

WYMAGANIA TECHNICZNE

1. Wymagania ogólne

1.1. Technologia wykonania (stos technologiczny)

Z powodu faktu, że Zamawiający posiada już rozwiązania, z którymi Zamawiane oprogramowanie musi się integrować aby uprościć utrzymywalność rozwiązania konieczne jest wybranie technologii które już w chwili obecnej są fundamentem rozwiązania.

1.1.1. Platformą do wykonania portali z interfejsem użytkownika jest .Net 5.0 lub wyższa. Warstwa interfejsu użytkownika może być zrealizowana przy pomocy mechanizmów Razor Pages, Controllers. Część interaktywna może być wykonana przy pomocy technologii Angular w najnowszej stabilnej wersji

1.1.2. Platformą do wykonania usług jest .Net 5.0 lub wyższy. Domyślnym standardem usług wymiany danych jest REST, a same usługi opisane są przy pomocy standardu Swagger. W przypadku gdy jest to dodatkowo oznaczone będą to usługi SOAP.

1.1.3. Aplikacje webowe jak i usługi będą hostowane w Azure jako Linux WebApps. Możliwe są odstępstwa od tej reguły i wykorzystanie w Azure mechanizmów Containers lub Azure Kubernetes Service.

1.1.4. Konfiguracja rozwiązania będzie oparta o natywne rozwiązania Microsoft Azure.

1.1.5. Dla rozwiązań opartych o relacyjną bazę danych bazą danych do wykonania rozwiązania jest Microsoft SQL Server.

1.1.6. W przypadku wykorzystania baz NOSQL preferowanym rozwiązaniem jest Azure Cosmos Db.

1.1.7. Zakłada się, że mechanizmy do monitoringu i diagnostyki również są oparte o natywne mechanizmy chmury Azure.

1.1.8. System musi zostać opracowany w architekturze co najmniej trójwarstwowej:

1.1.8.1. Warstwa Prezentacji

1.1.8.2. Warstwa Aplikacji

1.1.8.3. Warstwa Danych

1.2. Bezpieczeństwo

1.2.1. Aplikacja musi być dopracowana pod względem bezpieczeństwa; w szczególności przetestowana pod kątem niebezpieczeństw OWASP Top 10 (<https://owasp.org/www-project-top-ten/>)

1.2.2. Używane biblioteki w projekcie muszą być skanowane pod kątem znanych podatności i musi być przedstawiony raport dotyczący podatności.

1.2.3. Każda zmiana danych w systemie musi być identyfikowalna w zakresie użytkownika który dokonał zmiany i daty zmiany.

1.2.4. W kluczowych obszarach systemu oczekuje się, że będzie pełna historia zmian danych.

1.2.5. System musi być zgodny z RODO

1.2.6. System musi umożliwiać zabezpieczenie komunikacji z systemem przy użyciu połączenia certyfikatem SSL.

- 1.2.7. Poświadczenia i connection stringi w systemie będą zabezpieczone poprzez mechanizm Azure Key Vault.
- 1.2.8. Minimalnym akceptowalnym standardem funkcji skrótu jest SHA-2
- 1.2.9. Minimalnym akceptowalnym algorytmem symetrycznym jest AES 128
- 1.2.10. Minimalnym akceptowalnym algorytmem asymetrycznym jest RSA 2048
- 1.3. Środowiska
 - 1.3.1. Zamawiający dostarczy rozwiązanie na co najmniej dwa środowiska: produkcyjne i testowe
 - 1.3.2. Przed przekazaniem aplikacji na środowisko produkcyjne należy wykonać testy na środowisku testowym.
 - 1.3.3. Wykonawca musi zapewnić procedury zachowania spójności pomiędzy środowiskami w zakresie konfiguracji oprogramowania oraz danych.
 - 1.3.4. Konfiguracja środowiska testowego powinna odzwierciedlać środowisko produkcyjne w zakresie umożliwiającym wykonanie procedur testowych.
- 1.4. Szkolenia
 - 1.4.1. Po zrealizowaniu oprogramowania Wykonawca przeprowadzi nagrywane szkolenia z architektury oraz kodu źródłowego dla Zamawiającego w wysokości nie przekraczającej 40 rbh. Celem tego szkolenia jest przygotowanie Zamawiającego do rozwoju oprogramowania we własnym zakresie.
 - 1.4.2. Jeżeli podczas testów okaże się, że system nie jest na tyle intuicyjny, że użytkownicy mogą sobie radzić bez instrukcji, przewiduje się, że Wykonawca przygotuje materiały szkoleniowe.
- 2. Używalność (usability)
 - 2.1.1. Ergonomia
 - 2.1.1.1. System musi być wyposażony w graficzny, funkcjonalny i przyjazny interfejs użytkownika z uwzględnieniem ergonomii użytkownika.
 - 2.1.1.2. System musi umożliwić dostęp do pełnej funkcjonalności poprzez przeglądarkę w co najmniej wersjach (i wyższych):
 - Chrome 87
 - Firefox 84
 - Edge 87
 - 2.1.1.3. System musi umożliwić dostęp do pełnej funkcjonalności mobilnej poprzez system Android w wersji 6.0 lub wyższej.
 - 2.1.1.4. System nie może wymagać od przeglądarki żadnych rozszerzeń
 - 2.1.1.5. System musi zapewnić interfejs w języku polskim tam, gdzie to możliwe. Dotyczy to między innymi komunikatów, interfejsu użytkownika, raportów, dokumentacji, plików pomocy.
 - 2.1.1.6. System musi informować użytkowników o przyczynie niepowodzenia w komunikacji pomiędzy poszczególnymi

- komponentami systemu. Informacje o problemach muszą być gromadzone w logach.
- 2.1.1.7. System musi wykorzystywać mechanizm Single Sign On oparty o istniejący system Identity Server wykorzystując OpenId Connect i OAuth 2.0. Obecny system oparty jest o następujący projekt
<https://github.com/IdentityServer/IdentityServer4>
- 2.1.1.8. System musi zachowywać się transakcyjnie - złożona informacja zostanie na trwałe zapisana tylko wówczas, gdy zapisane zostaną wszystkie elementy składowe. Odstępstwa od tego mogą być tylko za zgodą Zamawiającego w uzasadnionych przypadkach i należy wtedy zapewnić mechanizmy kompensacji (eventual consistency)
- 2.1.2. Spójność
- 2.1.2.1. Tworzone moduły muszą być spójne pod względem nazewnictwa, słownictwa, grafik i interfejsu użytkownika. Pomimo rozbicia na różne moduły użytkownik powinien mieć poczucie, że pracuje z jednym, spójnym systemem.
- 2.1.2.2. System musi zapewnić pełną spójność danych we wszystkich modułach
- 2.1.3. Dostępność
- 2.1.3.1. Wszystkie moduły opracowywane muszą być pod względem dostępności zgodne z następującym dokumentem:
Załączniki nr 2 Standardy dostępności dla polityki spójności 2014-2020, a w szczególności rozdział VI Standard cyfrowy:
<https://rpo.slaskie.pl/file/download/3364>
- 2.1.4. Lokalizacja
- 2.1.4.1. System musi być przygotowany wykorzystując standardowe mechanizmy lokalizacji (wielojęzykowości) w platformie .Net i dla aplikacji mobilnych w oparciu o zasoby (Resources). Wraz z systemem oczekiwane są dwie wersje językowe - polska i angielska. Oczekuje się, że wykonawca przedstawi instrukcję lub szkolenie dodawania kolejnych języków do rozwiązania. Przewiduje się, że dodanie kolejnego języka do aplikacji może się wiązać z koniecznością kompilacji rozwiązania i drobnych zmian w kodzie (np. dodanie kolejnej pozycji rozwijanej, ikonki flagi), jednakże zmiany te powinny być marginalne.
- 2.1.4.2. Usługi REST API/SOAP nie muszą być lokalizowane.
- 2.1.5. Czas dostępności
- 2.1.5.1. System musi zapewniać dostępność ciągłą w systemie 24 godziny na dobę, 7 dni w tygodniu.
- 2.1.5.2. Poprzez nieplanowaną niedostępność Systemu rozumie się wystąpienie uniemożliwiające lub utrudniające pracę środowiska w sposób odczuwalny dla Zamawiającego.

- 2.1.5.3. Dostępność systemu WWW wynikająca z nieplanowanych przerw nie może być niższa, niż 99,5%
- 2.1.5.4. Dostępność systemu bezpośrednio wynikająca z nieplanowanych przerw nie wpływającego na możliwość deponowania odpadów nie może być niższa, niż 99,8%.
- 2.1.5.5. Dopuszczalna jest uzgodniona z Zamawiającym z co najmniej tygodniowym wyprzedzeniem przerwa w pracy Systemu na czas konserwacji. Przerwa ta nie może występować częściej, niż 2x w miesiącu i nie może trwać dłużej, niż 4 godziny. Wykonawca musi zaplanować tak przerwy, aby wypadały one w okresie o możliwie najniższym wykorzystaniu systemu.
- 2.1.6. Odzyskiwalność
 - 2.1.6.1. System powinien być tak zaprojektowany, aby czas przywrócenia do działania po awarii utraty danych.
 - 2.1.6.2. Wykonawca powinien określić czasy RTO i RPO dla systemu i przedstawić je zamawiającemu.
 - 2.1.6.3. Wykonawca przedstawi Zamawiającemu możliwe scenariusze związane z wysoką dostępnością rozwiązania i wpływem na koszty hostowanie w Azure.
- 2.1.7. Sprawdzalność
 - 2.1.7.1. System powinien być tak zaprojektowany i wdrożony, aby zapisywać w dziennikach zdarzeń błędy w działaniu aplikacji. Mając na myśli dzienniki zdarzeń nie ma się tutaj na myśli mechanizmu dziennika zdarzeń systemu Windows - tylko określone miejsce w systemie uzgodnione z Zamawiającym.
- 2.1.8. Mechanizmy naprawcze
 - 2.1.8.1. W ramach projektu systemu należy zaprojektować mechanizmy pozwalające na automatyczną naprawę systemu po awarii dla niektórych typów awarii (np. awaria jednego z węzłów)
- 2.1.9. Monitorowalność
 - 2.1.9.1. System powinien definiować metryki dla każdej kluczowej akcji, pozwalającej na analizę danych. Przykładem takich metryk może być liczba założonych kont z aplikacji mobilnej, liczba żądań resetu hasła, liczba deponowanych odpadów, wartość kaucji itp.
 - 2.1.9.2. Wykonawca określi kluczowe mierniki wydajnościowe dla systemu i aplikacji oraz określi wartości progowe dla tych liczników, dzięki którym możliwe będzie proaktywne monitorowanie Systemu. W szczególności wskazane zostaną przez Wykonawcę dopuszczalne wartości wskaźników wydajnościowych systemu których przekroczenie będzie uznawane za sytuację alarmową lub krytyczną.
 - 2.1.9.3. Wykonawca określi mechanizmy pozwalające na weryfikację dostępności systemu z punktu widzenia użytkownika końcowego. Dostarczone rozwiązanie powinno w sposób

jednoznaczny umożliwić weryfikację poziomu dostępności i jakości usług systemu.

3. Wydajność (performance)

3.1.1. Przepustowość

3.1.1.1. System po wdrożeniu i uruchomieniu powinien obsłużyć równoległe korzystanie przez 1000 użytkowników równolegle przy wykorzystaniu maszyny wirtualnej o parametrach 4 rdzenie i 8GB Ramu.

3.1.1.2. Wykonawca zobowiązany jest do przeprowadzenia optymalizacji samego Systemu oraz doboru odpowiednich parametrów celem uzyskania najwydajniejszego i najbardziej bezpiecznej konfiguracji Systemu.

3.1.2. Responsywność

3.1.2.1. System powinien 99% akcji standardowa zrealizować w czasie do 5 sekund zakładając obciążenie liczbą użytkowników nie przekraczając założonego w stosunku do użytych zasobów.

3.1.2.2. System powinien zapewnić przełączanie się między poszczególnymi elementami systemu w czasie do 3 sekund.

3.1.2.3. Generowanie raportu nie powinno przekroczyć 10 minut, a jeżeli czas będzie przekraczał 10 sekund należy odpowiednio poinformować użytkownika

3.1.3. Skalowalność

3.1.3.1. Należy przekazać informacje o skalowalności systemu wraz ze wzrostem przyznanych zasobów do systemu.

3.1.3.2. Należy przygotować mechanizmy służące do automatycznego skalowania rozwiązania w środowisku Microsoft Azure.

4. Wsparcie (Supportability)

4.1.1. Audytowalność

4.1.1.1. System musi zapewnić ślad audytorski umożliwiający śledzenie dokonywanych operacji i zmian w danych zawierający dokładny opis zmian, datę, godzinę, użytkownika i wartość zmiany.

4.1.1.2. Ślad audytowalny nie powinien być dostępny tylko w trybie do odczytu.

4.1.2. Kompatybilność

4.1.2.1. System musi umożliwić dostęp do pełnej funkcjonalności poprzez przeglądarkę w co najmniej wersjach (i wyższych):

- Chrome 87
- Firefox 84
- Edge 87

4.1.2.2. System musi umożliwić dostęp do pełnej funkcjonalności mobilnej poprzez system Android w wersji 6.0 lub wyższej.

4.1.3. Konfigurowalność

4.1.3.1. System powinien być konfigurowalny przez system parametryzacji oparty o standardowe mechanizmy Azure.

4.1.4. Instalowalność

- 4.1.4.1. Wraz z systemem Wykonawca prześle szereg skryptów które instalują odpowiednie komponenty w chmurze i dostarczą na nie odpowiednie oprogramowanie. Proces powinien być maksymalnie automatyczny.
- 4.1.4.2. Wykonawca dostarczy kroki w postaci skryptów konieczne do uruchomienia systemu na środowisku developerskim.
- 4.1.5. Testy
 - 4.1.5.1. System musi być testowalny przy pomocy mechanizmów testów jednostkowych.
 - 4.1.5.2. Zamawiający wymaga przeprowadzenia następujących testów wewnętrznych przez Wykonawcę:
 - funkcjonalnych
 - integracyjnych
 - wydajnościowych
 - bezpieczeństwa
 - 4.1.5.3. Zamawiający ma prawo do wglądu do przedstawionych wyników testów i spełnienie powyższych wymagań Zamawiającego.
 - 4.1.5.4. Oczekuje się, że wszystkie kluczowe funkcjonalności użytkownika indywidualnego, takie jak przykładowo elementy związane z deponowaniem odpadów czy odbiorem nagród będą zawierały testy End-to-End.
- 4.1.6. Standardy

Wykonawca zapewni zgodność systemu z powszechnie akceptowanymi standardami takimi jak między innymi:

 - REST
 - XSD
 - XLS
 - XLSX
 - CSV
 - SOAP
 - UTF-8
 - PDF
 - CSS
- 4.1.7. Łatwość pielęgnacji
 - 4.1.7.1. Wykonawca dostarczy kod źródłowy aplikacji wykonanych w ramach wdrożonego systemu. Do każdego oprogramowania zostanie dostarczona instrukcja kompilacji, instalacji i konfiguracji na środowisku developerskim i produkcyjnym. Wdrożenie w chmurze musi być zrealizowane poprzez skrypty instalacyjne oparte o mechanizmy Azure.
 - 4.1.7.2. Kod zrealizowany na platformę .Net w języku C# zostanie przygotowany zgodnie z obowiązującymi dobrymi praktykami, a w szczególności będzie zgodny z wytycznymi w załączniku nr 2.

- 4.1.7.3. W dostarczonym kodzie nie będzie ostrzeżeń (warnings) kompilatora.
- 4.1.7.4. Dostarczony kod będzie weryfikowany zgodnie ze statycznymi analizatorami kodu Code Analysis z wszystkimi włączonymi walidacjami. <https://docs.microsoft.com/en-us/dotnet/fundamentals/code-analysis/overview>
- 4.1.7.5. Standardy kodu będą zgodne z załącznikiem **Standardy kodu**.
- 4.1.7.6. Dostarczony kod będzie zgodny z ogólnie przyjętymi dobrymi wzorcami programowania takimi jak SOLID.