

ZAPYTANIE O INFORMACJĘ NA ZAIMPLEMENTOWANIE API DLA ZEWNĘTRZNYCH SYSTEMÓW

„QNeuro – zintegrowany e-system wspierający diagnostykę i prowadzenie chorych na padaczkę” współfinansowany ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Inteligentny Rozwój lata 2014-2020, realizowany w ramach poddziałania 1.1.2 PO IR Prace B+R związane z wytworzeniem instalacji pilotażowej/demonstracyjnej w konkursie 2/1.1.2/2015 PO IR.

RFI-8.2-2018_Zaimplementowanie_API_dla_zew_systemow_v1.0

SIEDZIBA

ENQuant Sp. z o.o.

ul. Marii Skłodowskiej-Curie 41; 87-100 Toruń

www.enquant.pl

enquant
NEW HORIZON OF IMAGINATION

TYP DOKUMENTU

Zapytanie o informację (RFI)

NAZWA PROJEKTU

Projekt pt.: „QNeuro – zintegrowany e-system wspierający diagnostykę i prowadzenie chorych na padaczkę” współfinansowany ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Inteligentny Rozwój lata 2014-2020. Projekt jest realizowany w ramach poddziałania 1.1.2 PO IR Prace B+R związane z wytworzeniem instalacji pilotażowej/demonstracyjnej w konkursie 2/1.1.2/2015 PO IR.

NUMER ZAMÓWIENIA

RFI-8.2-2018_Zaimplementowanie_API_dla_zew_systemow_v1.0

OSOBA KONTAKTOWA

Karolina Słowikowska, mail: karolina.slowikowska@enquant.pl mobile +48 782 992 683

DATA UTWORZENIA

19.10.2018

DATA WAŻNOŚCI

29.10.2018

ZAMAWIAJĄCY

enQuant Sp. z o.o. ul. Marii Skłodowskiej-Curie 41; 87-100 Toruń, KRS 0000325333, Regon 340549671, NIP 9562248388

OSOBA UPOWAŻNIONA

Jacek Mazurek – Prezes Zarządu

Spis treści

1. Opis przedmiotu zamówienia.....	4
2. Architektura systemu QNeuro.....	5
3. Zakres prac.....	6
4. Szczegółowe wymagania dla dostarczonych produktów.....	7
4.1. Analiza, specyfikacja i implementacja mEngine API w mobilnej oraz webowej aplikacji Pacjent	7
4.1.1. Funkcjonalności aplikacji Pacjent.....	7
4.1.2. Wymagania architektoniczne dla mobilnej oraz webowej aplikacji Pacjent	8
4.1.3. Zakres prac integracyjnych.....	8
4.2. Analiza, specyfikacja i implementacja mEngine API w mobilnej oraz webowej aplikacji Lekarz	8
4.2.1. Funkcjonalności aplikacji Lekarz.....	9
4.2.2. Wymagania architektoniczne aplikacji mobilnej Lekarz.....	10
4.2.3. Moduł konsultanta.....	10
4.2.4. Zakres prac integracyjnych.....	10
4.3. Analiza, specyfikacja i implementacja mEngine API w aplikacji mobilnej homeEEG	10
4.3.1. Przeznaczenie aplikacji:	10
4.3.2. Użytkownik aplikacji:.....	11
4.3.3. Funkcje / wymagania funkcjonalne aplikacji:	11
4.3.4. Łączenie aplikacji z rejestratorem i przeprowadzenie badania:	11
4.3.5. Praca aplikacji z rozszerzonym stanowiskiem homeVEEG + homeEEG.....	12
4.3.6. Pozostałe wymagania i cechy aplikacji.....	12
4.3.7. Obsługa komunikatów.....	13
4.3.8. Zakres prac integracyjnych.....	14
4.4. Wspólne wymagania dla aplikacji	14
4.4.1. Wygląd.....	14
4.4.2. Dokumentacja	15
4.4.3. Bezpieczeństwo.....	15
5. Formaty danych i API.....	16
5.1. Przykładowe komendy i format plików rejestratora homeEEG.....	16
5.2. Przykładowa specyfikacja API mEngine.....	19
5.2.1. Moduł Lekarza.....	19
5.2.2. Moduł Pacjenta.....	32
6. Wycena oraz harmonogram realizacji prac.....	45
7. Forma odpowiedzi na niniejsze RFI.....	46
8. Termin składania odpowiedzi na RFI.....	47
9. Spis załączników/oświadczeń wymaganych od Oferenta.....	48

1. Opis przedmiotu zamówienia

Niniejszy dokument zawiera specyfikację prac rozwojowych systemu QNeuro na potrzeby zapytania RFI dla usług zewnętrznych.

System QNeuro zakłada stworzenie rozwiązania informatycznego mającego gromadzić i analizować dane medyczne w celach diagnostyki i monitorowania stanu zdrowia chorych na padaczkę. Dane medyczne zostaną włączone do zintegrowanej bazy danych, zawierającej ponadto na bieżąco aktualizowaną historię choroby pacjentów, bazę lekarzy oraz moduły umożliwiające kontakt z lekarzem na odległość. Dzięki takiemu rozwiązaniu pacjent będzie mógł bez długiego oczekiwania na wizytę wybrać jednego spośród wszystkich zarejestrowanych w systemie lekarzy specjalistów skorzystać z jego pomocy zdalnie (wybór na podstawie zdefiniowanych w systemie danych: wiedza, dorobek, umiejętności danego lekarza). Lekarz natomiast zdobędzie wsparcie w procesie diagnostyki i terapii:

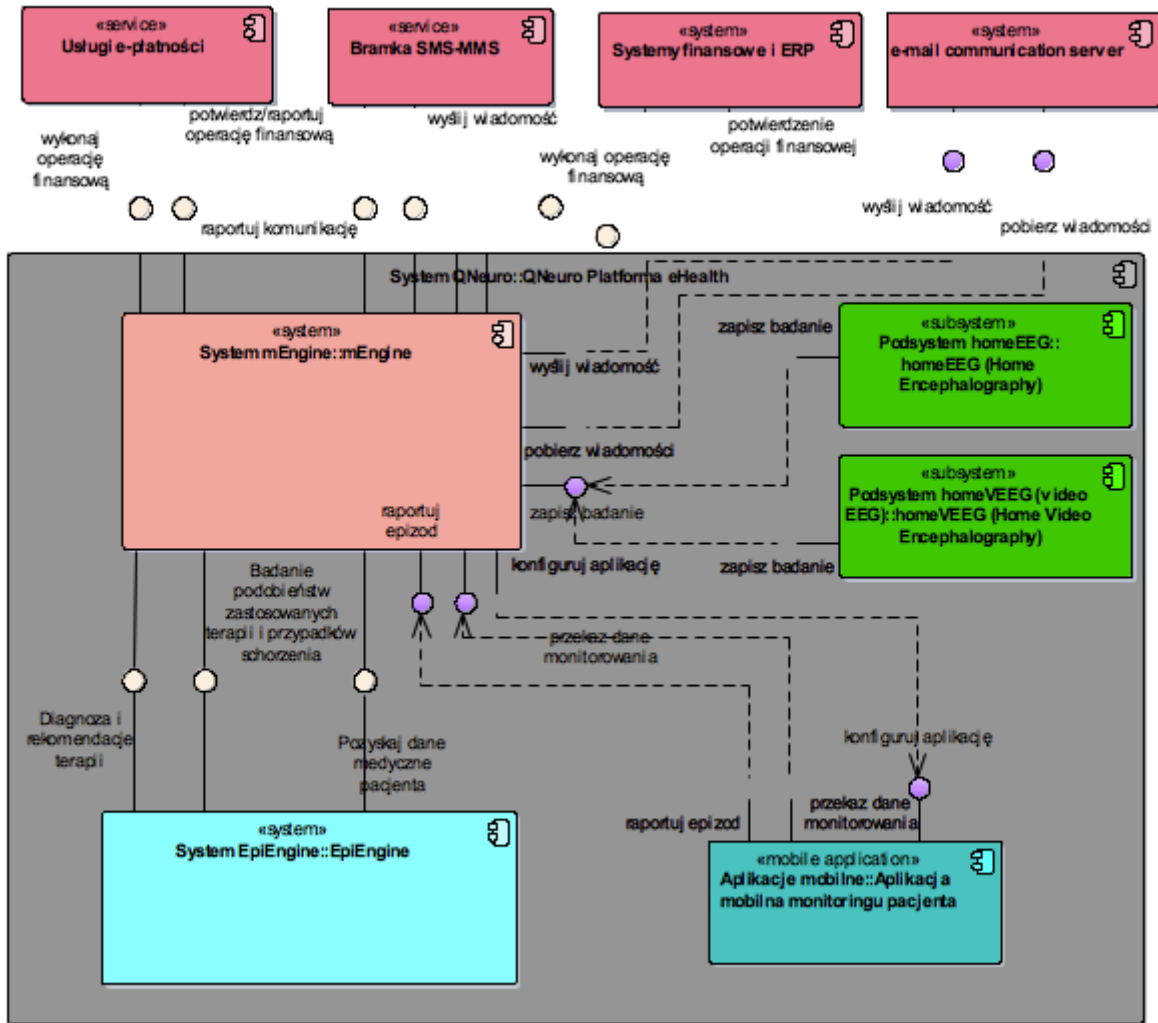
- uzyska dostęp do systemu wspierającego jego decyzje medyczne, które dodatkowo w ramach QNeuro będzie mógł konsultować z innymi zarejestrowanymi w systemie specjalistami (bez bariery czasu oraz lokalizacji),
- uzyska dostęp do systemu pozwalającego na wgląd w kompletną dokumentację medyczną pacjentów, przedstawioną w postaci dostosowanej do specyfiki leczenia pacjentów chorujących na padaczkę.

Elementami systemu QNeuro będą urządzenia homeEEG i homeVEEG, służące do pomiarów bez konieczności wizyty w placówce ochrony zdrowia (i bez wcześniejszego oczekiwania). Dane z pomiarów będą przesyłane zdalnie do systemu. Urządzenia te wpłyną na usprawnienie procesu diagnostyki i terapii oraz zmniejszą koszt opieki (eliminacja kosztów wykorzystania drogich pracowni EEG, kosztów przejazdów do placówek ochrony zdrowia, etc.).

Elementy składające się na moduł mEngine są niezbędne ze względu na konieczność zapewnienia pełnej funkcjonalności systemu QNeuro, takie jak ułatwienie kontaktu z lekarzem, stałe monitorowanie stanu zdrowia pacjentów objętych opieką, prowadzenie kompletnej dokumentacji medycznej pacjentów, zapewnienie bezpieczeństwa przechowywanych i przetwarzanych danych, a także udostępnienie API dla systemów zewnętrznych. Bez tych elementów system QNeuro będzie niepełny i nie będzie możliwe przeprowadzenie pełnego procesu diagnostyki, doboru terapii i nadzoru nad jej przebiegiem wobec pacjentów, nad którymi sprawowana jest opieka.

2. Architektura systemu QNeuro

cmp 6. Model Komponentów Systemu - HLD interfaces



Rys. 1 Komponenty składowe QNeuro wraz z elementami (systemami) zewnętrznymi przewidywanymi do integracji i wsparcia działania podsystemów funkcjonalnych mEngine

3. Zakres prac

W ramach prac implementacji API do systemów zewnętrznych, zrealizowane będą następujące prace i produkty:

1. Analiza i implementacja mEngine API w mobilnej oraz webowej aplikacji Pacjent
2. Analiza i implementacja mEngine API w mobilnej oraz webowej aplikacji Lekarz
3. Analiza i implementacja mEngine API oraz implementacja API urządzenia homeEEG/VEEG w aplikacji mobilnej homeEEG

Zakres prac nie obejmuje implementacji API po stronie systemu mEngine.



4. Szczegółowe wymagania dla dostarczonych produktów

4.1. Analiza, specyfikacja i implementacja mEngine API w mobilnej oraz webowej aplikacji Pacjent

4.1.1. Funkcjonalności aplikacji Pacjent

Autoryzacja użytkownika w systemie

Skrót najistotniejszych informacji medycznych

- Aktualna diagnoza
- Aktualna terapia lekowa (wraz z aktualnym stanem zaopatrzenia w leki)
- Zdiagnozowane rodzaje napadów padaczkowych (wraz z trendem ich występowania)
- Dane lekarza prowadzącego
- Zaplanowane wizyty i zdarzenia medyczne

Obsługa komunikacji z prowadzącym personelem medycznym

- Funkcjonalność inicjowania komunikacji z lekarzem prowadzącym
- Powinna być możliwość przejrzenia historii komunikacji pacjenta z innymi użytkownikami systemu.
- Komunikaty "emergency" (automatyczne wysłanie zdefiniowanego przez użytkownika komunikatu do wybranych osób kontaktowych)
- Możliwość wysłania do lekarza prowadzącego prośby o wystawienie recepty na leki.

Przypominanie o przyjmowaniu leków

- Powiadomienia o określonych porach dnia
- Możliwość oznaczenia, że lek został przyjęty / nie przyjęty
- Funkcjonalność "drzemki" (możliwość odłożenia w czasie odpowiedzi na pytanie o przyjęcie leków).

Kalendarz napadów

- Rejestrowanie napadów i innych zdarzeń medycznych (obejmuje dodawanie, edycję i usuwanie zdarzeń medycznych)
- Prezentacja kalendarza napadów pacjenta.

Wyraźne informowanie o zaplanowanych czynnościach medycznych

- Zaplanowanych wizytach
- Niezrealizowanych skierowaniach
- Kończących się lekach
- Zaplanowanych zdarzeniach (zaplanowanych kontaktach, badaniach, lub innych czynnościach ustalonych z lekarzem prowadzącym podczas konsultacji)

Obsługa powiadomień (wyświetlanie i zarządzanie wiadomościami generowanymi przez system, archiwizacja wiadomości).

Saldo pacjenta

- Stan konta
- Lista operacji finansowych
- Możliwość zrealizowania płatności

4.1.2. Wymagania architektoniczne dla mobilnej oraz webowej aplikacji Pacjent

Aplikacja powinna pozwalać na pracę pacjenta off-line, jeżeli aplikacja nie ma dostępu do sieci powinna mieć zapisany ostatni pobrany stan choroby pacjenta. Minimalny zakres funkcjonalności działających off-line:

- wprowadzanie danych o napadach i innych zdarzeniach medycznych,
- wprowadzanie danych o przyjętych lekach.
- Przypominania o lekach
- Kalendarz wizyt

Po uzyskaniu dostępu do internetu dane wpisane off-line powinny zostać zsynchronizowane z systemem QNeuro.

4.1.3. Zakres prac integracyjnych

Zakres prac w ramach integracji aplikacji Pacjent obejmuje:

1. analizę, projekt i dostarczenie specyfikacji API mEngine zgodnie z załączoną specyfikacją
2. projekt i implementacja aplikacji mobilnej zintegrowanej z API mEngine dla systemów Android i iOS wraz z udostępnieniem tych aplikacji w Google Play oraz Apple App Store.
3. projekt i implementacja aplikacji webowej zintegrowanej z API mEngine

4.2. Analiza, specyfikacja i implementacja mEngine API w mobilnej oraz webowej aplikacji Lekarz

4.2.1. Funkcjonalności aplikacji Lekarz

Lista pacjentów znajdujących się pod opieką lekarza

- Krótkie podsumowanie stanu zdrowia pacjenta (etap diagnostyki, skrócone informacje z konsultacji lekarskich, aktualna terapia lekowa, opisy wyników badań, kalendarz napadów i zdarzeń medycznych, zagregowane informacje o nasileniu lub zmniejszeniu ilości napadów padaczkowych, oraz o poziomie realizacji przez pacjenta ustalonej terapii lekowej)
- Wywoływanie akcji w kontekście pacjenta:
 - komunikacja z pacjentem
 - obsługa kalendarza wizyt

Obsługa komunikacji między lekarzem a pacjentami znajdującymi się pod opieką lekarza

- Powiadamanie o nowych wiadomości wysłanych przez pacjentów znajdujących się pod opieką lekarza.
 - W ich kontekście powinna być możliwość przejrzenia historii komunikacji między lekarzem a pacjentem
 - Powinny być uwzględnione mechanizmy filtrowania i oznaczania wiadomości jako istotnej medycznie
- Odpowiadanie na wiadomości pacjentów

Kalendarz lekarza

- Terminarz wizyt, z krótką informacją o stanie zdrowia pacjentów.
 - Umawianie, przenoszenie i usuwanie wizyt.
- Zgłaszanie urlopów i zmian w planie pracy.

Wsparcie w realizacji konsultacji telefonicznych

- Inicjowanie rozmowy na podstawie terminarza lekarza
- Rejestrowanie czasu trwania rozmowy i oznaczanie jej jako odbytej.

Obsługa powiadomień

- Wyświetlanie oraz zarządzanie powiadomieniami systemowymi. Archiwizacja wiadomości systemowych.
- Wyświetlanie przypomnienia o zaplanowanych zdarzeniach dotyczących pacjentów pozostających pod opieką lekarza.

Saldo lekarza

- Stan konta

- Lista operacji finansowych
- Możliwość zrealizowania płatności

4.2.2. Wymagania architektoniczne aplikacji mobilnej Lekarz

Aplikacja powinna pozwalać na pracę lekarza off-line, jeżeli aplikacja nie ma dostępu do sieci powinna mieć zapisany ostatni pobrany stan worklist lekarza. Minimalny zakres funkcjonalności działających off-line:

kalendarz wizyt (ostatni zapisany stan).

4.2.3. Moduł konsultanta

Moduł konsultanta umożliwi zaangażowanie w konsultację dodatkowych lekarzy wraz z dostępem do historii choroby oraz wyników badań.

Aplikacja powiadomi konsultanta o nowych zleceniach spływających do systemu.

4.2.4. Zakres prac integracyjnych

Zakres prac w ramach integracji aplikacji Lekarz obejmuje:

- analizę, projekt i dostarczenie specyfikacji API mEngine zgodnie z załączoną specyfikacją
- projekt i implementacja aplikacji mobilnej zintegrowanej z API mEngine dla systemów Android i iOS wraz z udostępnieniem tych aplikacji w Google Play oraz Apple App Store.
- projekt i implementacja aplikacji webowej zintegrowanej z API mEngine

4.3. Analiza, specyfikacja i implementacja mEngine API w aplikacji mobilnej homeEEG

4.3.1. Przeznaczenie aplikacji:

Celem aplikacji mobilnej dla urządzeń Android i iOS jest opracowanie narzędzia które będzie współpracując z aparatem homeEEG pozwalało na przygotowanie, konfigurację i przeprowadzenie badania EEG. Aplikacja ma umożliwiać również asynchroniczną wysyłkę zarejestrowanych danych. Aplikacja mobilna powinna umożliwiać retencje historii przeprowadzonych badań, pokazywanie statusów wysyłanych danych na serwer. Aplikacja zapewni niezaprzeczalność i spójność przekazywanych danych. Aplikacja mobilna ma korzystać z kanału komunikacji BLE i WiFi w zależności od wykonywanych operacji konfiguracji lub odbioru danych z wielokanałowego EEG. Zarejestrowane dane powinny być możliwe do przejrzania. Należy zapewnić również utrzymanie profilu pacjenta w aplikacji.

Aplikacja mobilna jest przeznaczona do wykonywania badania EEG w warunkach domowych. Stanowi ona interfejs urządzenia – rejestratora EEG („homeEEG”). Aplikacja odpowiada bezpośrednio za przygotowanie badania, zarejestrowanie sygnału EEG oraz przesłanie wyników na serwer. Aplikacja komunikuje się z urządzeniem homeEEG jednocześnie za pomocą sieci WiFi i Bluetooth, natomiast z serwerem poprzez sieć

Internet dostępną na urządzeniu mobilnym. Aplikacja musi być przystosowana do pracy w dwóch trybach: z urządzeniem homeEEG oraz w wersji rozszerzonej, z dołączonym stanowiskiem homeVEEG.

Zakres prac obejmuje analizę, projekt i dostarczenie API wraz z aplikacją mobilną dla systemów Android i iOS. W ramach prac wymagane jest udostępnienie aplikacji mobilnych w Google Play oraz Apple App Store.

4.3.2. Użytkownik aplikacji:

Użytkownikiem aplikacji jest osoba, która we własnym zakresie będzie wykonywała sobie badanie EEG lub opiekun badanego. Zwykle jest to użytkownik o profilu nietechnicznym, wobec czego aplikacja powinna być bardzo intuicyjna, a etap jej konfiguracji i uruchomienia powinien być zbliżony do aplikacji typu wizard.

4.3.3. Funkcje / wymagania funkcjonalne aplikacji:

Główną funkcją aplikacji jest poprowadzenie użytkownika przez proces badania EEG. W tym celu aplikacja musi zweryfikować dostępność urządzenia homeEEG, sprawdzić połączenie z serwerem, sprawdzić ważność licencji użytkownika lub opcjonalnie zarejestrować użytkownika i urządzenie, sprawdzić aktualizację oprogramowania, natomiast w procesie badania wyświetlić użytkownikowi niezbędne wskazówki odnośnie założenia czepka EEG, sprawdzenia statusu urządzenia rejestrującego, sprawdzenia poprawności założenia czepka (pomiar impedancji elektrod). W procesie badania aplikacja musi zarejestrować sygnał EEG przesyłany przez rejestrator, opatrzyć go odpowiednimi identyfikatorami (id użytkownika, data, komentarz użytkownika, wersja firmware) a następnie przesłać na serwer.

Identyfikacja urządzenia, do którego ma być dowieziona aplikacja będzie realizowane poprzez wpisanie w aplikację numeru seryjnego stanowiska homeEEG/homeVEEG lub zeskanowanie umieszczonego na nim kodu QR.

Aplikacja mobilna jest dołączana do urządzenia mobilnego najpierw poprzez sieć Bluetooth. Aplikacja musi najpierw odnaleźć rejestrator i sparować się z nim poprzez PIN (na rejestratorze i w dokumentacji). Następnie poprzez Bluetooth aplikacja konfiguruje rejestrator, sprawdza jego status, wersję firmware oraz informacje dotyczące połączenia WiFi. Poprzez sieć WiFi urządzenie będzie wymieniało pliki z rejestratorem poprzez TFTP.

4.3.4. Łączenie aplikacji z rejestratorem i przeprowadzenie badania:

1. Numer seryjny rejestratora i dane dostępowe wpisane w aplikacje lub zeskanowane z etykiet urządzeń.
2. Rejestrator jest wykrywalny przez Bluetooth.
3. Aplikacja łączy się z rejestratorem poprzez Bluetooth i PIN.
4. Aplikacja weryfikuje numer seryjny rejestratora, status, wersję oprogramowania. W tym celu weryfikuje te dane z bazą na serwerze. Ponadto aplikacja dostaje informacje o sieci WiFi rejestratora. Szczegóły rejestracji do ustalenia na etapie implementacji.
5. Aplikacja dołącza się do rejestratora poprzez WiFi (równolegle do łączności Bluetooth). Poprzez sieć WiFi wymienia pliki. Nadajnik WiFi rejestratora jest wstrzymywany na czas rejestracji EEG.
6. Aplikacja konfiguruje rejestrator, sprawdza poprawność podłączenia elektrod i informuje o statusie użytkownika. Jeśli instalacja elektrod jest poprawna możliwe jest rozpoczęcie badania.
7. Działanie całego procesu aplikacji powinno mieć charakter wizard'a, który będzie prowadził użytkownika

- przez proces realizacji badania.
8. Rozpoczęcie badania uruchamiane jest przyciskiem na urządzeniu lub poprzez aplikację. Po jego przeprowadzeniu (w zadanym przez aplikację czasie) jest ono kończone w analogiczny sposób.
 9. Po zakończeniu rejestracji pliki z danymi (eeg, akcelerometr, log) przesyłane są do aplikacji.
 10. Aplikacja wysyła zarejestrowane pliki na serwer.

4.3.5. Praca aplikacji z rozszerzonym stanowiskiem homeVEEG + homeEEG

W trybie rozszerzonym aplikacja musi współpracować jednocześnie z rejestratorem i stanowiskiem VEEG, które służy do rejestracji obrazu (przebiegu badania EEG). Obraz rejestrowany jest za pomocą dwóch kamer HD dołączonych do komputera homeVEEG.

Aplikacja musi umożliwiać przełączenie trybu pracy w tryb współpracy z homeVEEG. W tym przypadku homeVEEG rozgłasza własną sieć WiFi oraz jest wykrywalne przez Bluetooth. Dane dostępne stanowiska homeVEEG i numer seryjny umieszczone są na etykiecie w sposób tekstowy i kodem QR.

Aplikacja powinna przesłać najpierw do rejestratora homeEEG dane do dołączenia go do stanowiska homeVEEG (informację o sieci WiFi i klucz oraz informację o sieci Bluetooth i pin). Po przesłaniu informacji rejestrator powinien dołączyć się do stanowiska homeVEEG. Aplikacja powinna po tym dołączyć się do stanowiska homeVEEG do rozgłaszanej przez nie sieci WiFi. HomeVEEG poprzez tą sieć będzie mógł udostępniać również sieć Internet, jeśli zostanie do niej dołączony. W tej konfiguracji homeVEEG otrzymuje komendy od aplikacji, wykonuje je lub przesyła do rejestratora homeEEG za pomocą WiFi lub Bluetooth.

Dane do logowania do homeVEEG dostępne są na etykiecie znajdującej się na stanowisku w formie tekstowej oraz kodu QR z kluczem do WiFi. Aplikacja powinna umożliwiać odczytanie kodu QR za pomocą aparatu urządzenia mobilnego.

Obraz video rejestrowany jest wyłącznie na komputerze stanowiska homeVEEG i z niego poprzez Internet przesyłany na serwer.

W celu prawidłowego ustawienia kamer na stanowisku homeVEEG aplikacja mobilna musi umożliwiać podgląd obrazu. Podgląd ten musi umożliwiać ustawienie każdej z kamer (jednej na twarz badanego, drugiej na sylwetkę). Kamery ustawiane są przez badanego ręcznie. Aplikacja musi udostępniać obraz na ekranie urządzenia mobilnego, w zmniejszonej rozdzielczości i przy niewielkiej liczbie klatek/s. Optymalne parametry obrazu zostaną ustalone podczas testów.

Po rozpoczęciu rejestracji aplikacja musi wysłać komendę do stanowiska homeVEEG, która uruchomi rejestrację obrazu.

Po zakończeniu rejestracji sygnały z homeEEG oraz obraz z homeVEEG muszą być przesłane na serwer. Opcjonalnie sygnały z homeEEG mogą być przesłane na homeVEEG, a następnie mogą być wysłane z homeVEEG na serwer. Po przesłaniu sygnałów stanowisko homeVEEG powinno wyłączyć się automatycznie.

4.3.6. Pozostałe wymagania i cechy aplikacji

1. Aplikacja powinna umożliwiać przypisanie jej i urządzenia do użytkownika systemu QNeuro
2. Urządzenie homeEEG / homeVEEG ma unikalny identyfikator, który może być przypisany tylko jednemu użytkownikowi systemu QNeuro
3. Urządzenie homeEEG / homeVEEG powinno potrafić zalogować się do systemu QNeuro automatycznie poprzez dane wprowadzone przez użytkownika oraz wydrukowane na etykietach urządzeń.

4. Aplikacja mobilna musi umożliwiać aktualizację oprogramowania homeEEG. W tym celu musi zweryfikować dane rejestratora oraz przeprowadzić proces upgrade'u. W tym celu aplikacja pobiera plik z firmwarem z serwera (producenta), wysyła go do urządzenia homeEEG (poprzez TFTP) a następnie restartuje urządzenie. Po restarcie urządzenie dokonuje aktualizacji oprogramowania automatycznie za pomocą wbudowanego bootloadera.

4.3.7. Obsługa komunikatów

1. Aplikacja homeEEG / vEEG powinna potrafić przekazać do systemu QNeuro ustalone komunikaty kontrolne:
 - 1.1. Brak miejsca na nowe badania diagnostyczne
 - 1.2. Błąd w czasie przesyłania danych
 - 1.3. Błąd w konfiguracji urządzenia
 - 1.4. System homeEEG / vEEG potrafi rozpoznać i zareagować na ustalone komunikaty z systemu QNeuro
 - 1.5. Wyczerpanie przez użytkownika przydzielonej przestrzeni na pliki multimedialne (w systemie QNeuro)
 - 1.6. Utrata praw do korzystania z systemu QNeuro przez pacjenta / placówkę medyczną
 - 1.7. Przechowywanie i transfer plików z zapisem badań diagnostycznych
 - 1.8. Urządzenie homeEEG / homevEEG przechowuje informacje o plikach, wysłanych na serwer QNeuro
 - 1.9. Urządzenie homeEEG / homevEEG potrafi wznowić przerwany proces wysyłki danych medycznych, ponowić wysyłkę jeżeli nie jest możliwe wznowienie transferu.
 - 1.10. Urządzenie homeEEG / homevEEG pozwala na przechowanie określonej ilości (zależnej od konfiguracji sprzętowej) zapisów badań.
 - 1.11. system homeEEG / homevEEG ostrzega o kończącym się miejscu na zapisy badań diagnostycznych
 - 1.12. system homeEEG / homevEEG w przypadku wyczerpania miejsca na składowanie zapisów usuwa zapisy najstarsze, które zostały już wysłane do systemu QNeuro
 - 1.13. Minimalny zakres danych do przesyłanych z systemu homeEEG
 - 1.14. data i czas zapisu
 - 1.15. identyfikator urządzenia
 - 1.16. dane zapis pozwalające na przetworzenie sygnałów zebranych na urządzeniu (dane cząstkowe pomiarów)
 - 1.17. każdy pomiar powinien być sygnowany czasem
 - 1.18. dane cząstkowe próbek powinny być możliwe do nałożenia na dane zapis obrazu z kamer urządzenia vEEG
 - 1.19. Urządzenie wykorzystywane w placówce służby zdrowia



4.3.8. Zakres prac integracyjnych

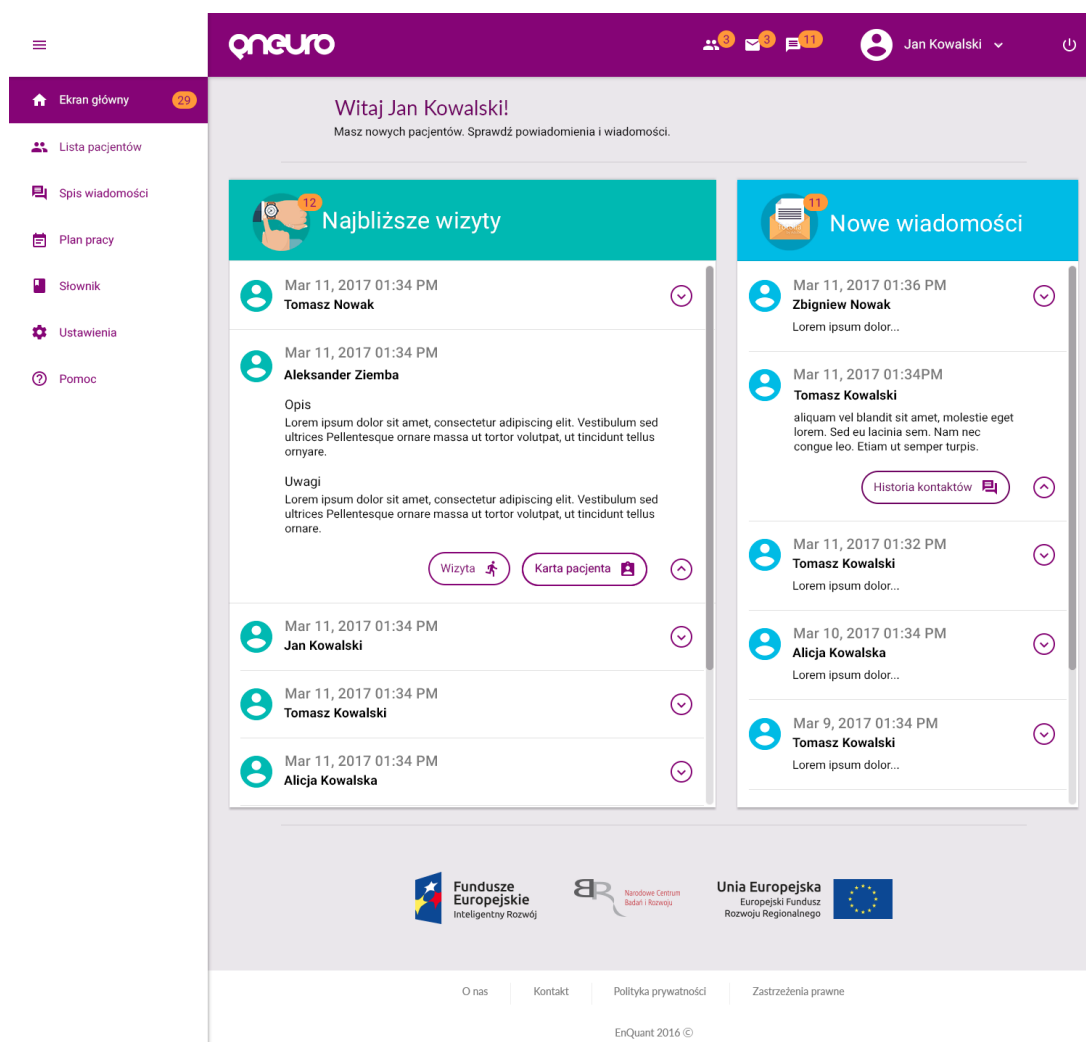
Zakres prac w ramach integracji aplikacji homeEEG obejmuje:

1. analizę, projekt i dostarczenie specyfikacji API mEngine zgodnie z załączoną specyfikacją
2. projekt i implementacja aplikacji mobilnej zintegrowanych z API mEngine oraz homeEEG/VEEG dla systemów Android i iOS wraz z udostępnieniem tych aplikacji w Google Play oraz Apple App Store.

4.4. Wspólne wymagania dla aplikacji

4.4.1. Wygląd

1. Szaty graficzne wraz z kolorystyką aplikacji mobilnych oraz webowych powinny być zgodne z istniejącymi projektami wybranych ekranów.



Rys. 2 . Widok ekranu głównego dla profilu (lekarz) (szerokość - 1500 px).

2. Aplikacje mobilne będą dostępne zarówno na telefonie jak i na tablecie. Zależnie od rozdzielczości ekranu urządzenia, aplikacja mobilna powinna dostosować widok i skalowanie, aby zachować najlepszą ergonomię obsługi.

4.4.2. Dokumentacja

Do każdej z aplikacji zostanie dołączona dokumentacja w języku polskim.

Na dokumentację składać się będą:

1. dokumentacja techniczna - opis funkcjonowania aplikacji, zastosowanych rozwiązań i algorytmów
2. dokumentacja użytkownika - opis dla użytkownika korzystającego z aplikacji mobilnej.

4.4.3. Bezpieczeństwo

1. Komunikacja aplikacji z częścią serwerową (backend środowiska QNeuro) przez sieć Internet będzie szyfrowana.
2. Przed przesłaniem danych, zarówno aplikacja jak i serwer muszą wzajemnie potwierdzić swoją tożsamość zanim przesłane zostaną informacje uwierzytelniające użytkownika

5. Formaty danych i API

5.1. Przykładowe komendy i format plików rejestratora homeEEG

Format pliku danych homeEEG

Podstawowym formatem zapisu danych będzie zapis binarny do pliku. Każdy plik binarny będzie posiadał swój opis w dodatkowym pliku xml. Plik ten będzie zawierał dodatkowe dane: nazwę/id urządzenia, liczbę kanałów (12 lub 200), częstotliwość próbkowania oraz datę badania

Na etapie budowy prototypu dodatkowo plik z danymi zapisywany będzie w formacie csv/txt na karcie SD urządzenia. Nazwa pliku zawierać będzie nazwę/id urządzenia, częstotliwość próbkowania oraz datę badania. Dane zawarte w pliku będą opisane w kolumnach tak, by móc taki plik zaimportować do arkusza kalkulacyjnego. Tytuły kolumn pozwolą też określić czy rejestracja sygnału EEG dotyczyła 12 czy 20 kanałów.

t	e1	...	e20	o1-o3	o4-o6	o7-o9
uint32, ms	int32, raw	int32, raw	int32, raw	uint16, m/s ²	uint16, Radians	uint16, μ T
Znacznik czasu.	Dane z elektrod. Jeśli elektrod jest 12, to pozostałe 8 jest pomijane.			Dane z czujnika orientacji (9 osi).		

Typ danych dla kanałów przetworników (e1 - e20) - zapis binarny w kodzie int32. W celu przeskalowania wartości binarnych na wartość sygnału wejściowego wyrażonego w mV w pliku .xml zawierającym opis badania zostaną dołączone dodatkowe dane przetwornika: Vref, GAINx - wzmacnienie dla kanału x.

Dane z czujnika ruchu są o rozdzielczości 16-bitowej i po skalibrowaniu zwracane w wartościach wielkości mierzonej.

Pliki danych oraz pliki zdarzeń dla każdego badania będą umieszczane w niezależnym folderze. Nazwa folderu to data i godzina.

Format pliku zdarzeń homeEEG

Oprócz pliku danych zostanie utworzony plik z dziennika zdarzeń. W przypadku, gdy podczas badania wystąpią jakieś błędy, np.: utrata kontaktu elektrody ze skórą w dzienniku zdarzeń zostanie takie zdarzenie zapisane. Nazwa pliku będzie zawierać nazwę/id urządzenia, datę badania oraz kod mówiący o tym, że jest to plik z błędami np "errors" czy "errorlog". Struktura pliku będzie zbliżona do pliku z zapisem badania z tym, że będzie posiadać kolumnę komentarza. Jako komentarz będzie kod błędu lub jego opis.



t	e1	...	e20	o1	...	o9	kod zdarzenia	komentarz
Znacznik czasu.	Dane z elektrod. Jeśli elektrod jest 12, to pozostałe 8 jest pomijane.			Dane z czujnika orientacji (9 osi).			Numeryczny kod błędu, zgodny ze słownikiem lub kod zdarzenia.	Komentarz lub parametry do kodu błędu

Lista kodów błędów lub zdarzeń w homeEEG:

Kod błędu/ kod zdarzenia	Komentarz
0x1000	połączenie
0x1001	rozłączenie
0x2xxx	błąd elektrody, gdzie xxx to maska bitowa wskazująca numer elektrody, np 0x2001 błąd na elektrodzie 1, 0x2003 błąd na elektrodach 1 i 2.
0x300x	status wciskania przycisków na urządzeniu podczas badania np.: START, gdzie x to maska bitowa wciśniętych przycisków
0x4000	komenda przesłana do urządzenia, treść komendy w komentarzu do zdarzenia
0x50xx	poziom baterii, gdzie xx to wysokość poziomu
0x60xx	RSSI, gdzie xx to wartość RSSI

Lista ta zostanie rozwinięta w trakcie dalszej implementacji.

Komunikacja przez REST

Zapytania z urządzenia EEG do komputera VEEG będą realizowane za pomocą zapytań HTTP/REST.

Nagłówki

JSON	"Content-Type": "application/json"
------	------------------------------------

Rozpoczęcie nagrania

GET /api/start/

parametry:

- rec_id - identyfikator nagrania, może nim być nazwa pliku lub inny ciąg znaków,
- timestamp - aktualny znacznik czasu na urządzeniu w chwili nadania zapytania. Znacznik ten będzie się różnił od znacznika w nagraniu, gdyż nagranie rozpocznie się dopiero po otrzymaniu pozytywnej odpowiedzi od VEEG

Odpowiedź:

200 – OK

400 – błędny znacznik czasu

Zakończenie nagrania

GET /api/stop/

parametry

- rec_id - identyfikator nagrania zgodny z nadanym podczas rozpoczęcia rejestracji EEG,
- url_data - adres do pobrania pliku z danymi zarejestrowanymi podczas nagrania, w formacie tftp://<ip>:<port>/<nazwa_pliku>
- url_log - adres do pobrania pliku z dziennikiem zarejestrowanym podczas nagrania, w formacie tftp://<ip>:<port>/<nazwa_pliku>

Odpowiedź:

200 – OK

400 – błędny adres

Komunikacja przez RFCOMM

Przewiduje się dwie możliwości połączenia do interfejsu RFCOMM - USB lub Bluetooth:

1. W trybie pracy urządzenia homeEEG z aplikacją mobilną komunikacja odbywać się z wykorzystaniem modułu Bluetooth.
2. W trybie pracy urządzenia homeEEG ze stanowiskiem VEEG komunikacja komunikacja odbywać się z wykorzystaniem portu USB.

Terminal / przyciski wysyłające dane jak z terminala. Polecenia z poniższych dwóch tabelek wydawane są przez komputer, interpretowane przez urządzenie, które na nie odpowiada wysyłając tylko żądane dane.

Komenda	Wynik
get id	Id urządzenia w formie tekstowej
get time	Znacznik czasu
get ip	Pobierz IP urządzenia
get status	Zwróć status urządzenia (połączenie, stan elektrod, RSSI, poziom baterii)

Wysyłanie danych do urządzenia

Komenda	Znaczenie
set time <i>timestamp</i>	Ustawia znacznik czasu
set ip <i>ip</i>	Ustawia statyczny adres IP urządzenia
set ssid <i>ssid</i>	Ustawia SSID
set params <i>params</i>	Ustawia parametry np. szyfrowania
set password <i>pass</i>	Ustawia hasło
set server <i>url</i>	Ustawia adres serwera HTTP/REST, do którego urządzenie ma zgłaszać zmiany stanów start/stop
start	Inicjuje rozpoczęcie rejestracji EEG
stop	Inicjuje zakończenie rejestracji EEG

5.2. Przykładowa specyfikacja API mEngine

5.2.1. Moduł Lekarza

Usługa bezpieczeństwa [mEngine - security]

1. Autoryzacja użytkownika

opis: Metoda pozwalająca na zalogowanie użytkownika do systemu

metoda: POST

url: /security/auth/login

parametry:

- [String] login

- [String] password

odpowiedź:

typ: JSON

wartość:

- [String] accessToken
- [String] refreshToken
- [Object] userDto
 - [String] login;
 - [String] sex – dopuszczalne wartości 'MALE', 'FEMALE';
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
 - [String] birthDate;
 - [String] email;
 - [String] phoneNumber;
 - [Array] profiles - dopuszczalne wartości 'DOCTOR', 'PATIENT';
 - [Array] roles
 - [String] accountStatus – dopuszczalne wartości 'NEW', 'CONFIRMED', 'ACTIVE'

2. Odświeżenie tokenu

opis: metoda pozwalająca na odświeżenie tokenu bezpieczeństwa

metoda: POST

url: /security/auth/token

parametry nagłówka: Authorization – zawierająca wartość refreshToken

odpowiedź:

typ: JSON

wartość:

- [String] accessToken

- [String] refreshToken
- [Object] userDto
 - [String] login;
 - [String] sex – dopuszczalne wartości 'MALE', 'FEMALE';
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
 - [String – ISO date format] birthDate;
 - [String] email;
 - [String] phoneNumber;
 - [Array] profiles - dopuszczalne wartości 'DOCTOR', 'PATIENT';
 - [Array] roles
 - [String] accountStatus – dopuszczalne wartości 'NEW', 'CONFIRMED', 'ACTIVE'

3. Pobranie informacji o zalogowanym użytkowniku

opis: metoda pozwalająca na pogranie danych o aktualnie zalogowanym użytkowniku

metoda: GET

url: /security/user/logged

odpowiedź:

typ: JSON

wartość:

- [Object] userDto
 - [String] login;
 - [String] sex – dopuszczalne wartości 'MALE', 'FEMALE';
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
 - [String – ISO date format] birthDate;

- [String] email;
- [String] phoneNumber;
- [Array] profiles - dopuszczalne wartości 'DOCTOR', 'PATIENT';
- [Array] roles

[String] accountStatus – dopuszczalne wartości 'NEW', 'CONFIRMED', 'ACTIVE'

4. Pobranie informacji o użytkowniku

opis: metoda pozwalająca na pogranie danych o aktualnie zalogowanym użytkowniku

metoda: GET

url: /security/user/{login}

parametr:

- [String] login

odpowiedź:

typ: JSON

wartość:

- [Object] userDto
 - [String] login;
 - [String] sex – dopuszczalne wartości 'MALE', 'FEMALE';
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
 - [String – ISO date format] birthDate;
 - [String] email;
 - [String] phoneNumber;
 - [Array] profiles - dopuszczalne wartości 'DOCTOR', 'PATIENT';
 - [Array] roles
 - [String] accountStatus – dopuszczalne wartości 'NEW', 'CONFIRMED', 'ACTIVE'

Usługa komunikacji [mEngine - communication]

5. Wysłanie wiadomości chat

opis: metoda pozwalająca wysłanie wiadomości typu chat

metoda: POST

url: /communication/chat/send

parametr body:

- [JSON] messageBody
 - [String] @NotNull toUserLogin;
 - [String] toUserFirstName;
 - [String] toUserSecondName;
 - [String] toUserLastName;
 - [String] @NotNull content;
 - [String – ISO time format] @NotNull requestTime;
 - [Long] parentMessageId;

odpowiedź:

typ: JSON (body)

wartość:

- [Long] id;
- [Long] severity;
- [UserDto] fromUser;
 - [String] login;
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
- [UserDto] toUser;
- [String - ISO time format] timeCreated;
- [String - ISO time format] timeSent;
- [String] content;

- [String] type [EMAIL, SMS, SYS, CHAT, EMERGENCY]

6. Wysłanie wiadomości email

opis: metoda pozwalająca wysłanie wiadomości typu email

metoda: POST

url: /communication/email/send

parametr body:

- [JSON] messageBody
 - [String] @NotNull toUserLogin;
 - [String] toUserFirstName;
 - [String] toUserSecondName;
 - [String] toUserLastName;
 - [String] @NotNull subject;
 - [String] @NotNull content;
 - [String – ISO time format] @NotNull requestTime;

odpowiedź:

typ: HTTP status

7. Pobranie historii komunikacji między użytkownikami

opis: metoda pozwalająca pobranie historii komunikacji między dwoma użytkownikami

metoda: POST

url: /communication/message/user/{firstUserLogin}/user/{secondUserLogin}

parametry:

- [String] firstUserLogin
 - [String] secondUserLogin

odpowiedź:

typ: JSON (body)

wartość: Array[Object]

- [Long] id;

- [Long] severity;
- [UserDto] fromUser;
 - [String] login;
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
- [UserDto] toUser;
- [String - ISO time format] timeCreated;
- [String - ISO time format] timeSent;
- [String] content;
- [String] type [EMAIL, SMS, SYS, CHAT, EMERGENCY]

8. Pobranie historii komunikacji między użytkownikami dla zadanego okresu.

opis: metoda pozwalająca pobrać historię komunikacji między dwoma użytkownikami dla zadanego okresu.

metoda: POST

url: /communication/mes/message/user/{firstUserLogin}/user/{secondUserLogin}/from/{fromDateString}/till/{tillDateString}

parametry:

- [String] firstUserLogin
- [String] secondUserLogin
- [String] fromDateString
- [String] tillDateString

odpowiedź:

typ: JSON (body)

wartość: Array[Object]

- [Long] id;
 - [Long] severity;
 - [UserDto] fromUser;

- [String] login;
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
- [UserDto] toUser;
- [String - ISO time format] timeCreated;
- [String - ISO time format] timeSent;
- [String] content;
- [String] type [EMAIL, SMS, SYS, CHAT, EMERGENCY]

Usługa danych plikowych

9. Pobieranie zdjęcia profilowego użytkownika systemu

metoda: GET

url: /user/{login}/avatar

Usługa dokumentacji medycznych

Informacje organizacyjne lekarza (harmonogram wizyt i zdarzeń medycznych)

10. Pobranie danych wizyty

metoda: GET

url: /visit/{visitId}

11. Pobranie listy wizyt lekarza dla wybranego dnia

metoda: GET

url: /visit//doctor/{doctorId}/date/{date}

12. Pobieranie listy wizyt dla zadanego okresu

metoda: GET

url: /visit//doctor/{doctorId}/from/{fromDate}/till/{tillDate}

13. Pobieranie listy wizyt pacjenta

metoda: GET

url: /visit/patient/{patientId}

14. Pobieranie listy zaplanowanych zdarzeń dla okresu

metoda: GET

url: /planned-event/doctor/{doctorId}/from/{fromDate}/till/{tillDate}

15. Pobieranie listy pacjentów lekarza

metoda: GET

url: /doctor/patient

Dokumentacja medyczna pacjenta

16. Pobieranie danych pacjenta

metoda: GET

url: /patient/{patientId}

17. Pobieranie skrótu historii choroby pacjenta

metoda: GET

url: /patient/{patientId}/medical-summary

18. Pobieranie danych o lekarzu prowadzącym pacjenta

metoda: GET

url: /patient/{patientId}/doctor

19. Pobieranie historii medycznej pacjenta

metoda: GET

url: /patient/{patientId}/history

20. Pobieranie historii medycznej pacjenta dla okresu

metoda: GET

url: /patient/{patientId}/history/from/{dateFrom}/till/{tillDate}

21. Pobieranie listy zdarzeń napadowych pacjenta dla okresu

metoda: GET

url: /patient/{patientId}/history/seizure/{dateFrom}/till/{tillDate}

22. Pobieranie słownika typów napadów pacjenta

metoda: GET

url: /patient/{patientId}/seizure-types

23. Pobieranie danych wyniku badania diagnostycznego (dane zlecenia wraz z opisem i komentarzami do badania)

metoda: GET

url: /medical-result/{medicalResultId}/patient/{patientId}

24. Pobranie zdarzeń medycznych pacjenta dla okresu

metoda: GET

url: /medical-event/patient/{patientId}

25. Pobranie listy terapii lekowych pacjenta

metoda: GET

url: /drug-therapy/patient/{patientId}

26. Pobranie listy aktualnych terapii lekowych pacjenta

metoda: GET

url: /drug-therapy/patient/{patientId}/active

Zarządzanie harmonogramem pracy lekarza

27. Pobieranie harmonogramu pracy lekarza w zadanym okresie

metoda: GET

url: /doctor/{doctorId}/harmonogram

28. Pobieranie wolnych terminów lekarza w zadanym okresie

metoda: GET

url: /doctor/{doctorId}/harmonogram/freeslot

29. Pobranie listy wizyt lekarza dla zadanego okresu

metoda: GET

url: /doctor/{doctorId}/visit

30. Pobranie szczegółów wizyty

metoda: GET

url: /doctor/{doctorId}/visit/{visitId}

31. Pobranie słownika dni wolnych od pracy

metoda: GET

url: /workplan/freeday

32. Pobranie listy urlopów lekarza

metoda: GET

url: /doctor/{doctorId}/workplan/vacation

33. Usunięcie urlopu lekarza

metoda: DELETE

url: /doctor/{doctorId}/workplan/vacation/{doctorVacationId}

34. Dodanie urlopu lekarza

metoda: PUT

url: /doctor/{doctorId}/workplan/vacation

35. Edycja urlopu lekarza

metoda: POST

url: /doctor/{doctorId}/workplan/vacation/{doctorVacationId}

36. Pobranie listy zmian w planie pracy lekarza dla okresu

metoda: GET

url: /doctor/{doctorId}/workplan/change

37. Usunięcie zmiany w planie pracy lekarza

metoda: DELETE

url: /doctor/{doctorId}/workplan/change/{changeId}

38. Dodanie zmiany w planie pracy lekarza

metoda: POST

url: doctor/{doctorId}/workplan/change

39. Pobranie planów pracy lekarza dla okresu



metoda: GET

url: /doctor/{doctorId}/workplan/

40. **Usunięcie planu pracy lekarza**

metoda: DELETE

url: /doctor/{doctorId}/workplan/base/{workPlanId}

41. **Dodanie planu pracy lekarza**

metoda: PUT

url: /doctor/{doctorId}/workplan/base

42. **Edycja planu pracy lekarza**

metoda: POST

url: /doctor/{doctorId}/workplan/base/{workPlanId}

Dane finansowe użytkownika

43. **Pobranie salda użytkownika**

metoda: GET

url: /user/balance

44. **Pobranie listy operacji finansowych użytkownika w wybranym okresie (z możliwością filtrowania)**

metoda: GET

url: /user/finacial-operation/from/{fromDate}/till/{tillDate}

Dane pomocnicze użytkownika

45. **Słowniki – Pobieranie listy adresów użytkownika systemu**

metoda: GET

url: /address

46. **Pobieranie danych lekarza dla zalogowanego użytkownika**

metoda: GET

url: /doctor/user/{userAccountId}

Słowniki systemowe

47. Słowniki - Pobranie listy oddziałów NFZ

metoda: GET

url: /dictionary/nfzbranches

48. Słowniki – Pobieranie listy typów badań diagnostycznych

metoda: GET

url: /dictionary/medical-result/types

49. Słowniki – Pobieranie słownika uprawnień pacjenta (dla recept)

metoda: GET

url: /dictionary/patientauthorisation

50. Słowniki – Pobieranie słownika podstawowych pozycji cennika systemowego

metoda: GET

url: /dictionary/pricelistitems

51. Słowniki – Pobieranie słownika specjalizacji lekarskich

metoda: GET

url: /dictionary/doctor-profession

52. Słowniki – Pobieranie słownika rozpoznań ICD10 (z filtrami)

metoda: GET

url: /dictionary/icd10-autocomplete

53. Słowniki – Pobieranie słownika procedur ICD9 (z filtrami)

metoda: GET

url: /dictionary/icd9-autocomplete

54. Słowniki – Pobieranie słownika leków (z filtrami)

metoda: GET

url: /dictionary/drug-autocomplete

55. Słowniki – Pobieranie słownika miast (z filtrami)

metoda: GET

url: /dictionary/city-autocomplete

56. Słowniki – Pobieranie słownika stowarzyszeń lekarskich

metoda: GET

url: /dictionary/society-autocomplete

57. Słowniki – Pobieranie słownika efektów ubocznych (dla napadów padaczkowych)

metoda: GET

url: /dictionary/seizure-side-effect

58. Słowniki – Pobieranie słownika wyzwalaczy napadów padaczkowych

metoda: GET

url: /dictionary/seizure-trigger

59. Słowniki – Pobieranie słownika zdarzeń niepożądanych

metoda: GET

url: /dictionary/undesirable-effect

60. Słowniki – Pobieranie słownika typów infekcji

metoda: GET

url: /dictionary/infection

61. Pobranie konsultacji powiązanej z wizytą pacjenta

metoda: GET

url: /consultation//visit/{visitId}

5.2.2. Moduł Pacjenta

Usługa bezpieczeństwa [mEngine – security]

1. Autoryzacja użytkownika

opis: Metoda pozwalająca na zalogowanie użytkownika do systemu

metoda: POST

url: /security/auth/login

parametry:

- [String] login

- [String] password

odpowiedź:

typ: JSON

wartość:

- [String] accessToken
- [String] refreshToken
- [Object] userDto
 - [String] login;
 - [String] sex – dopuszczalne wartości 'MALE', 'FEMALE';
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
 - [String] birthDate;
 - [String] email;
 - [String] phoneNumber;
 - [Array] profiles - dopuszczalne wartości 'DOCTOR', 'PATIENT';
 - [Array] roles
 - [String] accountStatus – dopuszczalne wartości 'NEW', 'CONFIRMED', 'ACTIVE'

2. Odświeżenie tokenu

opis: metoda pozwalająca na odświeżenie tokenu bezpieczeństwa

metoda: POST

url: /security/auth/token

parametry nagłówka: Authorization – zawierająca wartość refreshToken

odpowiedź:

typ: JSON

wartość:

- [String] accessToken

- [String] refreshToken
- [Object] userDto
 - [String] login;
 - [String] sex – dopuszczalne wartości 'MALE', 'FEMALE';
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
 - [String – ISO date format] birthDate;
 - [String] email;
 - [String] phoneNumber;
 - [Array] profiles - dopuszczalne wartości 'DOCTOR', 'PATIENT';
 - [Array] roles
 - [String] accountStatus – dopuszczalne wartości 'NEW', 'CONFIRMED', 'ACTIVE'

3. Pobranie informacji o zalogowanym użytkowniku

opis: metoda pozwalająca na pogrnięcie danych o aktualnie zalogowanym użytkowniku

metoda: GET

url: /security/user/logged

odpowiedź:

typ: JSON

wartość:

- [Object] userDto
 - [String] login;
 - [String] sex – dopuszczalne wartości 'MALE', 'FEMALE';
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
 - [String – ISO date format] birthDate;

- [String] email;
- [String] phoneNumber;
- [Array] profiles - dopuszczalne wartości 'DOCTOR', 'PATIENT';
- [Array] roles
- [String] accountStatus – dopuszczalne wartości 'NEW', 'CONFIRMED', 'ACTIVE'

4. Pobranie informacji o użytkowniku

opis: metoda pozwalająca na pogrnięcie danych o aktualnie zalogowanym użytkowniku

metoda: GET

url: /security/user/{login}

parametr:

- [String] login

odpowiedź:

typ: JSON

wartość:

- [Object] userDto
 - [String] login;
 - [String] sex – dopuszczalne wartości 'MALE', 'FEMALE';
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
 - [String – ISO date format] birthDate;
 - [String] email;
 - [String] phoneNumber;
 - [Array] profiles - dopuszczalne wartości 'DOCTOR', 'PATIENT';
 - [Array] roles
 - [String] accountStatus – dopuszczalne wartości 'NEW', 'CONFIRMED', 'ACTIVE'

Usługa komunikacji [mEngine - communication]

5. Wysłanie wiadomości chat

opis: metoda pozwalająca wysłanie wiadomości typu chat

metoda: POST

url: /communication/chat/send

parametr body:

- [JSON] messageBody
 - [String] @NotNull toUserLogin;
 - [String] toUserFirstName;
 - [String] toUserSecondName;
 - [String] toUserLastName;
 - [String] @NotNull content;
 - [String – ISO time format] @NotNull requestTime;
 - [Long] parentMessageId;

odpowiedź:

typ: JSON (body)

wartość:

- [Long] id;
- [Long] severity;
- [UserDto] fromUser;
 - [String] login;
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
- [UserDto] toUser;
- [String - ISO time format] timeCreated;
- [String - ISO time format] timeSent;
- [String] content;

- [String] type [EMAIL, SMS, SYS, CHAT, EMERGENCY]

6. Wysłanie wiadomości email

opis: metoda pozwalająca wysłanie wiadomości typu email

metoda: POST

url: /communication/email/send

parametr body:

- [JSON] messageBody
 - [String] @NotNull toUserLogin;
 - [String] toUserFirstName;
 - [String] toUserSecondName;
 - [String] toUserLastName;
 - [String] @NotNull subject;
 - [String] @NotNull content;
 - [String – ISO time format] @NotNull requestTime;

odpowiedź:

typ: HTTP status

7. Pobranie historii komunikacji między użytkownikami

opis: metoda pozwalająca pobranie historii komunikacji między dwoma użytkownikami

metoda: POST

url: /communication/message/user/{firstUserLogin}/user/{secondUserLogin}

parametry:

- [String] firstUserLogin
 - [String] secondUserLogin

odpowiedź:

typ: JSON (body)

wartość: Array[Object]

- [Long] id;

- [Long] severity;
- [UserDto] fromUser;
 - [String] login;
 - [String] firstName;
 - [String] secondName;
 - [String] lastName;
- [UserDto] toUser;
- [String - ISO time format] timeCreated;
- [String - ISO time format] timeSent;
- [String] content;
- [String] type [EMAIL, SMS, SYS, CHAT, EMERGENCY]

8. Pobranie historii komunikacji między użytkownikami dla zadanego okresu.

opis: metoda pozwalająca pobranie historii komunikacji między dwoma użytkownikami dla zadanego okresu.

metoda: POST

url: /communication/mes/message/user/{firstUserLogin}/user/{secondUserLogin}/from/{fromDateString}/till/{tillDateString}

parametry:

- [String] firstUserLogin
- [String] secondUserLogin
- [String] fromDateString
- [String] tillDateString

odpowiedź:

typ: JSON (body)

wartość: Array[Object]

- [Long] id;
- [Long] severity;
- [UserDto] fromUser;
 - [String] login;

- [String] firstName;
- [String] secondName;
- [String] lastName;
- [UserDto] toUser;
- [String - ISO time format] timeCreated;
- [String - ISO time format] timeSent;
- [String] content;
- [String] type [EMAIL, SMS, SYS, CHAT, EMERGENCY]

Usługa danych plikowych

9. Pobieranie zdjęcia profilowego użytkownika systemu

metoda: GET

url: /user/{login}/avatar

Usługa dokumentacji medycznych

Obsługa wizyt

10. Pobranie danych wizyty

metoda: GET

url: /visit/{visitId}

11. Pobieranie listy wizyt pacjenta

metoda: GET

url: /visit/patient/{patientId}

12. Pobieranie listy wizyt pacjenta w okresie

metoda: GET

url: /visit//patient/{patientId}/from/{fromDate}/till/{tillDate}

Obsługa zaplanowanych zdarzeń

13. Pobieranie listy zaplanowanych zdarzeń dla okresu

metoda: GET

url: /planned-event/patient/{patientId}/from/{fromDate}/till/{tillDate}

Dokumentacja medyczna pacjenta

14. Pobieranie danych pacjenta

metoda: GET

url: /patient/{patientId}

15. Pobieranie skrótu historii choroby pacjenta

metoda: GET

url: /patient/{patientId}/medical-summary

16. Pobieranie danych o lekarzu prowadzącym pacjenta

metoda: GET

url: /patient/{patientId}/doctor

17. Pobieranie historii medycznej pacjenta

metoda: GET

url: /patient/{patientId}/history

18. Pobieranie historii medycznej pacjenta dla okresu

metoda: GET

url: /patient/{patientId}/history/from/{dateFrom}/till/{tillDate}

19. Pobieranie listy zdarzeń napadowych pacjenta dla okresu

metoda: GET

url: /patient/{patientId}/history/seizure/{dateFrom}/till/{tillDate}

20. Pobieranie danych pacjenta dla zalogowanego użytkownika

metoda: GET

url: /patient/user/{userAccountId}

21. Pobieranie słownika typów napadów pacjenta

metoda: GET

url: /patient/{patientId}/seizure-types

22. Pobieranie danych wyniku badania diagnostycznego (dane zlecenia wraz z opisem i komentarzami do badania)

metoda: GET

url: /medical-result/{medicalResultId}/patient/{patientId}

23. Dodanie nowego zdarzenia medycznego (napady padaczkowe, zdarzenia z kalendarza napadów)

metoda: PUT

url: /medical-event/

24. Edycja istniejącego zdarzenia medycznego

metoda: POST

url: /medical-event/

25. Usunięcie zdarzenia medycznego

metoda: DELETE

url: /medical-event/{medicalEventId}

26. Pobranie zderzeń medycznych pacjenta dla okresu

metoda: GET

url: /medical-event/patient/{patientId}

27. Pobranie szczegółów wizyty

metoda: GET

url: /doctor/{doctorId}/visit/{visitId}

28. Pobranie listy terapii lekowych pacjenta

metoda: GET

url: /drug-therapy/patient/{patientId}

29. Pobranie listy aktualnych terapii lekowych pacjenta

metoda: GET

url: /drug-therapy/patient/{patientId}/active

30. Dodanie informacji o przyjęciu leku

metoda: POST



url: /drug-therapy/usage/patient/{patientId}

31. Pobranie konsultacji powiązanej z wizytą pacjenta

metoda: GET

url: /consultation//visit/{visitId}

Informacje o lekarzach

32. Pobieranie cennika lekarza

metoda: GET

url: /patient/doctor-pricelist/{doctorId}

33. Pobranie listy lekarzy (z możliwością filtrowania)

metoda: GET

url: /doctor

Dane pomocnicze użytkownika systemu

34. Słowniki – Pobieranie listy adresów użytkownika systemu

metoda: GET

url: /address

Dane finansowe użytkownika

35. Pobranie salda użytkownika

metoda: GET

url: /user/balance

36. Pobranie listy operacji finansowych użytkownika w wybranym okresie (z możliwością filtrowania)

metoda: GET

url: /user/finacial-operation/from/{fromDate}/till/{tillDate}

Słowniki systemowe

37. Słowniki - Pobranie listy oddziałów NFZ

metoda: GET

url: /dictionary/nfzbranches

38. Słowniki – Pobieranie listy typów badań diagnostycznych

metoda: GET

url: /dictionary/medical-result/types

39. Słowniki – Pobieranie słownika uprawnień pacjenta (dla recept)

metoda: GET

url: /dictionary/patientauthorisation

40. Słowniki – Pobieranie słownika podstawowych pozycji cennika systemowego

metoda: GET

url: /dictionary/pricelistitems

41. Słowniki – Pobieranie słownika specjalizacji lekarskich

metoda: GET

url: /dictionary/doctor-profession

42. Słowniki – Pobieranie słownika rozpoznań ICD10 (z filtrami)

metoda: GET

url: /dictionary/icd10-autocomplete

43. Słowniki – Pobieranie słownika procedur ICD9 (z filtrami)

metoda: GET

url: /dictionary/icd9-autocomplete

44. Słowniki – Pobieranie słownika leków (z filtrami)

metoda: GET

url: /dictionary/drug-autocomplete

45. Słowniki – Pobieranie słownika miast (z filtrami)

metoda: GET

url: /dictionary/city-autocomplete

46. Słowniki – Pobieranie słownika stowarzyszeń lekarskich

metoda: GET

url: /dictionary/society-autocomplete

47. Słowniki – Pobieranie słownika efektów ubocznych (dla napadów padaczkowych)

metoda: GET

url: /dictionary/seizure-side-effect

48. Słowniki – Pobieranie słownika wyzwalaczy napadów padaczkowych

metoda: GET

url: /dictionary/seizure-trigger

49. Słowniki – Pobieranie słownika zdarzeń niepożądanych

metoda: GET

url: /dictionary/undesirable-effect

50. Słowniki – Pobieranie słownika typów infekcji

metoda: GET

url: /dictionary/infection

6. Wycena oraz harmonogram realizacji prac

Prosimy o oszacowanie ceny oraz czasu trwania realizacji prac dla każdego poszczególnego zakresu tj.:

1. Analiza i implementacja mEngine API w mobilnej oraz webowej aplikacji Pacjent
2. Analiza i implementacja mEngine API w mobilnej oraz webowej aplikacji Lekarz
3. Analiza i implementacja mEngine API oraz implementacja API urządzenia homeEEG/VEEG w aplikacji mobilnej homeEEG

Opcjonalnie - Oferent może dany zakres prac rozwinąć o określenie kamieni milowych.

7. Forma odpowiedzi na niniejsze RFI

Odpowiedź powinna zawierać:

- (a) pełną nazwę Oferenta, NIP, adres lub siedzibę, numer telefonu do osoby kontaktowej;
- (b) datę sporządzenia odpowiedzi;
- (c) wycenę przedmiotu RFI – Oferent zobowiązany jest do zaprezentowania kwoty netto w PLN niezbędną do realizacji przedmiotu zapytania dla każdego z obszarów prac:
 - Analiza i implementacja mEngine API w mobilnej oraz webowej aplikacji Pacjent
 - Analiza i implementacja mEngine API w mobilnej oraz webowej aplikacji Lekarz
 - Analiza i implementacja mEngine API oraz implementacja API urządzenia homeEEG/VEEG w aplikacji mobilnej homeEEG
- (d) Harmonogram projektowy - określenie czasu realizacji poszczególnych prac obejmujących zakresy określone w pkt 6.
- (e) aktualny odpis z właściwego rejestru lub z centralnej ewidencji i informacji o działalności gospodarczej;
- (f) oświadczenie o braku powiązań osobowych i kapitałowych z Zamawiającym (wzór oświadczenia stanowi **Załącznik nr 1** do niniejszego zapytania ofertowego);
- (g) deklarację o przekazaniu pełnej dokumentacji technicznej, kodów źródłowych oraz praw własności do przedmiotu zamówienia (wzór oświadczenia stanowi **Załącznik nr 2** do niniejszego zapytania ofertowego);
- (h) opis propozycji wykonania przedmiotu zapytania (konceptcja, zastosowana technologia etc.);
- (i) wykaz kluczowych komponentów (np. zewnętrzne licencje, oprogramowania)
- (j) opis głównych czynników ryzyka związanych z realizacją zamówienia;
- (k) pozostałe informacje, które Oferent uzna za istotne.

8. Termin składania odpowiedzi na RFI

Termin składania odpowiedzi upływa w dniu: 29.10.2018 r.

Odpowiedź powinna być przesłana za pośrednictwem poczty elektronicznej na adres: karolina.slowikowska@enquant.pl (w tytule maila prosimy umieścić informację: Odpowiedź na **RFI** - Zaimplementowanie API dla zewnętrznych systemów)

lub pocztą tradycyjną na adres **ENQuant sp. z o.o. ul. Marii Skłodowskiej-Curie 41, 87-100 Toruń**. Na kopercie prosimy umieścić dopisek: Odpowiedź na **RFI** - Zaimplementowanie API dla zewnętrznych systemów. Liczy się data i czas dostarczenia oferty na wskazany powyżej adres.

Wszelkie pytania należy przysyłać wyłącznie drogą elektroniczną na adres:
karolina.slowikowska@enquant.pl

Termin zadawania pytań upływa z dniem 23.10.2018 r.

Termin udzielania odpowiedzi do dnia 26.10.2018 r.

9. Spis załączników/oświadczeń wymaganych od Oferenta

- Załącznik nr 1 – Oświadczenie o braku powiązań osobowych i kapitałowych z Zamawiającym
- Załącznik nr 2 – Deklarację o przekazaniu pełnej dokumentacji technicznej, kodów źródłowych oraz praw własności do przedmiotu zamówienia

Załącznik nr 1 do RFI z dnia 19.10.2018

nr RFI-8.2-2018_Zaimplementowanie_API_dla_zew_systemow_v1.0

OŚWIADCZENIE O BRAKU POWIĄZAŃ OSOBOWYCH I KAPITAŁOWYCH Z ZAMAWIAJĄCYM

dotyczy: **RFI – Zaimplementowanie API dla zewnętrznych systemów** w ramach projektu pt.: „QNeuro – zintegrowany e-system wspierający diagnostykę i prowadzenie chorych na padaczkę” jest współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Inteligentny Rozwój lata 2014-2020, Projekt jest realizowany w ramach poddziałania 1.1.2 PO IR Prace B+R związane z wytworzeniem instalacji pilotażowej/demonstracyjnej w konkursie 2/1.1.2/2015 PO IR.

ZAMAWIAJĄCY:

ENQuant Sp. z o.o. ul. Marii Skłodowskiej-Curie 41; 87-100 Toruń, KRS 0000325333, Regon 340549671, NIP 9562248388

WYKONAWCA

Nazwa Oferenta	Adres	Dane firmowe
		NIP _____ REGON _____

Niniejszym oświadczam, że jako Dostawca (Oferent) nie pozostaję z Zamawiającym w powiązaniach osobowych i kapitałowych polegających w szczególności na:

- Uczestniczeniu w spółce jako wspólnik spółki cywilnej lub spółki osobowej;
- posiadaniu udziałów lub co najmniej 10% akcji;
- pełnieniu funkcji członka organu nadzorczego lub zarządzającego, prokurenta, pełnomocnika;
- zostawaniu w związku małżeńskim, w stosunku pokrewieństwa lub powinowactwa w linii prostej, pokrewieństwa lub powinowactwa drugiego stopnia w linii bocznej lub w stosunku przysposobienia, opieki lub kurateli;
- pokrewieństwa lub powinowactwa w linii bocznej do drugiego stopnia lub w stosunku przysposobienia, opieki lub kurateli.

Podpis(y):

Lp.	Data	Nazwisko i imię osoby (osób) uprawnionej(ych)	Podpis(y) osoby(osób) uprawnionej(ych)

Załącznik nr 2 do RFI z dnia 19.10.2018

nr RFI-8.2-2018_Zaimplementowanie_API_dla_zew_systemow_v1.0

DEKLARACJA O PRZEKAZANIU KODÓW ŹRÓDŁOWYCH ORAZ PRAW WŁASNOŚCI

dotyczy: **RFI – Zaimplementowanie API dla zewnętrznych systemów** w ramach realizacji projektu pt.: „QNeuro – zintegrowany e-system wspierający diagnostykę i prowadzenie chorych na padaczkę” współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Inteligentny Rozwój lata 2014-2020. Projekt jest realizowany w ramach poddziałania 1.1.2 PO IR Prace B+R związane z wytworzeniem instalacji pilotażowej/demonstracyjnej w konkursie 2/1.1.2/2015 PO IR.

ZAMAWIAJĄCY:

ENQuant Sp. z o.o. ul. Marii Skłodowskiej-Curie 41; 87-100 Toruń, KRS 0000325333, Regon 340549671, NIP 9562248388

OFERENT

Nazwa Oferenta	Adres	Dane firmowe

Niniejszym oświadczam, że jako Oferent w ramach składanej odpowiedzi na [RFI-8.2-2018_Zaimplementowanie_API_dla_zew_systemow_v1.0](#) z dnia **19.10.2018** r. deklaruje, iż w przypadku nawiązania współpracy z Zamawiającym i tym samym otrzymania zamówienia na wykonanie przedmiotowych prac, dostarczę pełną dokumentację techniczną, prześlę kody źródłowe oraz prześlę na rzecz Zamawiającego autorskie prawa majątkowe wraz z prawem wykonywania praw zależnych i osobistych do opracowanych utworów. Przeniesienie autorskich praw majątkowych do utworu obejmuje wszystkie pola eksploatacji, zgodnie z Ustawą o prawie autorskim i prawach pokrewnych z dnia 4 lutego 1994.

Podpis(y):

Lp.	Data	Nazwisko i imię osoby (osób) uprawnionej(ych)	Podpis(y) osoby(osób) uprawnionej(ych)