

# Мобильное приложение для поддержки носимых сенсоров

Подготовил: Жуков Д.Н. 401-ПК

## Область применения

Приложение будет использоваться с целью отображения графиков, на основе полученных данных из облачного сервиса.

# Актуальность

Приложение основывается на данных носимых сенсоров

# Цель работы

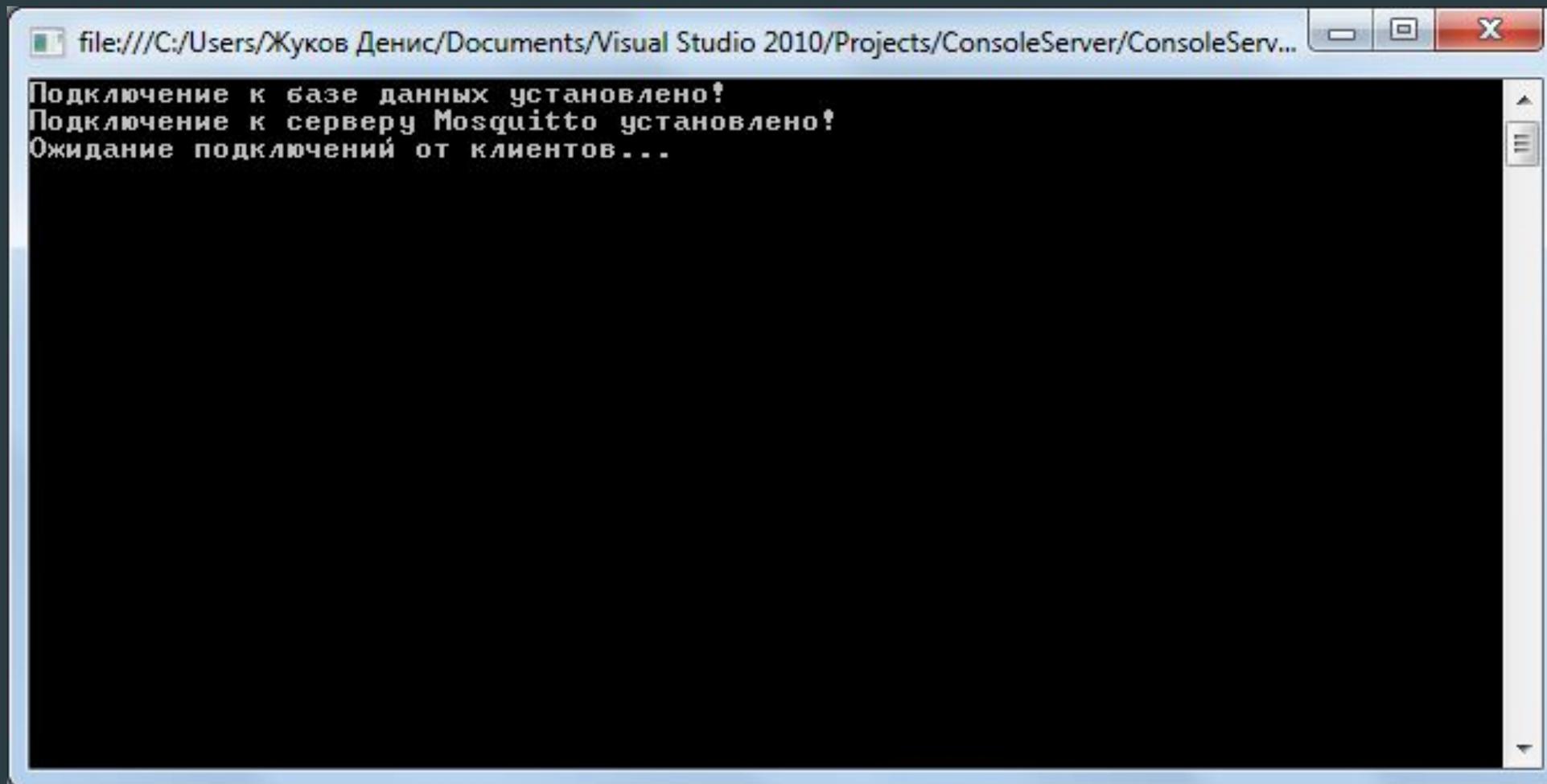
Разработать мобильное приложение для ОС Android, которое будет делать запрос в базу данных через определённый промежуток времени и строить графики.

# Постановка задачи

- 1) Разработать тестовый сервер для записи данных в БД
- 2) Создать базу данных
- 3) Создать эмулятор датчиков для тестирования пропускной способности MQTT протокола
- 4) Создать приложение для ОС Android

# Этап 1. Разработать тестовый сервер для записи данных в БД

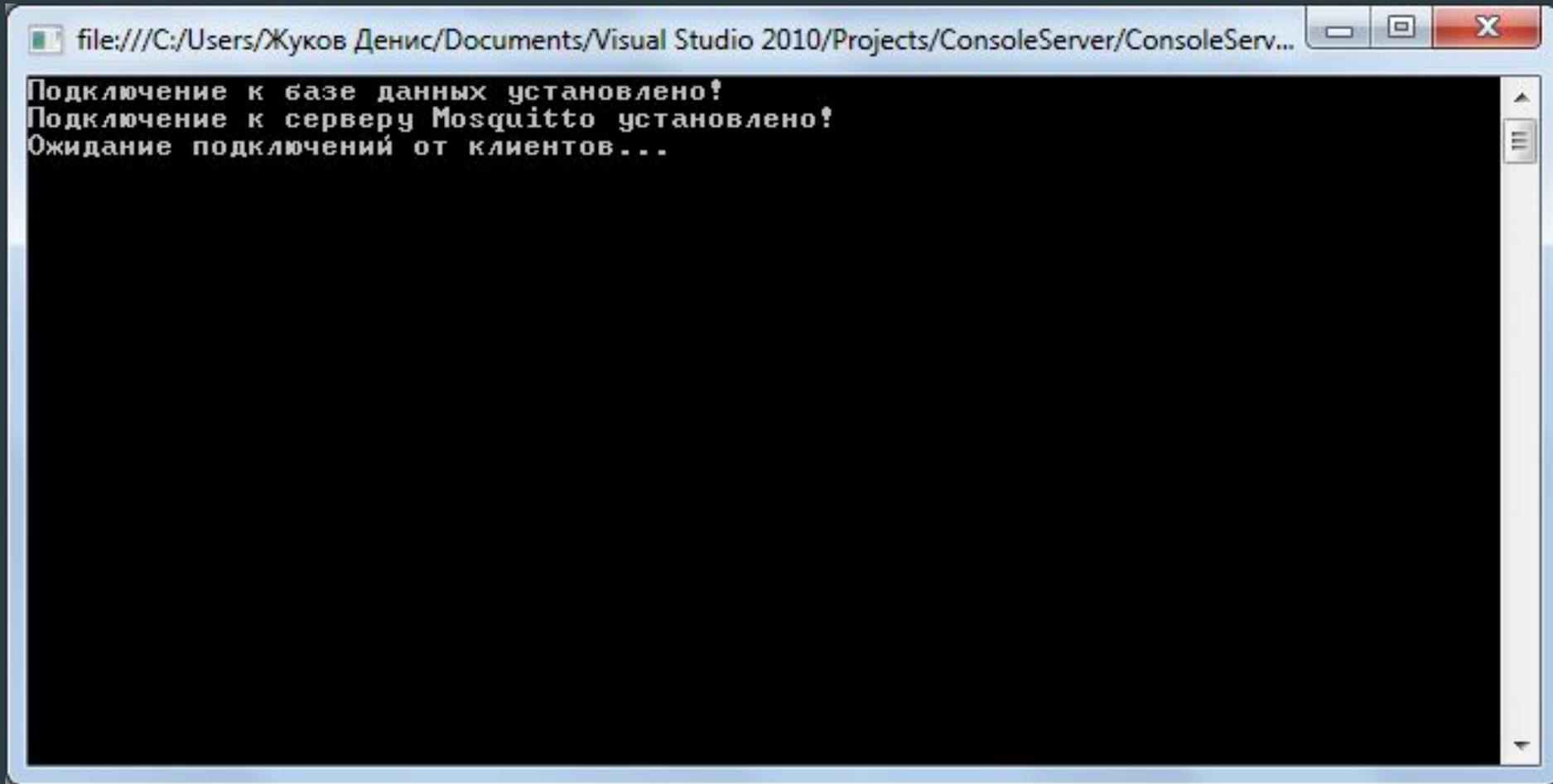
Для создания тестового сервера, я выбрал среду разработки Visual Studio 2010. Язык программирования C#. Для более быстрой работы, сервер будет консольным



The screenshot shows a console application window with the following text:

```
file:///C:/Users/Жуков Денис/Documents/Visual Studio 2010/Projects/ConsoleServer/ConsoleServ...  
Подключение к базе данных установлено!  
Подключение к серверу Mosquitto установлено!  
Ожидание подключений от клиентов...
```

# Работа сервера



The image shows a screenshot of a console window from a Windows operating system. The window title bar contains the file path: `file:///C:/Users/Жуков Денис/Documents/Visual Studio 2010/Projects/ConsoleServer/ConsoleServ...`. The console output consists of three lines of text in Russian, indicating successful database and server connections, followed by a message about waiting for client connections.

```
Подключение к базе данных установлено!  
Подключение к серверу Mosquitto установлено!  
Ожидание подключений от клиентов...
```

## Этап 2. Создать базу данных

В качестве базы данных я выбрал MySQL, для администрирования БД использовал PhpMyAdmin. Для поднятия локального сервера, использовалась платформа Open Server.

# Описание таблиц

База данных будет иметь 2 таблицы, такие как Users и Sensor.

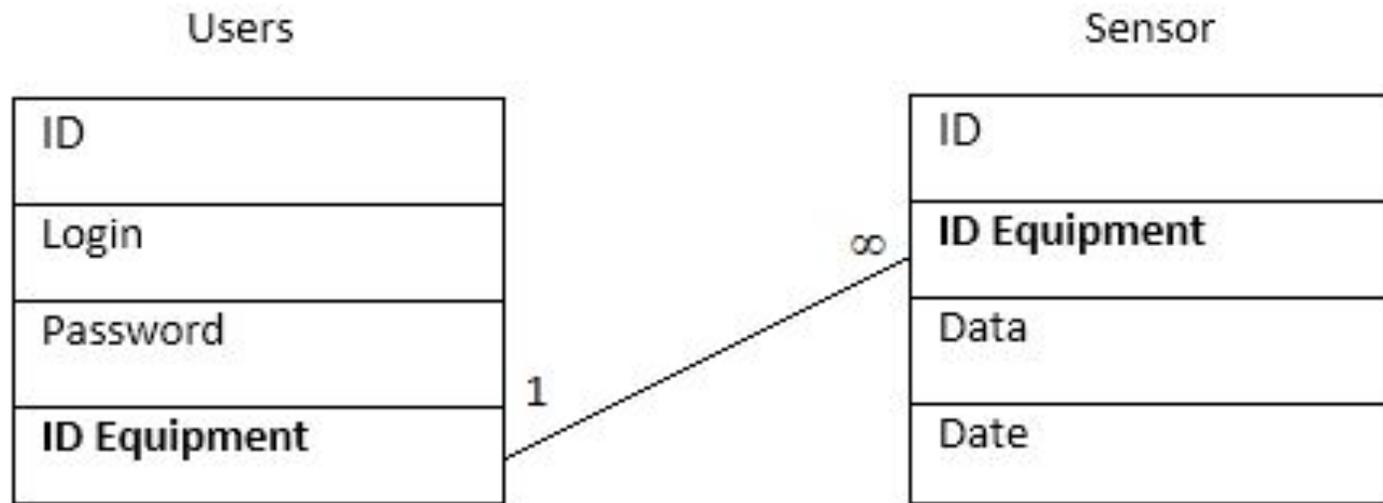
Таблица Sensor содержит 4 поля.

1. Id - уникальный идентификатор записи в БД
2. IDequipment - уникальный идентификатор датчика (ключевое поле)
3. Data - данные с датчика
4. Date - дата записи строки в БД

Таблица Users содержит 4 поля.

1. Id - уникальный идентификатор пользователя
2. Login - логин пользователя
3. Password - пароль пользователя
4. IDequipment - уникальный идентификатор датчика (ключевое поле)

# Схема базы данных



# ER-модель



# Физическая модель данных БД

**Users**

Наименование столбца	Тип данных	Ограничения
ID	Числовой	Значение должно быть уникальным
Login	Текстовый	Значение должно быть уникальным
Password	Текстовый	Значение не должно быть уникальным
<b>ID Equipment</b>	Числовой	Значение должно быть уникальным

1

**Sensor**

Наименование столбца	Тип данных	Ограничения
ID	Числовой	Значение должно быть уникальным
<b>ID Equipment</b>	Числовой	Значение должно быть уникальным
Data	Числовой	Значение не должно быть уникальным
Date	Дата	Значение не должно быть уникальным

M



# Нормализация таблиц

**Users**

ID	L	P	IDE
----	---	---	-----

1

**Sensor**

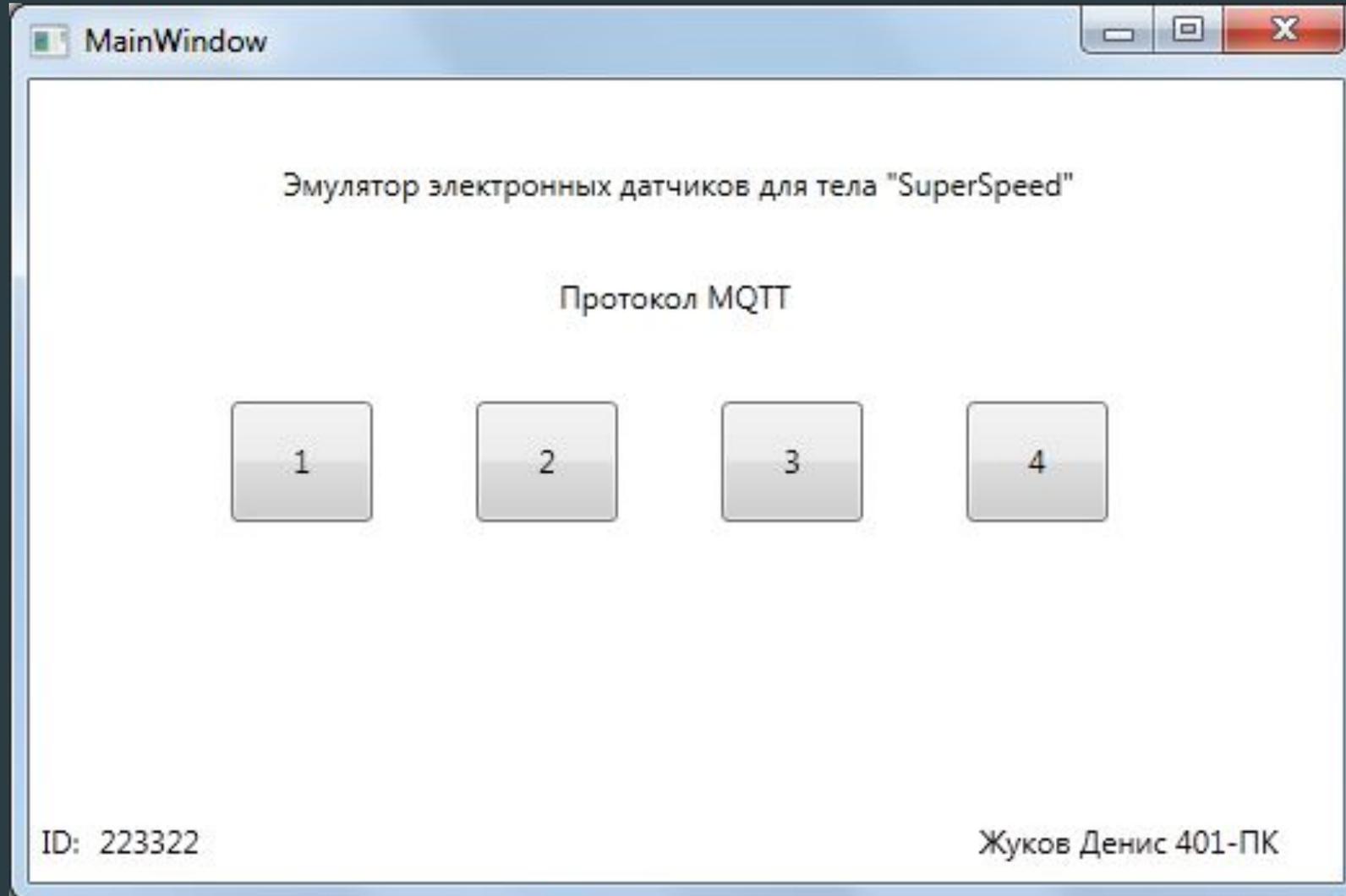
ID	IDE	D	D
----	-----	---	---

M



## Этап 3. Создать эмулятор датчиков

Для создания эмулятора датчиков, так же была использована среда разработки Visual Studio и язык программирования C#



## Этап 4. Создать приложение для ОС Android

Для разработки приложения был выбран игровой движок Unity3D. Его основные плюсы:

- ▶ Бесплатен для коммерческих проектов
- ▶ Прост в освоении
- ▶ Облегчена возможность портирования под другие платформы

## Пример работы приложения, при запуске

Всё хорошо, мы подключены!

Denis

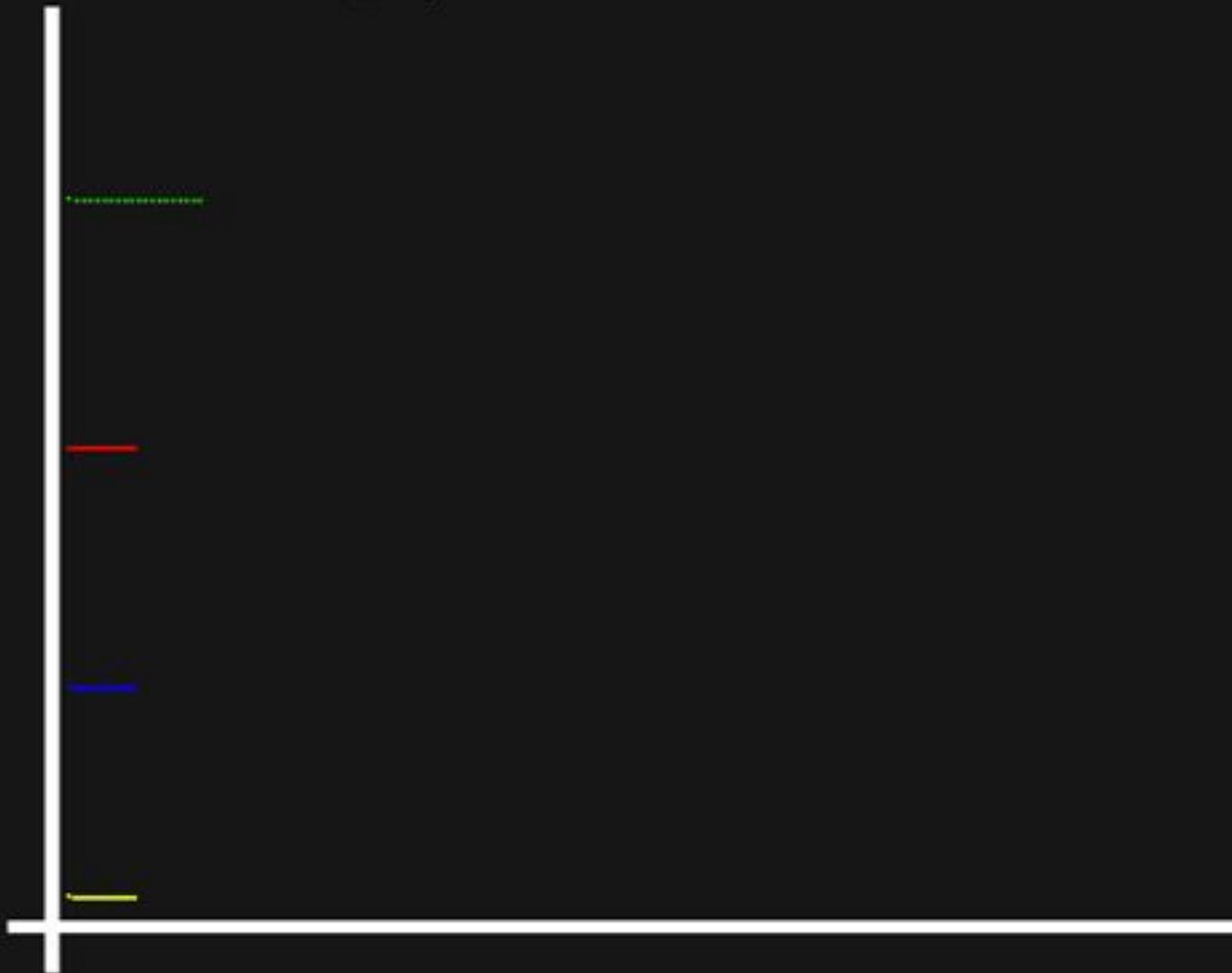
12345

223322

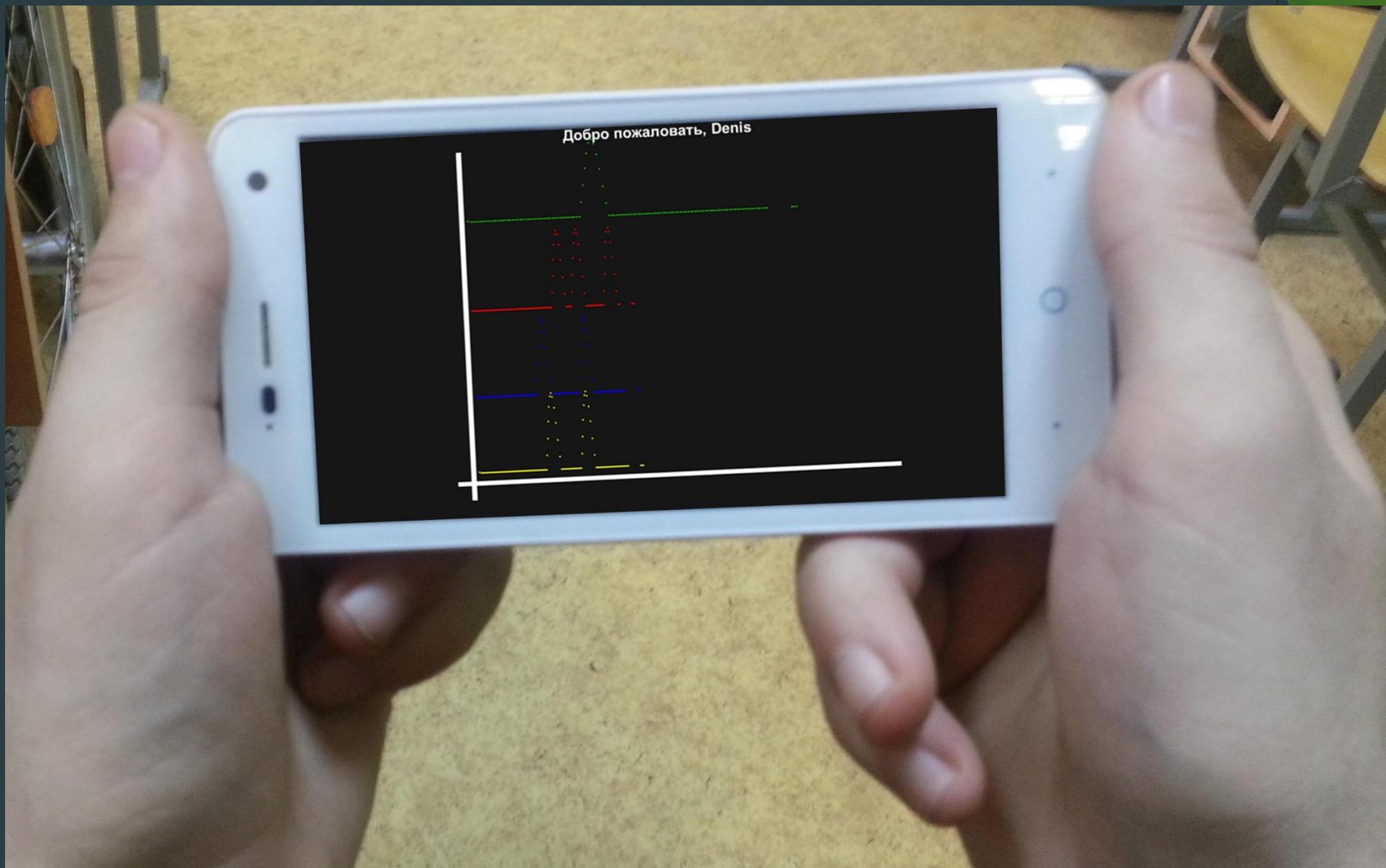
Send

# Пример работы приложения, после авторизации

Добро пожаловать, Denis



Пример работы приложения, при получении данных



Спасибо за внимание!