**Курсов проект**

**Дисциплина: Проектиране и интегриране на софтуерни системи**

**Фаза 3: Реализация на системата**

**Версия 1.1**

|  |  |  |
| --- | --- | --- |
| **Фак. №** | **Име на студент** | **Секция от документа** |
| 61908 | **Бранимир Кирилов** | **1,2,3,4,5,6** |

**Август, 2019**

**Съдържание**

[1 Въведение 3](#_gjdgxs)

[1.1 Цел 3](#_30j0zll)

[1.2 Резюме 3](#_1fob9te)

[2 Използвани технологии 4](#_2et92p0)

[3 Реализация на](#_tyjcwt) хардуера 5

[4 Реализация на бизнес логиката 6](#_3dy6vkm)

[5 Реализация на потребителския интерфейс 7](#_4d34og8)

[6 Внедряване на системата 8](#_2s8eyo1)

# Въведение

## Цел

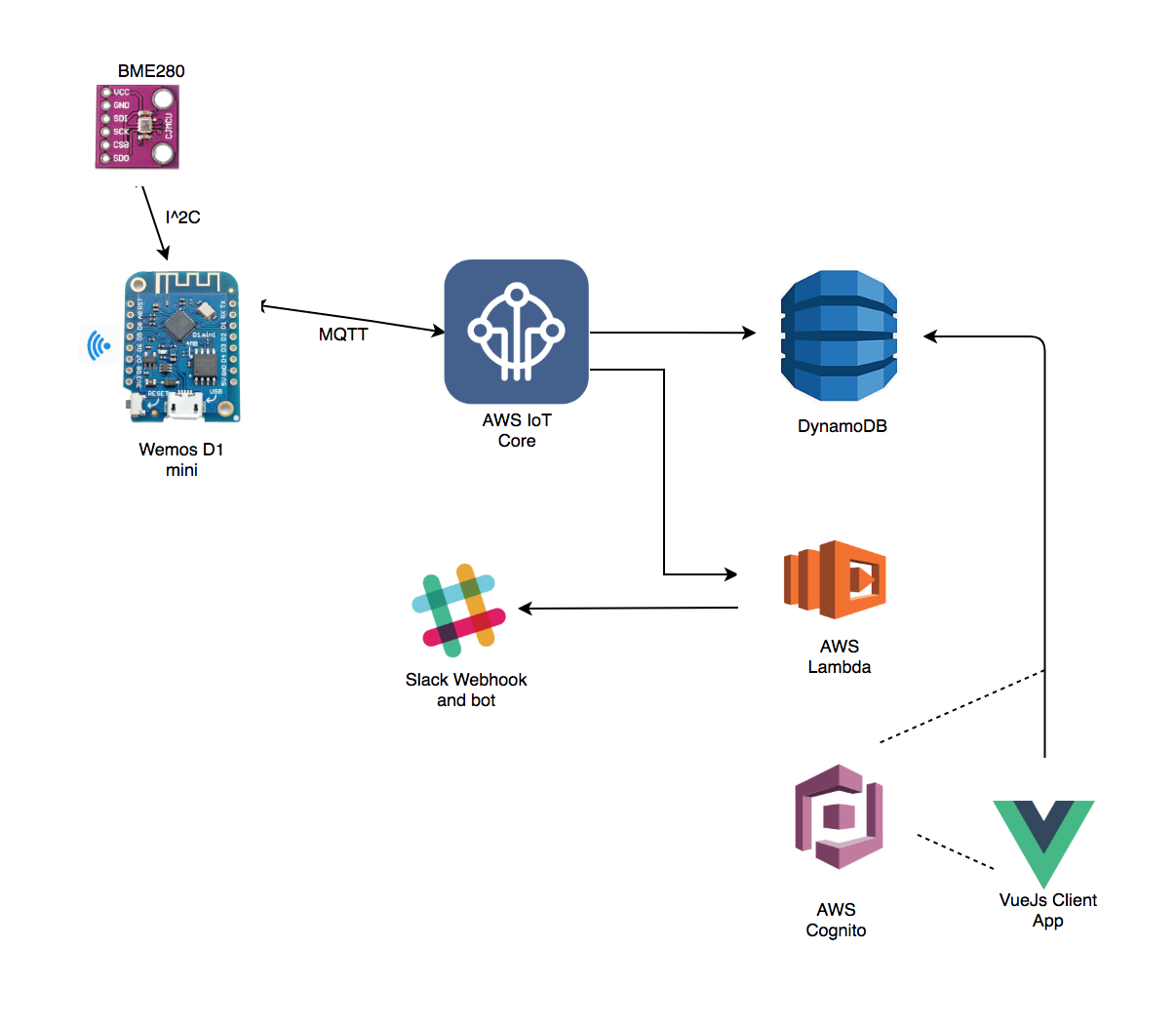
Настоящият документ има за цел да покаже използваните технологии, реализацията на бизнес логиката, архитектурата на приложението, както и потребителския интерфейс на системата за размяна на подаръци **Io.tty**.

## Резюме

Документът предоставя описание на системното проектиране и реализацията на системата за отдалечен мониторинг на помещения. Втората част на документа предоставя използваните технологии, разделени на няколко части – за всеки от модулите на системата. Третата част описва реализацията на хардуерната част от системата. В четвъртата част е представена реализацията на бизнес логиката. Петата част от документа показва потребителския интерфейс. Шестата част описва стъпките, необходими за внедряването на системата.

# Използвани технологии

Проектът е изграден от няколко отделни системи, за всяка от които са използвани различни технологии:



* **Hardware**

Хардуерното устройство е изградено от платка **Wemos** **D1** **mini**. Към нея са прикачени два сензора - **BME 280**, който измерва температура, влажност и атмосферно налягане, както и **HC** **SR04**, който измерва разстояние. За разработка и качване на кода на платката е използвано **Arduino** **IDE**.

* **Сървър** Бизнес логиката на системата се извършва от няколко **AWS** сървиса. AWS **IoT** **Core** е използван за да се осъществи връзката с устройството посредством **MQTT** протоколи да се пренасочат данните към други два AWS сървиса - **DynamoDB** и **AWS** **Lambda**. **DynamoDB** представлява база от данни където се записват измерванията. **AWS** **Lambda** се използва за изпращането на данните в **Slack** канал като нотификации.

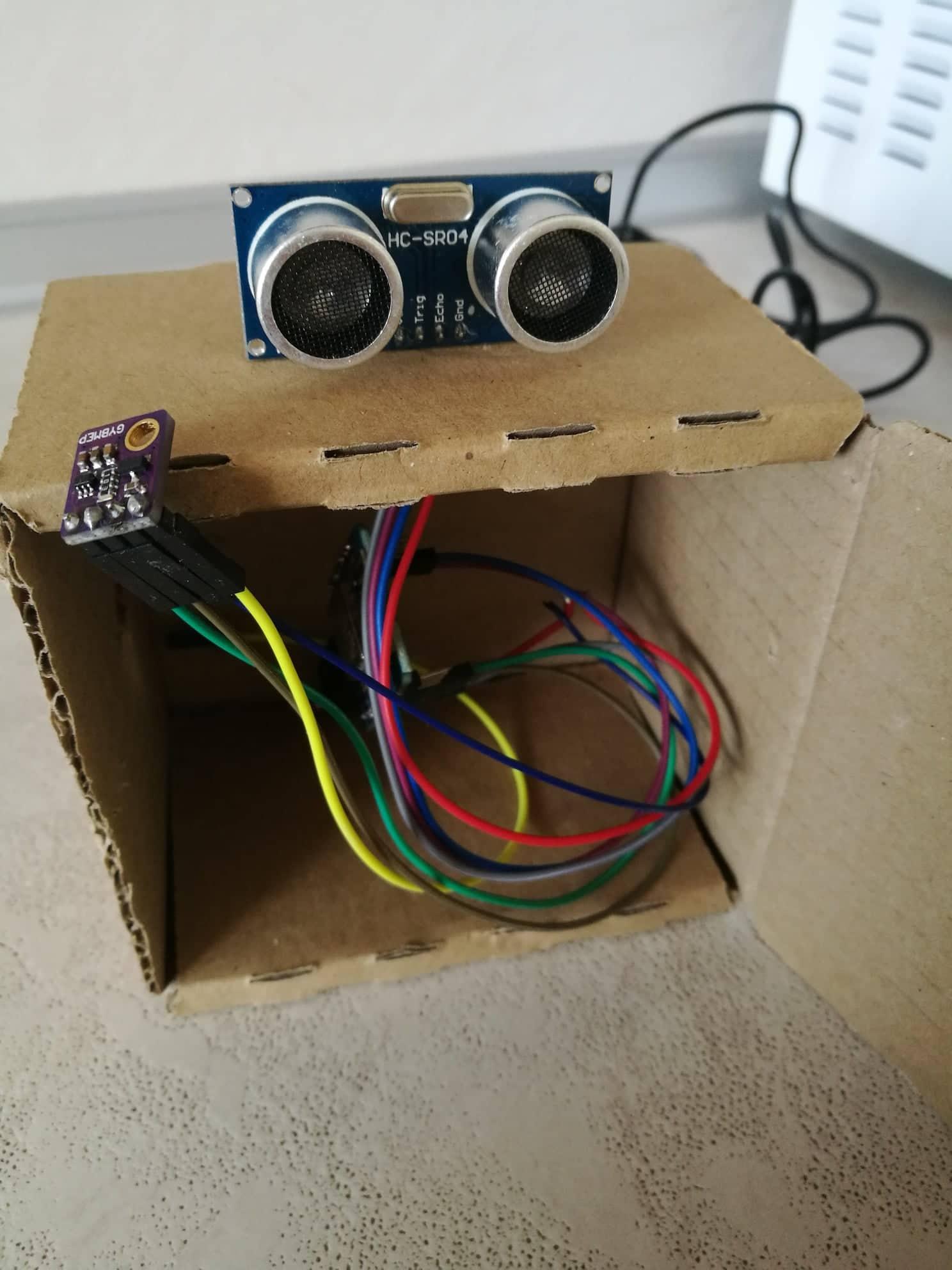
* **Клиент**

Клиентската част е реализирана като single page application (**SPA**), използвайки библиотеката **Vue.js** + **aws-amplify**. Клиента комуникира директно с **DynamoDB** като аутентикацията на отделните потребители е реализиране с **AWS Cognito**.

# Реализация на хардуера

*Свързване на* ***BME******280*** *(температура, влажност и атмосферно налягане):*

|  |  |
| --- | --- |
| ***Wemos*** | ***BME280*** |
| *GND* | *GND* |
| *3.3V* | *VCC* |
| *D3 (gpio0)* | *SDA* |
| *D4 (gpio2)* | *SCL* |

*Свързване на* ***HC-SR04*** *(разстояние):*

|  |  |
| --- | --- |
| ***Wemos*** | ***HC-SR04*** |
| *GND* | *GND* |
| *5V* | *VCC* |
| *D6 (gpio12)* | *Echo via voltage divider* |
| *D7 (gpio13)* | *Trigger* |

# Реализация на бизнес логиката

*Системата е разделена на 3 основни подсистеми – сървър, клиентско приложение и хардуер.*

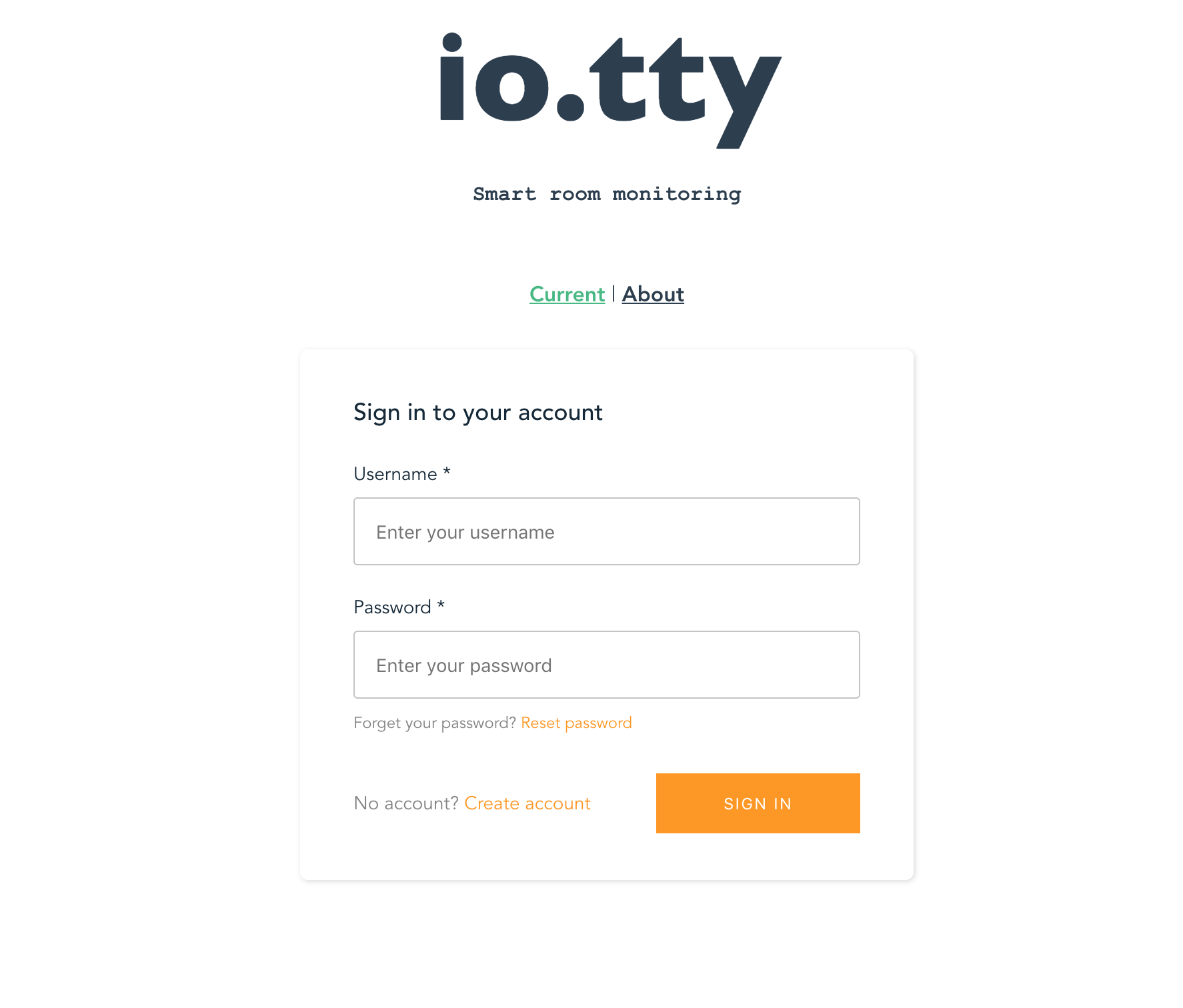
*Сървърното приложение е на база на microservice архитектура, като всички сървиси са базирани в Amazon Web Services (****AWS****). Всеки microservice отговаря за точно определена задача и може лесно да бъде подменен с друг. Също така заради използването на AWS системата може безпроблемно* ***да******скалира****.*

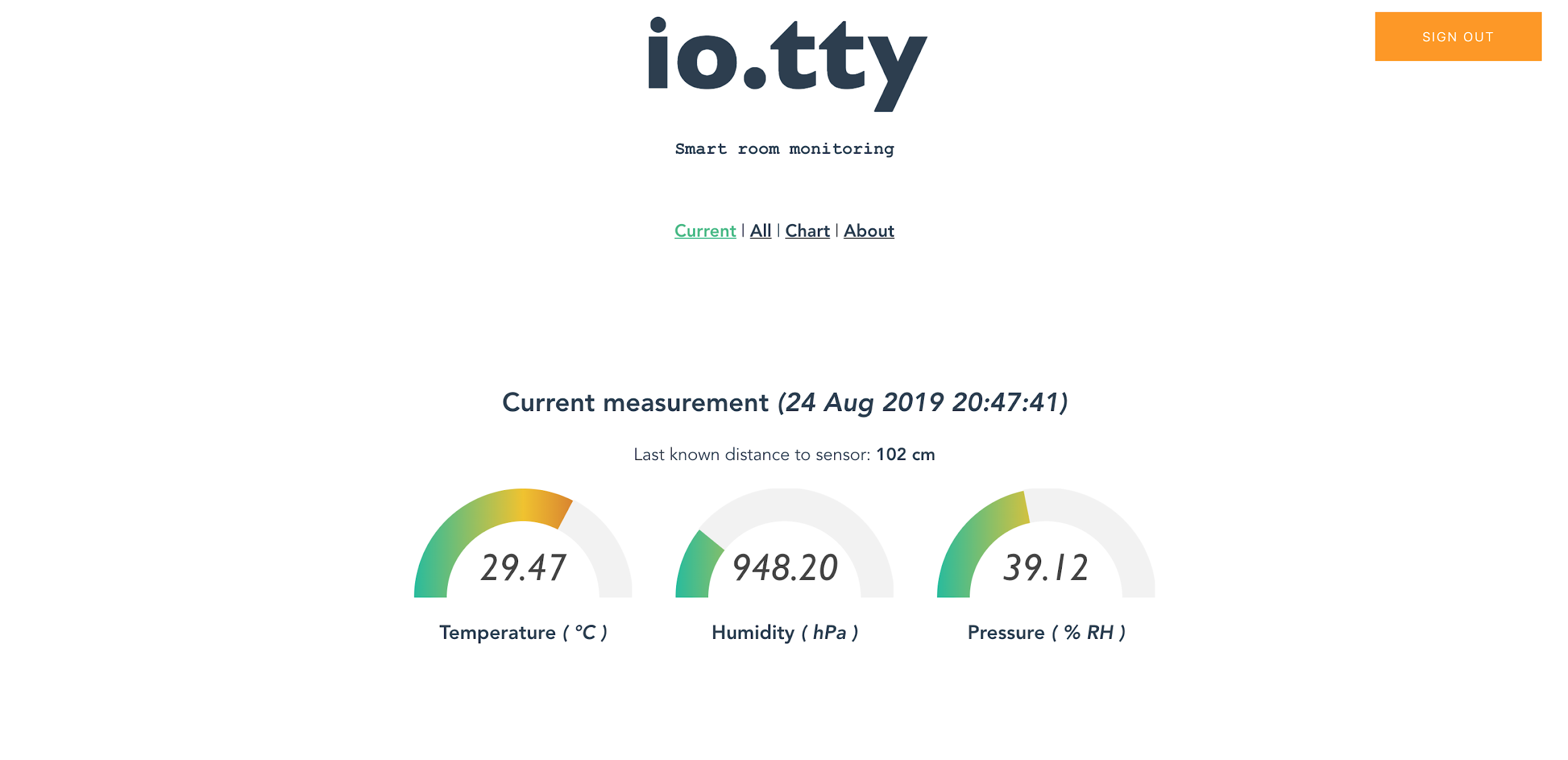
*Клиентското приложение е разделено на отделни domain-и (модули), водейки се от концепциите в Domain Driven Design. Всеки отделен модул отговаря за конкретно entity в приложението (например measurement, user, chart и т.н).*

# Реализация на потребителския интерфейс

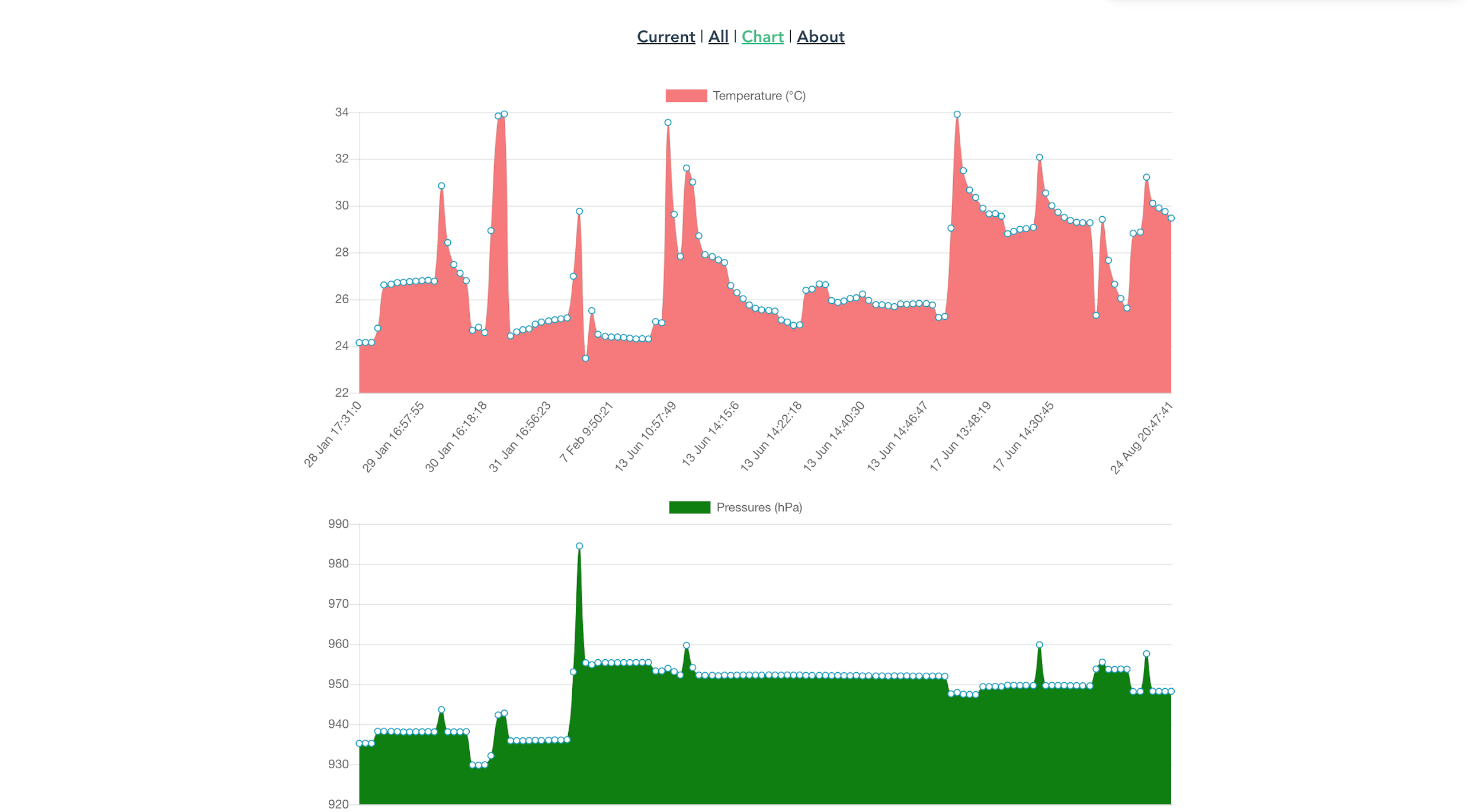
*Потребителският интерфейс представлява клиентско Single Page Application, реализиране с помощта на Vue.js и aws-amplify. Има основна структура на html документа, и всеки един компонент (изглед) се рендерира на определена част от екрана.*

*Екран за вписване в системата:*

**

*Основен екран след вписване на потребител, в който са показани последните измервания: *

*Екран с всички измервания представени като графики, на който лесно може да се видят всички амплитуди:*

**

# Внедряване на системата

*Опишете изискванията и последователността от действия за внедряване на системата.*

*За стартиране на устройството е нужно:*

1. *Да се конфигурира устройството посредством Arduino IDE с правилните данни за вход в интернет мрежата която ще използва*
2. *Micro USB и източник на ток - батерия/лаптоп*

*За стартиране на клиентското приложение са нужни следните действия:*

1. *Изпълнява се командата* ***npm install*** *– инсталира всички пакети, нужни на приложението.*
2. *Изпълнява се командата* ***npm run serve –*** *стартира приложението.*

*Стартирането и конфигурирането на AWS сървисите става от конзолата (AWS console) или от командния ред, независимо от операционната система.*

1. *DynamoDB:*
   1. *създават се полетата в базата от данни ;*
2. *IoT Core:*
   1. *създава се тема (topic) към която устройството използвайки MQTT изпраща данните си;*
   2. *създава се act към DynamoDB с информацията получена от устройството;*
   3. *създава се act към AWS Lambda с информацията получена от устройството;*
   4. *създават се правилните роли и правила за да бъде тази комуникация възможна;*
3. *AWS Lambda:*
   1. *качва се source кодът на lambda-та написан на Node.js и се конфигурира с API ключът нужен от Slack;*
4. *AWS Cognito:* 
   1. *създава се user група с подходящите права за да има достъп за четене от таблицата създадена в точка 1 в DynamoDB;*