

## ZAŁĄCZNIK NR 1 DO SWZ

**OPIS PRZEDMIOTU ZAMÓWIENIA**

Przedmiotem zamówienia jest świadczenie na rzecz Zamawiającego usług programistycznych w wymiarze 640 h - roboczogodzin polegających na wsparciu modernizacji oprogramowania Centrum Infrastruktury Badawczej Indukowanej Sejsmiczności (CIBIS) dla Centrów Infrastruktury Badawczej – Indukowanej Sejsmiczności (CIBIS), Analitycznych Laboratoriów (CIBAL), Sejsmicznych Badań Litosfery (CIBSBL) oraz Obserwacji Geomagnetycznych i Magnetotellurycznych (CIBOGM).

Zamówienie realizowane będzie w ramach projektu "EPOS - System Obserwacji Płyty Europejskiej" współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Inteligentny Rozwój 2014–2020, Priorytet IV: Zwiększenie potencjału naukowo- badawczego, działanie 4.2: Rozwój nowoczesnej infrastruktury badawczej sektora nauki nr POIR.04.02.00-14-A003/16.

**I. Wstęp**

Oprogramowanie Centrum Infrastruktury Badawczej Indukowanej Sejsmiczności (CIBIS) zostało stworzone w ramach projektu pn. „Cyfrowa przestrzeń badawcza sejsmiczności indukowanej dla celów EPOS”, finansowanego w ramach Programu Operacyjnego Innowacyjna Gospodarka Działania 2.3 *Inwestycje związane z rozwojem infrastruktury informatycznej nauki* oraz projektu „EPOS - System Obserwacji Płyty Europejskiej” współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego w ramach Programu Operacyjnego Inteligentny Rozwój 2014–2020, Priorytet IV: Zwiększenie potencjału naukowo- badawczego, działanie 4.2: Rozwój nowoczesnej infrastruktury badawczej sektora nauki nr POIR.04.02.00-14-A003/16

**Jest ono obecnie wykorzystywane przez:**

- eNode Kraków (CIBIS3) do gromadzenia, opisywania metadanymi oraz udostępnienia danych z zakresu indukowanej sejsmiczności do platformy IS-EPOS (tcs-ah.epos.eu),
- Centrum Infrastruktury Badawczej Analitycznych Laboratoriów do gromadzenia, opisywania metadanymi danych z zakresu paleomagnetyzmu (CIBAL2) oraz magnetyzmu środowiskowego (CIBAL1) oraz udostępniania danych poprzez tworzone Platformę Paleomagnetyzmu i Platformę Magnetyzmu Środowiskowego,
- Centrum Infrastruktury Badawczej Sejsmicznych Badań Litosfery do gromadzenia, opisywania metadanymi rezultatów sejsmicznych badań litosfery oraz udostępniania danych poprzez tworzoną platformę sejsmicznych badań litosfery,



- Centrum Infrastruktury Obserwacji Geomagnetycznych i Magnetotellurycznych do gromadzenia, opisywania metadanymi obserwacji geomagnetycznych i magnetotellurycznych oraz udostępniania danych poprzez tworzoną platformę obserwacji geomagnetycznych i magnetotellurycznych.

**Na wykonaną usługę modernizacji oprogramowania Centrum Infrastruktury Badawczej Indukowanej Sejsmiczności (CIBIS), Wykonawca udziela gwarancji na wszystkie elementy przedmiotu zamówienia na okres co najmniej 12 miesięcy, na warunkach określonych w projektowanych postanowieniach umowy, stanowiących załącznik nr 2 do SWZ.**

## **II. Opis oprogramowania CIBIS i jego podstawowych funkcjonalności.**

**Oprogramowanie CIBIS składa się z następujących części:**

1. Panel zarządzania użytkownikami i uprawnieniami – w tej części oprogramowania możliwy jest podgląd ustawień użytkownika, grup użytkowników i ich uprawnień oraz zarządzania nimi.
2. Epizody – Moduł służy do grupowania folderów niezależnie od ich magazynu danych. Epizod jest jednym z elementów publikacji danych do publikacji w LDAP System oferuje możliwość definiowania wielu epizodów. Obiekty te mają charakter informacyjny i agregują dane zgromadzone w wielu katalogach danych (FilesDirectory). Do każdego epizodu można przypisać wartości na zasadzie pól klucz-wartość.
3. Magazyny Danych – Moduł służy do fizycznego grupowania folderów. Określa miejsce zapisu wszystkich folderów podpiętych pod dany magazyn danych. Jest to jedna z encji stworzona na potrzebę przechowywania danych - FilesStorage – obiekty tej klasy definiują przestrzeń fizyczną na dysku w obrębie której tworzone będą katalogi na pliki danych. Katalogi te reprezentowane będą w systemie jako obiekty klasy FilesDirectory
4. Foldery - Moduł służący do operacji na plikach (*File*) i folderach. Pozwala na tworzenie folderów głównych, do których można dodać podrzędne foldery i pliki. Podrzędna struktura folderów może być utworzona na podstawie schematu. Oprócz dodawania plików i folderów, możliwe jest też m.in. kopiowanie, pobieranie, pakowanie do archiwum, rozpakowywanie, zmiana nazwy, przenoszenie lub podgląd zawartości. Pliki są wersjonowane (*FileVersionInfo*). Jest to także jedna z encji stworzona na potrzebę przechowywania danych- *FilesDirectory* - instancja tej klasy reprezentuje główny katalog w którym system może tworzyć struktury podkatalogów oraz zapisywać pliki danych. Instancja klasy *FilesDirectory* nie może istnieć bez powiązania z instancją klasy FilesStorage. W przypadku folderów można zdefiniować alternatywne źródło danych – *DataSource* – Datasource PUT, SeisComP.

5. Schematy - Moduł służy do importowania schematów LDAP, które są wykorzystywane do tworzenia drzewa katalogów w folderach. Moduł pozwala również na określenie reguł metadanych Danych do publikacji. Podstawową klasą jest `cibisDataItem` która opisuje poszczególne node'y w strukturze drzewiastej danych, określone jako `STRUCTURAL` i dziedziczące po klasie `top`. System będzie pozwalał na dostarczenie schematów LDAP w postaci plików. Schemat po dostarczeniu do systemu poprzez stworzony w tym celu formularz zostanie przekonwertowany na format XML i w takiej postaci zostanie zapisany w bazie danych systemu CIBIS. Na potrzeby przechowywania schematów i ich wersjonowania zostaną wyodrębnione następujące klasy:

- `Scheme` – klasa reprezentująca schemat. Instancja tej klasy nie może istnieć bez powiązania do co najmniej jednego obiektu klasy `SchemeVersion`.
- `SchemeVersion` – klasa reprezentująca wersję schematu, instancja tej klasy nie może istnieć bez relacji do obiektu klasy `Scheme`.

System podczas wczytywania schematu pozwala użytkownikowi na wybór czy jest to nowy schemat czy tylko nowa wersja istniejącego już w systemie schematu. Jeśli użytkownik wczytuje do systemu nowy schemat system utworzy obiekt klasy `Scheme` oraz obiekt klasy `SchemeVersion` i powiąże je wzajemną relacją. Jeśli natomiast użytkownik wczytuje do systemu nową wersję istniejącego schematu, musi wybrać schemat (obiekt klasy `Scheme`) którego dotyczy wczytywana wersja. System następnie utworzy instancję klasy `SchemeVersion` i powiąże ją z wybranym przez użytkownika schematem (obiekt klasy `Scheme`).

Pliki reguł służą do wskazania w systemie, które metadane mogą być edytowalne oraz jakie wartości mogą przyjmować.

Schemat pliku reguł:

```
<rule>
<schemaRule objectClass="cibisDataItem">
<attribute name="name" />
<attribute name="itemType" />
<attribute name="path" />
<attribute name="episode" maxLength="10" isRequired="true"/>
<attribute name="text" />
<attribute name="coordinateSystem" enum="['LOCAL:POL001','WGS-84']" />
</schemaRule>
<dataTypeRule>
<item type="episode" required="episode" optional="text" />
<item type="directory" required="text" optional="coordinateSystem" />
<item type="file" required="text" optional="coordinateSystem" />
</dataTypeRule>
<requirementPolicy>
<requirement condition="latitude='12' and (episode='ESI-2' or episode='ESI-1')"
required="coordinateSystem" optional=""/>
</requirementPolicy>
</rule>
```

W pliku reguł wyróżniamy 3 bloki z danymi:

1). Blok schemaRule:

Zawiera atrybuty z objectClass używanej przez schemat, w bloku tym użytkownik określa kolejność pól wyświetlających się w formularzu edycji metadanej oraz pola które mają się w nim pojawić. Każdy atrybut może zawierać odpowiednie wartości służące do konfiguracji systemu:

- maxLength: określa maksymalną ilość znaków dla pola,
- minLength: określa minimalną ilość wymaganych znaków dla pola,
- maxValue: określa maksymalną wartość liczbową pola,
- minValue: określa minimalną wartość liczbową pola
- regexp: określa patern służący do walidacji pola
- enum: wartość numerowana dla pola określa wartości jakie mogą zostać wybrane dla danego pola.

2). Blok dataTypeRule:

Zawiera atrybuty określające pola wymagane oraz opcjonalne dla pól: epizodu, folderu oraz pliku.

3). Blok requirementPolicy:

Zawiera konfigurowalne atrybuty służące do określania, pojawienia się wymagalności pola lub jego opcjonalności w przypadku jego spełnienia. Język zawarty w zmiennej condition przybiera formę <nazwa\_pola>=<wartość\_wymagana> wartości mogą być łączone za pomocą warunków logicznych and oraz or. Można je również zagnieżdżać.

W pliku reguł pola name, path oraz objectType nie są edytowalne oraz nie mogą przybierać warunków logicznych.

6. Konfiguracje - Moduł służący do tworzenia list zadań, na podstawie istniejących elementów zadań. Elementem zadań jest konwerter. Lista zadań operuje na wskazanych plikach w sposób zależny od rodzaju jej elementów. By ją wykonać należy przypisać ją do folderu. Oprócz edycji oraz tworzenia, konfiguracje (listy zadań) można eksportować i importować.

7. Dane do publikacji Moduł służy do tworzenia metadanych na podstawie zaimportowanych schematów i reguł dla wybranych folderów. Metadane wykorzystywane są później przy publikacji danych do LDAP. Sekcja bazy danych odpowiada za przygotowanie danych i ich opisu metadaneowego w celu udostępnienia ich do Data Center. Użytkownik może utworzyć nową bazę danych przypisując ją do epizodu, folderu, schematu i jego wersji. Następnie przeglądając bazę danych system prezentuje użytkownikowi dane dostarczone przez użytkowników oraz wyniki ich konwersji. Do każdego obiektu typu plik i folder system udostępnia formularz pól metadanych wygenerowany na podstawie wersji schematu oraz pliku XML z ograniczeniami. Z poziomu tego formularza użytkownik może podać wartości metadanych wpisując je ręcznie lub zdefiniować reguły dla automatu odczytującego wartości. Ponadto dla każdego obiektu opisanego metadanymi system oferuje możliwość pobrania pliku XML metadanych, importu tego pliku z dysku oraz ponowne odczytanie wartości z tego pliku na dysku serwera.

Dla każdego obiektu opisanego metadanymi, lub całej bazy, system oferuje możliwość uruchomienia automatu odczytującego metadane. Pliki metadanych (zapisane w formacie XML) opisują każdy element danych (plik lub folder) i przechowywane są w tym samym miejscu w którym znajdują się opisywane przez nie dane. Nazwa tego pliku budowana jest zgodnie ze schematem:

{nazwa\_pliku\_lub\_folderu}.{wersja\_bazy}.{wersja\_schematu}.meta.xml.

W przypadku gdy baza danych jest poprawna system pozwala na jej publikację w serwerze LDAP. Publikowana struktura bazy danych wygląda następująco:

Schemat/Wersja

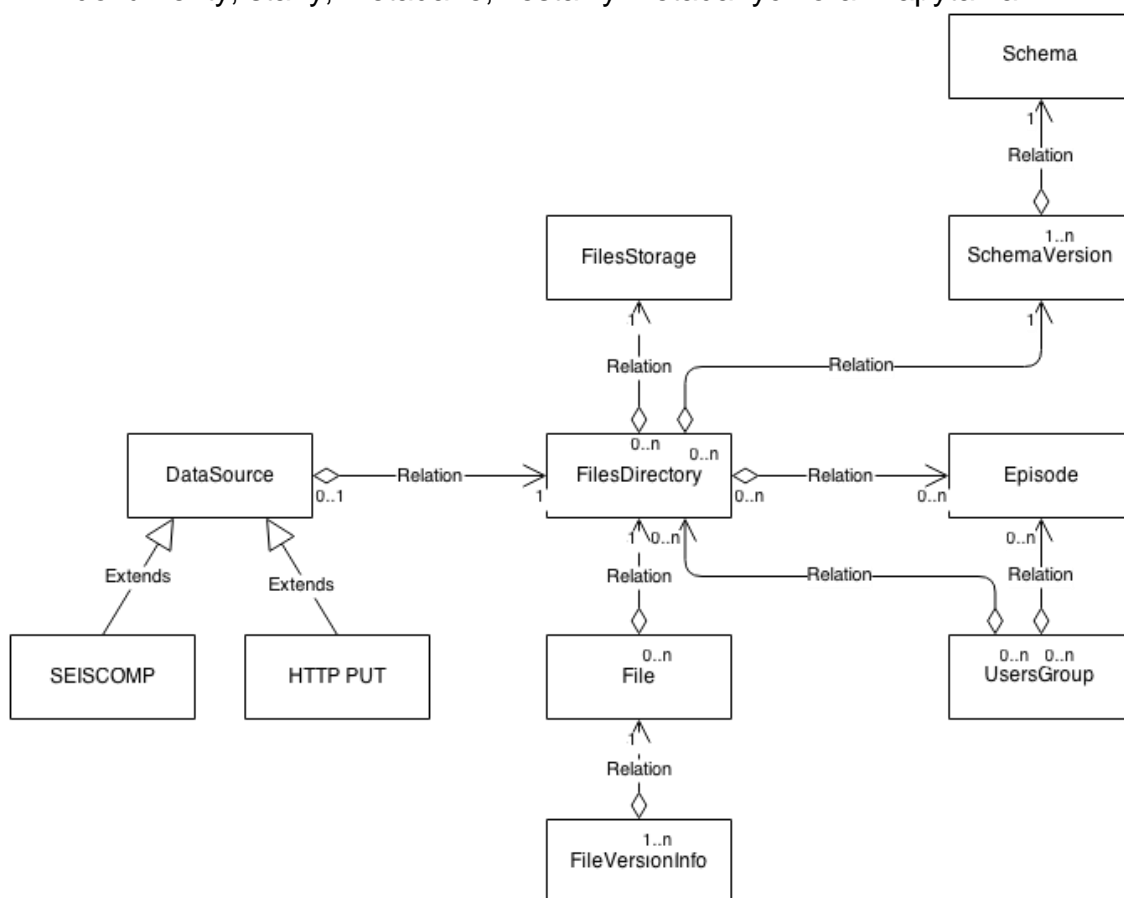
-- Epizod

---- Baza danych

----- [Struktura bazy danych]

Pliki metadanowe użyte w bazie danych udostępniane są przez LDAP.

8. Repozytorium - Moduł służący do wymiany dokumentów. Wszystkie jego części mają na celu jak najtrafniejsze opisanie pliku lub zestawu plików dokumentu, ułatwiając tym samym wyszukiwanie konkretnych dokumentów. Oprócz możliwości wyszukiwania, repozytorium określa jakie osoby mają dostęp do jakich zasobów. Repozytorium składa się z sekcji: kategorie, dokumenty, stany, metadane, zestawy metadanych oraz zapytania.



Rys 1 – schemat zależności pomiędzy głównymi modułami systemu.

## Wykaz licencji dostawców trzecich oprogramowania

Lp.	Moduł / Oprogramowanie	Licencja	Używane	Link do produktu
1	Python 2.7 + biblioteka standardowa	PSF <a href="https://docs.python.org/3/license.html">https://docs.python.org/3/license.html</a>	Całość projektu	<a href="http://www.python.org">www.python.org</a>
2	Django 1.7.1	BSD <a href="https://github.com/django/django/blob/master/LICENSE">https://github.com/django/django/blob/master/LICENSE</a>	Całość projektu	<a href="https://www.djangoproject.com/">https://www.djangoproject.com/</a>
3	Angular JS	MIT <a href="http://opensource.org/licenses/MIT">http://opensource.org/licenses/MIT</a>	Widok UI	<a href="https://angularjs.org/">https://angularjs.org/</a>
4	SB Admin 2 Bootstrap Admin Theme	APACHE <a href="http://www.apache.org/licenses/LICENSE-2.0">http://www.apache.org/licenses/LICENSE-2.0</a>	Widok UI	<a href="http://startbootstrap.com/template-overviews/sb-admin-2/">http://startbootstrap.com/template-overviews/sb-admin-2/</a>
5	Fabric	OWN LICENSE <a href="https://github.com/fabric/fabric/blob/master/LICENSE">https://github.com/fabric/fabric/blob/master/LICENSE</a>	Instalator	<a href="http://www.fabfile.org/en/latest/">http://www.fabfile.org/en/latest/</a>
6	Django Rest Framework	OWN LICENSE <a href="http://www.django-rest-framework.org/">http://www.django-rest-framework.org/</a>	Api REST	<a href="http://www.django-rest-framework.org/">http://www.django-rest-framework.org/</a>
7	Django-Countries	OWN LICENSE <a href="https://github.com/SmileyChris/django-countries/blob/master/LICENSE">https://github.com/SmileyChris/django-countries/blob/master/LICENSE</a>	Użytkownicy	<a href="https://github.com/SmileyChris/django-countries/">https://github.com/SmileyChris/django-countries/</a>
9	Django-Requestlogging	OWN LICENSE <a href="https://bitbucket.org/trustcentric/django-requestlogging/src/a391cb7a36a168550563058c95bb6269cc785c05/LICENSE?at=default">https://bitbucket.org/trustcentric/django-requestlogging/src/a391cb7a36a168550563058c95bb6269cc785c05/LICENSE?at=default</a>	Logowanie zdarzeń	<a href="https://bitbucket.org/trustcentric/django-requestlogging/">https://bitbucket.org/trustcentric/django-requestlogging/</a>
10	Django-Phonenumber-Field	OWN LICENSE <a href="https://github.com/stefanfoulis/django-phonenumber-field/blob/develop/LICENSE">https://github.com/stefanfoulis/django-phonenumber-field/blob/develop/LICENSE</a>	Użytkownicy	<a href="https://github.com/stefanfoulis/django-phonenumber-field">https://github.com/stefanfoulis/django-phonenumber-field</a>
11	Sphinx	BSD	Generowanie	<a href="http://sphinx-">http://sphinx-</a>



		<a href="https://github.com/django/django/blob/master/LICENSE">https://github.com/django/django/blob/master/LICENSE</a>	anie dokumen tacji	doc.org/
12	UI Bootstrap	MIT <a href="http://opensource.org/licenses/MIT">http://opensource.org/licenses/MIT</a>	Widok UI	<a href="http://angular-ui.github.io/bootstrap/">http://angular-ui.github.io/bootstrap/</a>
13	Angular-Gettext	MIT <a href="http://opensource.org/licenses/MIT">http://opensource.org/licenses/MIT</a>	Tłumacz enie	<a href="https://angular-gettext.rocketeer.be/">https://angular-gettext.rocketeer.be/</a>
14	Grunt	MIT <a href="http://opensource.org/licenses/MIT">http://opensource.org/licenses/MIT</a>	Automat yzacja generow ania js	<a href="http://gruntjs.com/">http://gruntjs.com/</a>

Opis środowiska serwerowego:

- Maszyna wirtualna VMware
  - system Ubuntu 18.04 LTS
  - 8 x vcpu
  - 16GB RAM
  - swap 4GB
  - dysk 512GB
  - Apache/2.4.29
  - PostgreSQL 10.15
  - OpenLdap 2.4.45

### III. Zakres i wymagania dotyczące realizacji przedmiotu umowy

1. Usługa wsparcia programistycznego w wymiarze 640h roboczogodzin polegającego na modernizacji oprogramowania Centrum Infrastruktury Badawczej Indukowanej Sejsmiczności (CIBIS) dla Centrów Infrastruktury Badawczej – Analitycznych Laboratoriów (CIBAL), Sejsmicznych Badań Litosfery (CIBSBL) oraz Obserwacji Geomagnetycznych i Magnetotellurycznych (CIBOGM).
  - 1) Zakres realizowanych prac będzie obejmował programowanie elementów aplikacji działających w środowisku serwerowym (Linux) i desktopowym (użytkownicy uzyskują dostęp do serwisu webowego systemu CIBIS poprzez przeglądarkę internetową, dlatego użyte technologie webowe powinny być obsługiwane przez każdą współczesną przeglądarkę).
  - 2) Prace będą realizowane w formie osobnych zleceń. Zakres prac w zleceniu obejmuje:
    - a) wykonanie elementów funkcjonalnych oprogramowania CIBIS;
    - b) opracowanie dokumentacji użytkowej;
    - c) przygotowanie instalatora zmodernizowanego oprogramowania CIBIS o zleczone elementy funkcjonalne – nowa wersja oprogramowania,

oraz przygotowanie plików aktualizujących działające oprogramowanie CIBIS (baza CIBIS3, CIBAL1, CIBAL2, CIBOGM, CIBSBL) do najnowszej wersji oprogramowania.

2. Wymagania:

- 1) Wykonawca przedstawi w terminie 6 dni roboczych od otrzymania zlecenia prac programistycznych, o którym mowa w pkt 1 ppkt 2) powyżej, harmonogram zawierający termin wykonania tych prac oraz ilość potrzebnych roboczogodzin do ich wykonania z wyszczególnieniem realizowanych podzadań. Ilość roboczogodzin wskazana w harmonogramie musi zostać określona przy uwzględnieniu konieczności zrealizowania wszystkich zadań opisanych w punkcie IV ppkt 1) i 2) w ramach określonych tam limitów roboczogodzin (łącznie do 640 roboczogodzin). Zamawiający ma prawo zgłosić uwagi do przedstawionego harmonogramu, w terminie do 4 dni roboczych od dnia jego otrzymania, które wykonawca jest zobowiązany uwzględnić w ciągu 4 dni roboczych. Prace mogą być rozpoczęte przez Wykonawcę tylko po akceptacji harmonogramu przez Zamawiającego;
- 2) Całość wykonanych prac, kodów, dokumentacja oraz prawa autorskie majątkowe wynikające z realizacji umowy przechodzą na Zamawiającego;
- 3) Końcowa dokumentacja użytkowa przekazywana będzie Zamawiającemu w formie elektronicznej, w formacie word;
- 4) Wykonawca ponosi pełną odpowiedzialność za wszelkie szkody powstałe w wyniku instalacji lub aktualizacji, wynikające z wad oprogramowania i zobowiązuje się do pokrycia wszystkich kosztów związanych z przywróceniem istniejącej bazy danych do stanu wyjściowego;
- 5) Wykonawca udziela co najmniej 12 miesięcznej gwarancji na wykonane oprogramowanie, a w przypadku stwierdzenia w okresie gwarancji wad wykonanego oprogramowania niezwłocznie przystąpi do usunięcia tych wad w terminie nieprzekraczającym 14 dni od daty zgłoszenia;
- 6) Osoby skierowane przez Wykonawcę do realizacji zamówienia muszą posiadać doświadczenie zawodowe z zakresu programowania interaktywnych serwisów internetowych oraz programowania obliczeń równoległych i kolejkovania zadań, instalacji i konfiguracji oprogramowania 'open software' w systemie Linux związanych z wykorzystaniem nw. technologii. Ze względu na to, że oprogramowanie CIBIS przetwarza produkty oprogramowania aplikacji naukowych, to pożądana jest znajomość Matlab, Octave, Fortran, c/c++, QGIDS, PostGIS, Spatial.

#### IV. Sposób realizacji zamówienia



Zamawiający będzie zlecał prace programistyczne wykonawcy w postaci zleceń, o których mowa w ust. III pkt. 1 ppkt 2) do momentu wyczerpania liczby 640 roboczogodzin programistycznych objętych przedmiotem zamówienia.

Wymiar prac realizowanych w ramach usługi wsparcia programistycznego stanowiącego przedmiot Umowy wynosi 640 roboczogodzin i obejmuje:

- 1) 300 roboczogodzin przeznaczonych na zrealizowanie 13 zaplanowanych następujących zadań:
  - 1.1) Możliwość personalizowania nazwy oprogramowania (z CIBIS na np. CIBAL, nie tylko adresu URL) (ogólny wygląd);
  - 1.2) Możliwość personalizowania pierwszej strony oprogramowania (nazwa, rysunki itp.) (ogólny wygląd);
  - 1.3) Możliwość zmiany nazwy "epizod" na inną (epizody);
  - 1.4) Możliwość dodawania nowej object class - z poziomu oprogramowania – (schematy);
  - 1.5) Operacje na plikach metadanych z poziomu przeglądarki na poziomie folderów: podgląd plików .metadata, dodawanie, kopiowanie, przenoszenie, pakowanie, rozpakowywanie (foldery);
  - 1.6) Przesyłanie dużych zestawów danych bezpośrednio z poziomu CIBIS (foldery);
  - 1.7) Powiązanie QC (Redmine) z oprogramowaniem CIBIS - możliwość dopisania i podglądu w folderze/bazie nr issue (np. GET /issues/2.xml) (foldery/dane do publikacji);
  - 1.8) Dodanie obsługi błędów w aplikacji (niedozwolone nazwy, opisy metadanych jeśli się nie zapisują, błąd wczytywania pliku rules, itp.), a także poprawa istniejącego opisu niektórych błędów (np. kopiowania bazy) (schematy/dane do publikacji);
  - 1.9) Integracja istniejących skryptów generujących metadane, tak aby generowanie metadanych dla wszystkich plików odbywało się z poziomu CIBIS (dane do publikacji);
  - 1.10) Obsługa głównych metadanych - możliwość czytania z pliku i pobrania pliku, tak jak dla pozostałych metadanych (dane do publikacji);
  - 1.11) Dodanie możliwości "zabijania" trwających zadań - publikacji i odczytywania metadanych z serwera (dane do publikacji);
  - 1.12) Rozwój języka pattern (dane do publikacji);
  - 1.13) Możliwość edycji wielokrotnych metadanych z poziomu epizodu (dane do publikacji);
- 2) 340 roboczogodzin, które Zamawiający przeznaczy na zrealizowanie dodatkowych minimum 10 zadań, których zakres zostanie wskazany na etapie realizacji Umowy w postaci dodatkowych minimum 10 Zleceń.

Zamawiający zapewnia niezwłoczne zlecenie kolejnych prac programistycznych, w odstępach czasu umożliwiających terminowe zrealizowanie przedmiotu zamówienia.

Każde zlecenie będzie wykonywane na podstawie przygotowanego harmonogramu zawierającego zakres wykonywanych czynności, jak i ilość roboczogodzin przewidzianych w tym zleceniu, ustalone i zaakceptowane przez Zamawiającego i Wykonawcę.

Na zakończenie realizacji zlecenia Wykonawca ma przekazać pliki pozwalające na aktualizację działającego oprogramowania CIBIS, w związku z wprowadzonymi modernizacjami oraz pliki instalacyjne zmodernizowanego oprogramowania CIBIS.

Odbiór prac w ramach każdego zlecenia odbywać się będzie na podstawie testów oprogramowania trwających nie dłużej niż 10 dni roboczych dla danego zlecenia. Zamawiający będzie zlecał prace programistyczne w postaci kolejnych Zleceń bez względu na trwające testy oprogramowania w ramach trwających odbiorów prac (kumulowanie Zleceń). Zmodernizowane oprogramowanie musi zachować wszystkie dotychczasowe funkcjonalności oprogramowania.

W sytuacji, gdy zamawiający stwierdzi że przedmiot zlecenia jest niezgodny z, SWZ, umową, załącznikami do umowy, ofertą wykonawcy lub nie będzie spełniał wymagań określonych w Opisie Przedmiotu Zamówienia oraz w Opisie Zlecenia niezwłocznie poinformuje o tym wykonawcę. Wykonawca usunie nieprawidłowości w terminie do 10 dni roboczych licząc od dnia zgłoszenia niezgodności. Prace te nie wliczają się do godzin programistycznych zlecenia.

Odbiór prac w ramach każdego zlecenia będzie potwierdzony każdorazowo protokołem odbioru zlecenia wykonania prac programistycznych i będzie podstawą do rozliczenia danego zlecenia na zasadach określonych w projektowanych postanowieniach umowy stanowiących Załącznik nr 2 do SWZ.

## **V. Termin wykonania zamówienia.**

Termin wykonania przedmiotu zamówienia: nie później niż do dnia 19.11.2021 r., przy czym wszystkie zadania wskazane w części IV pkt 1) ppkty 1.1)-1.13) powyżej powinny zostać zakończone w terminie 2 miesięcy od dnia zawarcia umowy.