**Документална част на проект по дисциплина „Разработване на софтуер за автомобилната индустрия“**

**Тема: Потребителски графичен интерфейс и screen streaming**

**Работили по проекта:  
  
Милен Вълчев  
Радослав Нейков  
Данило Христов  
Петър Христов  
Димитър Станишев**

**Кратко описание на проекта**

Разработва се софтуерно решение, което получава, записва и обработва данни от друг екип, който подава информация относно отчетените стойности от определени сензори ( газов, звуков, ПИР, температурен и сензор за влажност ).

За реализацията на backend логиката се използва Spring framework. Технологична рамка с отворен код за Java платформа. Тя предоставя много функции, които улесняват разработването на Java-базирани enterprise решение.

Пазим данните в релационната база данни с отворен код – PostgreSQL. Предимството й е, че поддържа множество типове данни и е мултиплатформена.

За изграждането на потребителския интерфейс си служим с Vue.js. Подходящ е за нашия случай, тъй като е събирателен за всички добри практики в различните JavaScript билблиотеки. Изключително гъвкав е и може да се използва в широк спектър от архитектурни решения. Допълнителен фактор е малкия размер, което води до по-бързо зареждане на уеб страниците. За бързодействието му се използва виртуален DOM ( Document Object Model ) за намиране на точното място за промяна, без да прегенерира целия реален DOM.

Общото между посочените технологии е, че са:

1. Силно разпространени.
2. Имат много добра документация.
3. Многото развити плъгини допринасят за по-лесната и приятна работа.

**Backend**

Както по-горе описахме, за backend логиката използваме Spring Framework. На изображение 1., имаме „мениджър“, който изважда и записва информация от базата данни.



*Изображение 1*

Решението се осъществява с помощта на два контролера. На изображение 2 е показан контролер, който приема информация от другите системи посредством .json. На изображение 3 е показан контролер, който комуникира с потребителския интерфейс.



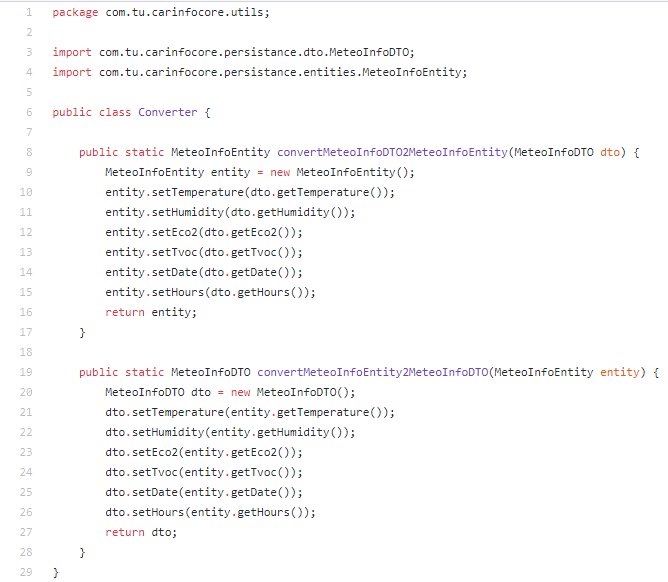
*Изображение 2*



*Изображение 3*

Приемаме модел от друга система – Изображение 4:

* entity записваме в базата
* dto – от интерфейса като го приемаме или изпращаме

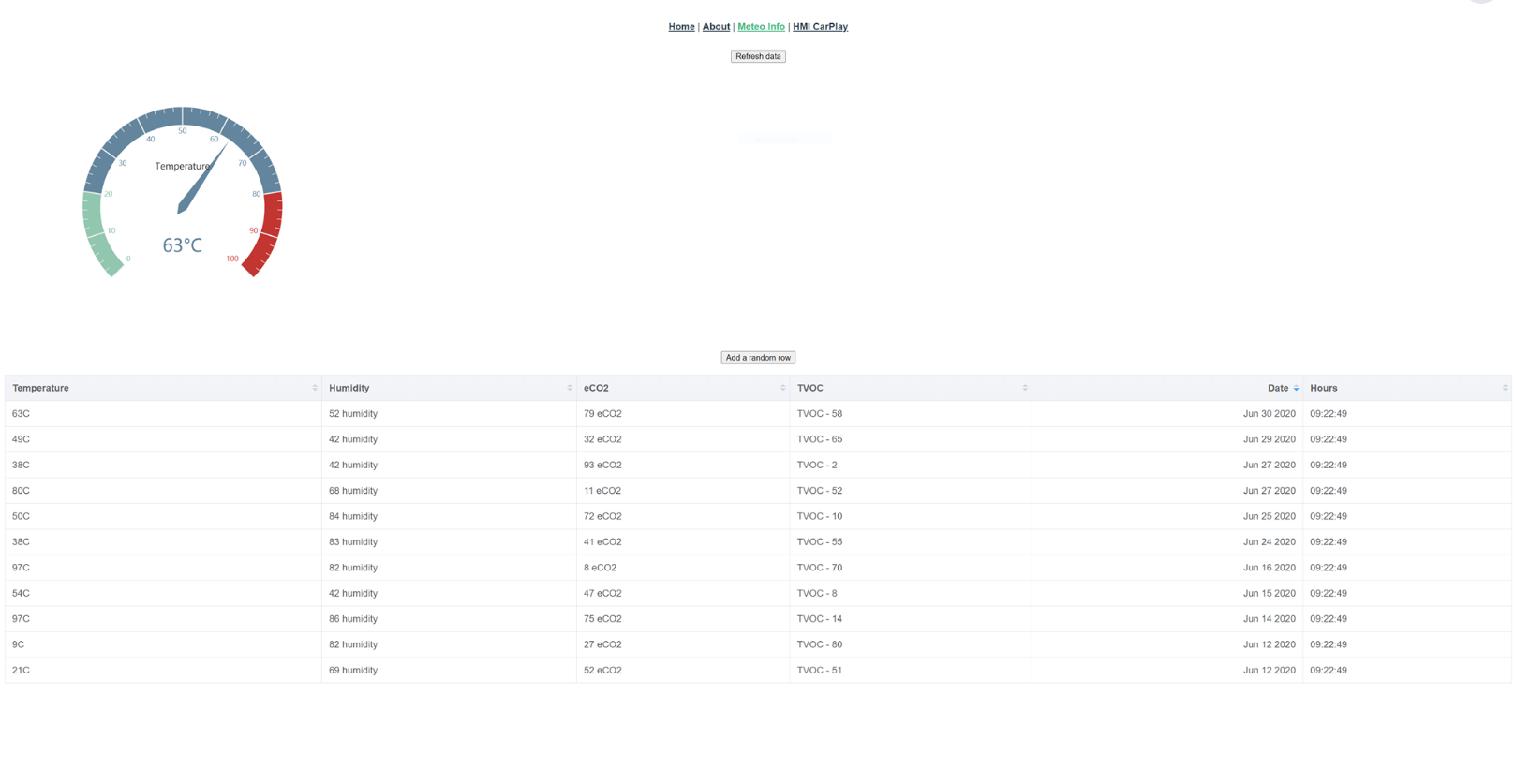


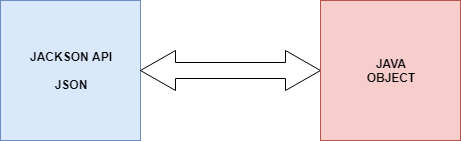
*Изображение 4*

**Потребителски графичен интерфейс**

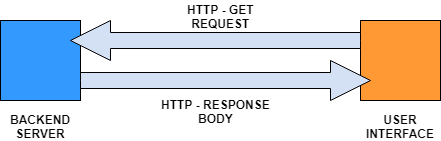
Използваме Vue.js – бърза и лека JavaScript библиотека, която ни дава възможност да се обръшаме към backend логиката с помощта на плъгина Axios. Един от най-популярните Vue.js плъгини за REST повикване. Изключително лесен за използване, като има и добър зключително лесен за използване, като има и добър error handling. Заявките, които правим към нашата логика са, за да вземем данните от метеорологичните сензори (информация, която се съхранява в базата) а след това тази информация визуализираме в таблица. На изображение 5 може да се види как задаваме данни за таблицате и как четем от базата данни. След откъса от кода на страницата, може да се види как изглежда и самата страница на изображение 6 (използваме TempGauge плъгин, за да визуализираме температурата в горния ляв ъгъл).



*Изображение 5*  
*Изображение 6*



*Изображение 7*



*Rest Endpoints*