Oprogramowanie Mind Reader dla urządzenia Epoc emotiv

Dokumentacja - wersja 1.0 - 02.01.2018

# **Spis treści**

[Cele projektu](#_g9bz0go2u8bq)

[Moduły aplikacji](#_dq6i0gomq89a)

[AvereageBandPowers.exe](#_cgzuhzzi1u6j)

[MindReader](#_4n7u23nrzoax)

[Brain Waves Awake.py](#_ddmfdptn3qk6)

[Diagramy](#_pukydcn3gwme)

[Diagram interakcji między modułami aplikacji](#_u8hwok8kmwxi)

[Diagram sekwencji dla odczytu danych](#_bud8amk3wo0j)

[Diagram sekwencji dla odczytu poziomu aktywności](#_t50svmgn8awr)

[Scenariusze testowe](#_pribiz1wcn15)

[Przydatne linki:](#_tgpukedpat6s)

# 

# **Cele projektu**

Epoc emotiv jest to urządzenie do mierzenia aktywności ludzkiego mózgu. Celem projektu było napisanie oprogramowania pozwalającego na wyświetlanie danych pobranych z urządzenia w przystępnej dla użytkownika formie. Aplikacja powinna równiez służyc jako framework do wykonywania różnego rodzajów testów. Powinna pozwalać dynamicznie definiować różnego rodzaju testy. Podczas projektu starano się również dodać elementy pozwalające na analizę aktualnych odczytów przez samo oprogramowanie co nie wymagałoby od użytkownika wiedzy eksperckiej w dziedzinie analizy fal mózgowych.

# **Moduły aplikacji**

Aplikacja podzielona jest na kilka modułów. W celu przejrzystości dokumentacji każdy moduł zostanie krótko omówiony po kolei w tym rozdziale w kolejnych podrozdziałach. Diagramy zależności pomiędzy poszczególnymi modułami znajdują się w kolejnych rozdziałach

## **AvereageBandPowers.exe**

Plik wykonywalny z poprzedniego projektu jaki był realizowany na tym urządzeniu. Jest to aplikacja pobierająca dane bezpośrednio z interfejsu urządzenia. Dane te zapisywane są w pliku .csv który tworzy się w miejscu wykonania pliku wykonywalnego. Aplikacja zapisuje moc różnych fal elektromagnetycznych mierzonych w watach.

Fale których moc jest pobierana przez aplikację AverageBandPowers:

-Alfa

-Low Beta

-High Beta

-Gamma

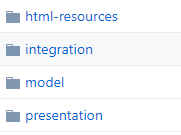
-Theta

## 

## **MindReader**

Aplikacja rozwijana w ramach tego projektu. Jest to aplikacja pobierająca dane z pliku .csv tworzonego przez aplikację AverageBandPowers. Oprogramowanie zostało napisane w języku C#. Do napisania layoutu aplikacji użyto biblioteki CefSharp umożliwiającej użycie chromium jako komponentu wypełniającego okno aplikacji. Pozwoliło to na użycie technologii webowych do stworzenia widoku aplikacji.

Kod źródłowy aplikacji został podzielony na następujące pakiety



* html-resources - odpowiada za warstwę front-endu aplikacji
* integration - odpowiada za integrację back-endu aplikacji z innymi modułami aplikacji (AverageBandPowers i Python)
* model - klasy modelowe potrzebne w aplikacji
* presentation - odpowiada za interakcję back-endu aplikacji z generowanych w chromium front-endem

## **Brain Waves Awake.py**

Moduł ten został napisany w języku Python. Pozwala on na określeni stanu skupienia badanego człowieka na podstawie mocy fal wysyłanych do skryptu

Wywołanie programu:

Skrypt wywoływany jest jak każdy inny skrypt w języku Python. Argumenty linii poleceń jakie należy podać w celu wywołania skryptu to:

-Moc fal alfa

-Moc fal beta

-moc fal gamma

-czy mamy defuzyfikować wynik (-fuzz)

Analiza aktywności badanej osoby oparta jest o logikę rozmytą. W tym celu wykorzystano bibliotekę scikit-fuzzy 0.3.1. Dokumentacja biblioteki dostępna pod adresem:

<https://pypi.python.org/pypi/scikit-fuzzy>

# 

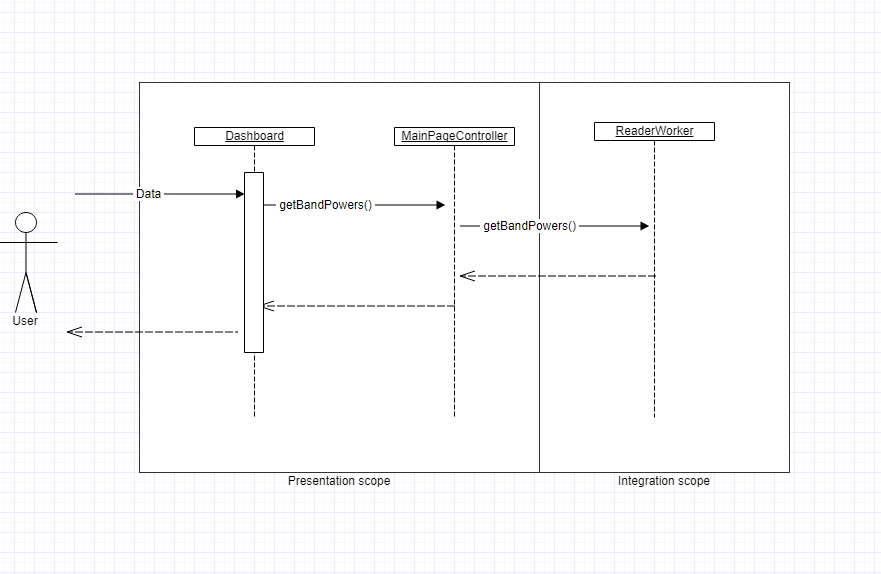
# 

# 

# **Diagramy**

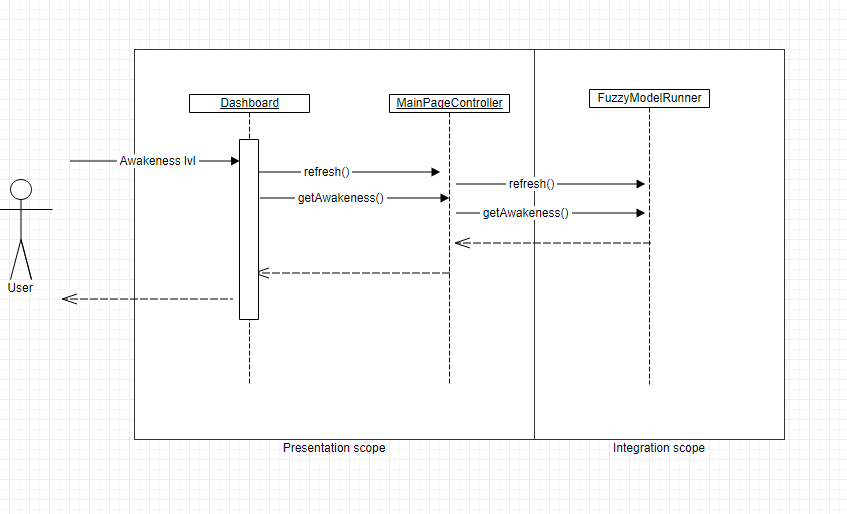
## **Diagram interakcji między modułami aplikacji**

## **Diagram sekwencji dla odczytu danych**



## 

## **Diagram sekwencji dla odczytu poziomu aktywności**



# 

# 

# 

# 

# **Scenariusze testowe**

Jedną z funkcjonalności wprowadzonych przez aplikację jest możliwość dynamicznego tworzenia listy scenariuszów testowych. W aplikacji funkcjonalność tą stworzono po stronie front-endu aplikacji. Scenariusze ładowane są z pliku JSON, przez skrypt dataLoader.js.

Przykładowy plik z testami:

{

"tests": [

{

"id": 1,

"name": "Audio test",

"description": "Brainwave entrainment, also referred to as brainwave synchronization and neural entrainment,",

"media": "https://www.youtube.com/watch?v=t3217H8JppI"

},

{

"id": 2,

"name": "Video test",

"description": "Test should stimulate alpha brain waves",

"media": "https://www.youtube.com/watch?v=J73kveWTj68"

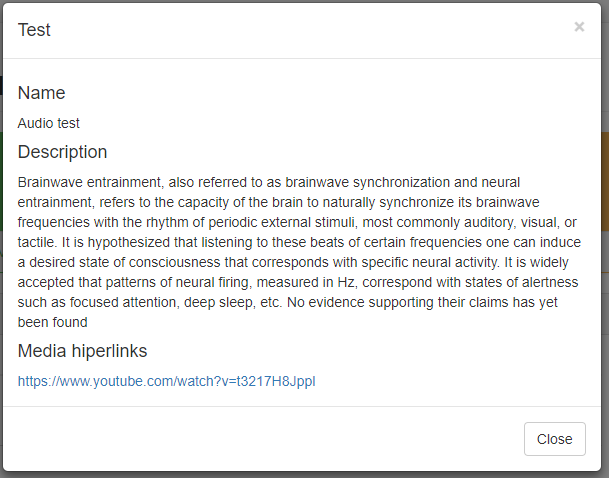
}

]

}

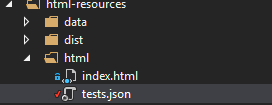
Opis pól pliku ze scenariuszami testowymi:

* id - unikalny identyfikator testu
* name - nazwa testu
* description - opis testu, warunki przeprowadzenia testu, co chcemy badać
* media - link do mediów które będą powodowały bodźce

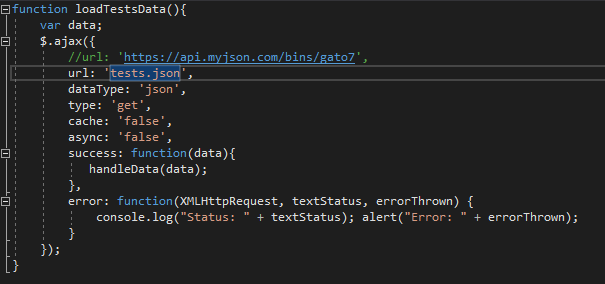


Powyżej widzimy modal opisujący test audio wypełniony danymi z pliku tests.json

Plik opisujący scenariusze testowe powinien znajdować się w katalogu html-resources w podkatalogu hmtl:



W wypadku ewentualnej zmiany lokalizacji tego pliku należy dokonać zmiany w kodach źródłowych aplikacji w pliku /html-resouces/js/dataLoader.js



# **Przydatne linki:**

1. [**https://github.com/AdamGuar/MindReader**](https://github.com/AdamGuar/MindReader) **-** repozytorium projektu
2. <http://www.brainworksneurotherapy.com/what-are-brainwaves> - Krótka charakterystyka mierzonych fal
3. <https://www.scientificamerican.com/article/what-is-the-function-of-t-1997-12-22/> - Artykuł o falach EEG
4. <http://pythonhosted.org/scikit-fuzzy/> - bliblioteka fuzzy logic dla Python
5. <https://github.com/cefsharp/CefSharp> - biblioteka CEFSharp
6. <https://ourcodeworld.com/articles/read/173/how-to-use-cefsharp-chromium-embedded-framework-csharp-in-a-winforms-application> - tutorial jak używać bilbioteki CEFSharp