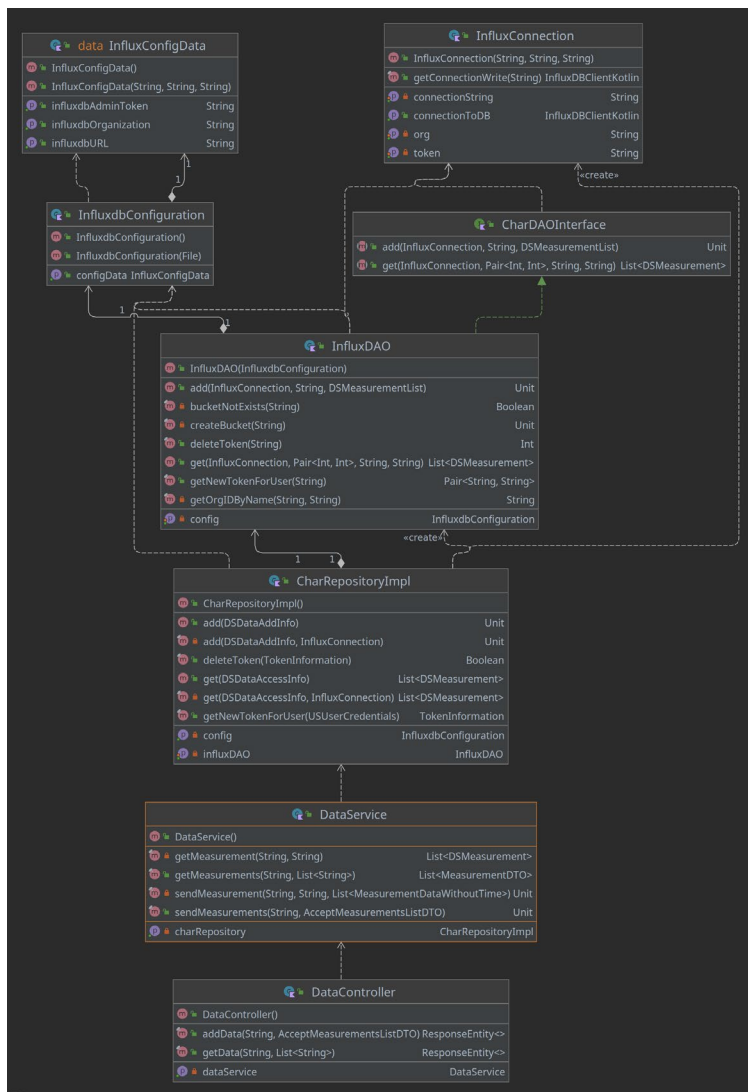
Имя компонента(-ов) YDVP	Структурные составляющие и описание
Запрос и ответ	<ul style="list-style-type: none"><li>• стартовая строка (определяет тип сообщения),</li><li>• заголовок (характеризует тело сообщения параметры передачи и др.),</li><li>• тело (данные сообщения).</li></ul>
Стартовая строка запроса	<ul style="list-style-type: none"><li>• метод (тип запроса),</li><li>• путь к запрашиваемому ресурсу,</li><li>• версия используемого протокола.</li></ul>
Стартовая строка ответа	<ul style="list-style-type: none"><li>• версия используемого протокола;</li><li>• код состояния (определяет дальнейшее содержимое сообщения и поведения клиента);</li><li>• пояснение к коду состояния.</li></ul>



# Технические средства реализации



# Диаграммы классов приложения



# Пример работы

В примере рассматривается вариант отправки клиентом на сервер некоторых характеристик пользователя.

## Результат выполнения запроса на стороне клиента

```
/home/trvehazzk3r/Downloads/jdk-11.0.12/bin/java ...
Client connected from /127.0.0.1:55082 to localhost/127.0.0.1:6666
Formed request in client:
POST /data/TestUser YDVP/0.1
Host: 127.0.0.1

{"measurements":[{"measurement":"pulse","values":[{"value":"30"}, {"value":"40"}]}, {"measurement":"botArterialPressu
Response in client:
YDVP/0.1 200 OK
Server: 127.0.0.1

{"code":200,"message":"Measurements were carefully sent","description":"We know all about you now \u003e:c"}

Process finished with exit code 0
```

## Результат выполнения запроса на стороне сервера

```
/home/trvehazzk3r/Downloads/jdk-11.0.12/bin/java ...
Server started on port 6666
Accepted client on /127.0.0.1:6666 from /127.0.0.1:55082
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by retrofit2.Platform (file:/home/trvehazzk3r/.gradle/caches/modules-2/files-2.1/
WARNING: Please consider reporting this to the maintainers of retrofit2.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Returned to client:
YDVP/0.1 200 OK
Server: 127.0.0.1

{"code":200,"message":"Measurements were carefully sent","description":"We know all about you now \u003e:c"}
```

# Заключение

- 1) цель была формализована,
- 2) была определена требуемая функциональность,
- 3) были проанализированы существующие решения,
- 4) был описан протокол взаимодействия клиента и сервера,
- 5) было разработано серверное приложение для доступа к базе данных, предназначенной для хранения информации о действиях и характеристиках, необходимых для определения усталости пользователей АРМ.

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ Н.Э. БАУМАНА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ)  
МОСКВА, 2021 ГОД