



Załącznik nr 3 do ZAPYTANIA OFERTOWEGO NR I/2024
o udzielenie zamówienia prowadzonego zgodnie z zasadą konkurencyjności
na zakup i wdrożenie oprogramowania, zakup urządzeń oraz przeprowadzenie szkoleń
w przedsiębiorstwie

Wymaganie funkcjonalne

1. Moduł skupu
 - 1.1. Możliwość dodawania i zarządzania informacjami o dostawcach, w tym dane kontaktowe, warunki handlowe oraz historia transakcji.
 - 1.2. Weryfikacja i autoryzacja dostawców w systemie, w celu zapewnienia zgodności z wymaganiami prawnymi i wewnętrznymi standardami firmy
 - 1.3. Automatyzacja procesu składania zamówień na surowce, z opcją ręcznego tworzenia zamówień przez uprawnionych użytkowników.
 - 1.4. Śledzenie statusu zamówień, od złożenia po dostawę, włącznie z aktualizacjami w czasie rzeczywistym.
 - 1.5. Funkcjonalność prognozowania, aby optymalizować zamówienia na surowce na podstawie danych historycznych i przewidywań popytu.
 - 1.6. Rejestracja przyjęcia surowców
 - 1.7. Integracja z systemami ważenia wag przemysłowych
 - 1.8. Integracja z RFID dla automatycznego rejestrowania wagi i innych danych krytycznych.
 - 1.9. Zarządzanie dokumentacją związaną z kontrolą jakości surowców przyjętych do magazynu
 - 1.10. Automatyczna aktualizacja stanów magazynowych po przyjęciu surowców.
 - 1.11. Optymalizacja poziomu zapasów na podstawie danych analitycznych, trendów zakupowych i prognozowanych potrzeb produkcji
 - 1.12. Możliwość generowania alertów przy niskim stanie magazynowym lub przekroczeniu zaplanowanych poziomów zapasów
 - 1.13. Automatyzacja procesu fakturowania i płatności dla dostawców
 - 1.14. Integracja z systemami FK Subiekt w celu zapewnienia spójności danych finansowych
 - 1.15. Możliwość generowania szczegółowych raportów dotyczących skupu surowców
 - 1.16. Analiza danych w czasie rzeczywistym, umożliwiającą szybką reakcję na zmiany w rynkowych warunkach surowców
 - 1.17. Generowanie dokumentów skupowych zgodnych z wymaganiami Inspekcji Jakości Handlowej Artykułów Rolno-Spożywczych
 - 1.18. Możliwość korzystania z funkcjonalności modułu skupu zarówno na terenie firmy jak również poza nią u dostawców
2. Moduł produkcji
 - 2.1. Funkcjonalność tworzenia harmonogramów produkcyjnych, które uwzględniają dostępność surowców, zdolności produkcyjne i zapotrzebowanie
 - 2.2. Wykorzystanie algorytmów optymalizacyjnych do maksymalizacji efektywności produkcji i minimalizacji odpadów
 - 2.3. Dostarczanie dashboardów z aktualnymi danymi o stanie produkcji, w tym wskaźnikami efektywności, statusami maszyn i postępami prac
 - 2.4. Konfiguracja systemu powiadomień o krytycznych zdarzeniach, takich jak przestoje maszyn, braki surowców czy odchylenia od planu produkcji

- 2.5. Wykorzystanie danych z systemu RFID do śledzenia przepływu materiałów na hali produkcyjnej, co pozwala na automatyczne aktualizowanie danych o zużyciu surowców.
- 2.6. Możliwość tworzenia szczegółowych raportów dotyczących wydajności produkcji
- 2.7. Generowanie dokumentów produkcyjnych zgodnych z wymaganiami Inspekcji Jakości Handlowej Artykułów Rolno-Spożywczych
3. Moduł laboratorium
 - 3.1. Możliwość łatwego wprowadzania danych o próbkach, w tym ich identyfikacja, pochodzenie, data i czas pobrania
 - 3.2. Automatyczne śledzenie statusu próbek od momentu ich przyjęcia, przez przetwarzanie, aż po usunięcie
 - 3.3. Wprowadzanie wyników testów, z możliwością dołączania dodatkowych notatek i dokumentacji.
 - 3.4. sprawdzanie zgodności wyników testów z normami jakościowymi.
 - 3.5. Funkcje do szybkiego raportowania i adresowania wszelkich niezgodności lub odchyłeń od norm
 - 3.6. Możliwość tworzenia szczegółowych raportów dotyczących działalności laboratorium
 - 3.7. Integracja modułu laboratorium z innymi modułami w przedsiębiorstwie, skup, zarządzania produkcją, magazynem.
4. Moduł magazyn
 - 4.1. Zapewnienie obsługi 5 magazynów na produkty gotowe
 - 4.2. Podział magazyny na rzędy, regał i poziomy z możliwością dynamicznego zarządzania.
 - 4.3. Funkcjonalność przyjęcia i wydania towarów z konkretnych miejsc paletowych.
 - 4.4. Funkcjonalność przesunięcia towarów.
 - 4.5. Funkcjonalność buforowania operacji przyjęcia i wydania towarów.
 - 4.6. Funkcjonalność walidacji możliwości wykonywania poszczególnych operacji.
 - 4.7. Raportowanie stanów magazynowych z podziałem na dowolne atrybuty opisujące produkty oraz czasu.
 - 4.8. Generowanie zamówień na podstawie dokumentów księgowych z Subiekt
 - 4.9. Generowanie dokumentów etykietowania dostosowanych do potrzeb wysyłki
 - 4.10. Wykorzystanie systemu RFID do identyfikacji i precyzyjnego lokalizowania surowców oraz produktów gotowych w magazynie.
 - 4.11. Tworzenie szczegółowych map magazynów, które umożliwiają łatwe śledzenie i optymalizację rozmieszczenia towarów.
 - 4.12. Śledzenie przemieszczeń towarów dzięki technologii RFID, co umożliwia aktualizację stanów magazynowych w czasie rzeczywistym.
 - 4.13. Redukcja błędów w inwentaryzacji i lokalizacji dzięki automatycznemu rejestrowaniu przepływu towarów.
 - 4.14. Optymalizacja procesów przyjęcia i wydania towarów w celu maksymalizacji efektywności operacyjnej magazynu.
 - 4.15. Zarządzanie poziomami zapasów magazynowych poszczególnych pozycji asortymentu
 - 4.16. Możliwość zarządzania strefami o specjalnych wymaganiach
 - 4.17. Możliwość tworzenia szczegółowych raportów dotyczących wydajności magazynu, obrotu towarowego i innych kluczowych wskaźników.
 - 4.18. Integracja z platformami e-commerce dla automatycznego zarządzania zamówieniami i realizacji wysyłek w tym zgodność ze standardem GS1

- 4.19. Implementacja strategii zarządzania przestrzenią magazynową, w tym dynamicznego przydziału miejsca na podstawie różnych cech produktów.

Wymaganie niefunkcjonalne

5. Ogólne

- 5.1. Interfejs użytkownika i administratora powinien posiadać przejrzyste i logiczne menu oraz jednolite rozwiązania graficzne. Językiem stosowanym w interfejsie jest język polski
- 5.2. Należy zapewnić intuicyjny i przyjazny interfejs użytkownika, który umożliwia łatwą nawigację i dostęp do informacji.
- 5.3. Portal powinien oferować interaktywne funkcje, takie jak formularze kontaktowe, wyszukiwarki i personalizację, które zwiększają zaangażowanie użytkowników.
- 5.4. Portal powinien spełniać standardy dostępności, takie jak WCAG 2.1.
- 5.5. Interfejs użytkownika portalu musi być responsywny, co oznacza, że powinien automatycznie dostosowywać się do różnych rozmiarów ekranów, od desktopów po smartfony
- 5.6. Portal powinien charakteryzować się wysoką szybkością ładowania stron, z czasem ładowania nie przekraczającym 3 sekund na połączeniach szerokopasmowych
- 5.7. Portal powinien być zgodny i optymalnie wyświetlany we wszystkich głównych przeglądarkach internetowych, takich jak Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, i Opera, zarówno w ich najnowszych wersjach, jak i kilku poprzednich
- 5.8. Należy przetestować i zapewnić prawidłowe funkcjonowanie portalu na urządzeniach mobilnych z systemami iOS i Android.
- 5.9. Aplikacja zapewni walidację wskazanych danych w tym wymuszanie na użytkownika wpisywanie danych zgodnie z ustalonymi formatami tj. daty, miary, kwoty, numery itp.
- 5.10. Aplikacja zapewni możliwość wprowadzania danych:
 - za pośrednictwem formularzy udostępnionych poprzez interfejs użytkownika;
 - za pośrednictwem wymiany danych z innymi systemami w technologii REST;
 - za pośrednictwem importu danych z plików xml, xls – migracja danych
- 5.11. Aplikacja zapewni rejestrowanie i śledzenie historii wprowadzanych danych oraz historię zmian danych biznesowych
- 5.12. Oczekiwane jest zapewnienie podstawowej dostępności i ciągłości pracy systemu 5 dni w tygodniu (poniedziałek-piątek), z wyłączeniem dni ustawowo wolnych od pracy, w godz. 6.00-22.00.
- 5.13. Dostarczenie pełnej dokumentacji technicznej dotyczącej architektury systemu, logiki biznesowej, schematów bazy danych, API oraz procedur wdrożeniowych i operacyjnych.

6. Aplikacja Web

- 6.1. Architektura: Aplikacja musi pracować w architekturze microfrontend
- 6.2. Modularyzacja: Aplikacja powinna być podzielona na mniejsze, niezależnie rozwijane i wdrażane mikroaplikacje, każda odpowiadająca za określoną funkcjonalność lub domenę biznesową
- 6.3. Niepowiązane wersje: Każda mikroaplikacja powinna być w stanie funkcjonować, być rozwijana i wdrażana niezależnie od innych, co umożliwia zespołom deweloperskim pracę z różnymi technologiami i frameworkami bez wpływu na całość systemu

- 6.4. Integracja: Wszystkie komponenty powinny być zintegrowane w jednolity interfejs użytkownika w ramach aplikacji szkieletowej, zapewniając płynność i spójność doświadczeń użytkownika, pomimo ich niezależnego rozwoju
- 6.5. Frameworka: Aplikacja powinna być napisana w oparciu o technologie Angulara v18 lub Reacta v18.
- 6.6. Interaktywność UI: SPA musi oferować wysoki poziom interaktywności i natychmiastowe reakcje na działania użytkowników bez przeładowywania strony
- 6.7. Stan Aplikacji: Zarządzanie stanem w aplikacji SPA powinno być wykonywane za pomocą zaawansowanych narzędzi i bibliotek (np. Redux dla React, NgRx dla Angulara), aby zapewnić spójność i łatwość zarządzania stanem na dużą skalę.
- 6.8. Izolacja Stylów: Stosowanie narzędzi takich jak CSS Modules lub styled-components w React, czy emulacja cienia DOM w Angularze, aby uniknąć konfliktów stylów między mikroaplikacjami.
- 6.9. Pakowanie i Optymalizacja: Wykorzystanie narzędzi takich jak Webpack lub Rollup do optymalizacji zasobów i minimalizacji kodu wynikowego.
- 6.10. Autentykacja i Autoryzacja: Implementacja zabezpieczeń na poziomie aplikacji, takich jak OAuth, JWT
- 6.11. Zabezpieczenia przed XSS/CSRF: Zaimplementowanie środków ochrony przed atakami typu Cross-Site Scripting (XSS) i Cross-Site Request Forgery (CSRF)..
7. Aplikacja mobilna
 - 7.1. Aplikacja powinna być kompatybilna z głównymi systemami operacyjnymi mobilnymi, takimi jak Android i iOS.
 - 7.2. Interfejs musi być intuicyjny i dostosowany do mniejszych ekranów, zapewniając łatwość nawigacji.
 - 7.3. Zastosowanie odpowiednich praktyk projektowania UX, które uwzględniają ograniczenia urządzeń mobilnych i zachowania użytkowników.
 - 7.4. Aplikacja powinna być zoptymalizowana pod kątem wydajności, minimalizując zużycie zasobów systemowych, takich jak bateria, pamięć i dane mobilne.
 - 7.5. Wybrane funkcjonalności mają możliwość funkcjonowania aplikacji w trybie offline, z wykorzystaniem lokalnego przechowywania danych, co jest kluczowe w miejscach z ograniczonym dostępem do internetu.
 - 7.6. Mechanizmy synchronizacji danych między urządzeniem mobilnym a serwerem, zapewniające aktualność danych po wznowieniu połączenia internetowego -
8. Platforma mikro-sewisowa
 - 8.1. Wszystkie funkcjonalności biznesowe muszą być napisane w oparciu o architekturę mikroserwisową
 - 8.2. Każdy mikroserwis powinien odpowiadać za określoną, izolowaną funkcję biznesową, co pozwoli na niezależne rozwijanie, testowanie i wdrażanie poszczególnych komponentów.
 - 8.3. Mikroserwisy mają być rozwijane w Spring Boot v3.3.3 lub Go v1.23, zgodnie z wymaganiami wydajności i specyfiką projektu.
 - 8.4. Mikroserwisy mają być konteneryzowane przy użyciu Docker i uruchamiane na K8S, co ułatwi ich wdrażanie, skalowanie i zarządzanie w różnorodnych środowiskach.
 - 8.5. Platforma K8S ma być tworzona w oparciu o oprogramowanie Rancher dzięki czemu będzie łatwo przenaszalna między dostawcami chmur, co oznacza, że można łatwo migrować aplikacje i zasoby pomiędzy różnymi dostawcami chmur, takimi jak własna chmura, AWS, Azure czy Google Cloud, bez konieczności zmiany architektury systemu.

- 8.6. Każdy mikroserwis odpowiedzialny za operacje na danych powinien być zintegrowany z bazą danych PostgreSQL
- 8.7. Należy zaimplementować mechanizmy zapewniające spójność danych między mikroserwisami, takie jak transakcje rozproszone lub wzorce kompensacyjne, jeśli operacje obejmują więcej niż jeden serwis.
- 8.8. Mikroserwisy powinny oferować RESTful API lub gRPC dla komunikacji zewnętrznej, umożliwiające łatwą integrację i standardizację.
- 8.9. Wszystkie usługi API powinny wymagać uwierzytelniania przy użyciu tokenów JWT, oraz szyfrowania SSL/TLS.
- 8.10. System powinien być zaprojektowany do łatwego skalowania poziomego, co umożliwi obsługę rosnącej liczby użytkowników i zapytań bez degradacji wydajności.
- 8.11. Wszystkie usługi API powinny być udokumentowane zgodnie z OpenAPI
9. Bezpieczeństwo
 - 9.1. Mechanizmy bezpieczeństwa Systemu muszą być oparte o zaawansowane rozwiązania klasy Identity Services (IS) oraz Identity and Access Management (IAM)
 - 9.2. Możliwość integracji z istniejącymi systemami zarządzania tożsamością (gmail, fb)
 - 9.3. Wsparcie dla standardów integracyjnych takich jak LDAP, Active Directory, OAuth, i SAML
 - 9.4. Portal powinien stosować aktualne standardy bezpieczeństwa, w tym szyfrowanie HTTPS na wszystkich Stronach
 - 9.5. Powinny być wdrożone mechanizmy ochrony przed atakami typu XSS, CSRF, SQL Injection i innymi powszechnymi wektorami ataków
 - 9.6. Zarządzanie Użytkownikami:
 - 9.6.1. Możliwość tworzenia, edycji, deaktywacji i usuwania kont użytkowników.
 - 9.6.2. Wymóg silnych haseł (min. długość, kombinacja znaków, wymagane symbole).
 - 9.6.3. Mechanizmy autoryzacji wieloskładnikowej (MFA) dla krytycznych ról użytkowników.
 - 9.6.4. Automatyczne wylogowanie po określonym czasie bezczynności.
 - 9.6.5. Blokadę konta użytkownika po określonej liczbie nieudanych
 - 9.7. Zarządzanie Rolami:
 - 9.7.1. Definiowanie ról i przypisywanie uprawnień zgodnie z zasadami najmniejszych uprawnień.
 - 9.7.2. Możliwość tworzenia niestandardowych ról dla różnych funkcji biznesowych.
 - 9.7.3. Weryfikacja i kontrola nad zmianami ról i uprawnień przez administratora bezpieczeństwa.
 - 9.8. Zarządzanie Uprawnieniami:
 - 9.8.1. Precyzyjne określenie uprawnień dostępu do funkcji i danych w systemie.
 - 9.8.2. Możliwość łatwego modyfikowania uprawnień w zależności od zmieniających się potrzeb organizacyjnych.
 - 9.8.3. Uprawnienia oparte na kontekście wykonywanych zadań, czasie dostępu i lokalizacji użytkownika.
 - 9.9. Audit i Śledzenie:
 - 9.9.1. Zapisywanie wszystkich działań użytkowników w zabezpieczonym dzienniku audytowym.
 - 9.9.2. Możliwość przeglądania i analizowania logów przez uprawnione osoby.
 - 9.9.3. Automatyczne raporty i powiadomienia o podejrzanych działaniach lub próbach naruszenia bezpieczeństwa.



9.9.4. Aplikacja powinna odkładać w logach wszystkie działania użytkowników:

- Dane autoryzacyjne: Data i czas logowania i wylogowania, próby nieudanego logowania, zmiany hasła i innych danych uwierzytelniających.
- Działania związane z danymi: Tworzenie, aktualizacja i usuwanie danych, w tym szczegóły dotyczące jakich danych dotyczyły te operacje.
- Zmiany w konfiguracji systemu: Wszelkie zmiany w konfiguracji systemu
- Działania administracyjne: Działania podejmowane przez użytkowników z uprawnieniami administracyjnymi, takie jak tworzenie nowych kont użytkowników, zmiana uprawnień użytkowników, interakcje z usługami systemowymi
- Błędy i awarie systemu: Szczegóły dotyczące błędów systemowych, awarii, problemów z wydajnością

9.10. Zarejestrowany, zautoryzowany i uwierzytelniony administrator będzie miał możliwość zarządzania słownikami. Będzie miał możliwość dodawania/ edycji/ usuwania/ aktywacji/ dezaktywacji danych słownikowych z zachowaniem pełnej historii zdarzeń i bez wpływu na strukturę danych historycznych

10. API

10.1. Wszyskie funkcjonalności biznesowe będą wysawione za pomocą API Gateway

10.2. Zapewni możliwość integracji z systemami zewnętrznymi w organizacji i poza organizacją.

10.3. Zapewni możliwość wymiany danych za pomocą Rest i gRpc

10.4. Dokumentacja

10.4.1. Wszystkie funkcjonalności zostaną udokumentowane w postaci kontraktów OpenAPI

10.4.2. Dokumentacja, zawierająca szczegółowe informacje o wszystkich dostępnych usługach, metodach, parametrach oraz typach danych

10.4.3. Przykłady użycia i odpowiedzi API dla wszystkich operacji ułatwiające integrację i testowanie.

10.5. Bezpieczeństwo

10.5.1. Zapewnienie szyfrowania danych w transmisji za pomocą protokołu HTTPS
Przeprowadzanie testów bezpieczeństwa, testy penetracyjne, oraz audyty zgodności z OWASP.

10.5.2. Aplikacja zapewni bezpieczeństwo wywołania API za pomocą tokenu JWT.
Uniemożliwi użytkownikowi wprowadzenia danych wykraczających poza dopuszczalny zakres uprawnień.

10.6. Inne

10.6.1. Aplikacja zapewni jasne i czytelne dla użytkownika komunikaty związane z błędami

10.6.2. Monitorowanie wydajności API, w tym czasów odpowiedzi i zużycia zasobów.

10.6.3. Optymalizacja wydajności poprzez caching odpowiedzi, zarządzanie sesjami i efektywne zarządzanie zasobami

10.6.4. Projektowanie API z myślą o przyszłej rozszerzalności i obsłudze wzrostu liczby użytkowników oraz zapytań

10.6.5. Wprowadzanie wersjonowania API, aby umożliwić równoczesne istnienie różnych wersji interfejsu.



10.6.6. Wprowadzenie limitów na ilość zapytań do API w danym czasie (rate limiting) w celu ochrony przed nadużyciami i zapewnienia sprawiedliwego dostępu dla wszystkich użytkowników

10.6.7. Mechanizmy transformacji danych, aby umożliwić konwersję między różnymi formatami i strukturami danych wymaganymi przez zintegrowane systemy

10.6.8. Konfigurowalne mapowanie danych, pozwalające na definiowanie, jak dane z jednego systemu są przekształcane i przypisane do struktur danych innego systemu.

10.6.9. Obsługa asynchronicznych mechanizmów przetwarzania danych

11. Monitoring

11.1. Wszystkie usługi oraz komponenty systemu są odpowiedzialne za zapisywanie logów i metryk w postaci JSON na jeden centralny system

11.2. Za zbieranie metryk z różnych mikroserwisów i infrastruktury odpowiedzialny jest Prometheus. Powinno to obejmować metryki systemowe, aplikacyjne i biznesowe

11.3. Za zbieranie logów z różnych mikroserwisów i infrastruktury odpowiedzialny jest FluentD który wysyła dane do Elastic.

11.4. Wizualizacja danych podzielona jest na
Metryki techniczne – Grafana
Metryki biznesowe – Grafana
Logi aplikacyjne – Kibana

11.5. Zapewnienie spersonalizowanych dashboardów w Grafanie, które zapewniają wgląd w kluczowe metryki operacyjne, wydajnościowe i biznesowe.

11.6. Konfiguracja kontroli dostępu na poziomie dashboardów, aby ograniczyć dostęp tylko do upoważnionych użytkowników.

11.7. Definiowanie polityk retencji danych, które zapewniają odpowiedni okres przechowywania danych i ich archiwizację dla celów audytowych i zgodności

11.8. System monitoringu musi być łatwy do skalowania w miarę wzrostu infrastruktury i liczby użytkowników, zarówno w kontekście zbierania danych, ich przechowywania, jak i wizualizacji.

11.9. Możliwość definiowania i zarządzania alertami w oparciu o zgromadzone metryki, z użyciem reguł bazujących na progach krytycznych

11.10. Dostarczenie kompleksowej dokumentacji technicznej i użytkownika dla systemu monitoringu, w tym instrukcji konfiguracji, obsługi i rozwiązywania problemów.

12. Integracja

12.1. Integracja z Systemem RFID

12.1.1. Integracja RFID dla śledzenia zasobów i automatyzacji procesów magazynowych, w tym odczytywanie i rejestrowanie ruchu produktów w czasie rzeczywistym.

12.1.2. Integracja z middleware RFID, które zarządza komunikacją między czytnikami RFID a systemem, przetwarzając dane z tagów.

12.1.3. System powinien automatycznie reagować na zdarzenia generowane przez system RFID, np. wejście/wyjście zasobów z określonych stref

12.2. Integracja z Subiektem GT

12.2.1. Automatyczna synchronizacja danych finansowych, zapasów i transakcji między systemem a Subiektem GT, zapewniająca spójność danych w całej organizacji.

12.2.2. Automatyzacja zamówień, fakturowania i innych procesów biznesowych poprzez integrację, co minimalizuje ryzyko błędów i zwiększa efektywność operacyjną.



12.3. **Integracja z SCADA**

12.3.1. Możliwość wymiany danych telemetrycznych i operacyjnych z systemem SCADA w czasie rzeczywistym, aby monitorować i kontrolować procesy przemysłowe.

12.3.2. Zapewnienie bezpiecznej komunikacji i autoryzacji między systemami, włącznie z szyfrowaniem transmisji danych.

12.4. **Integracja z Systemem Kamer**

12.4.1. System powinien wspierać odbiór i przetwarzanie strumieni wideo z kamer dla celów monitoringu i analizy w czasie rzeczywistym.

12.4.2. Interfejs API do integracji z oprogramowaniem zarządzającym kamerami, umożliwiające zdalne pobieranie obrazów.

12.4.3. System powinien obsługiwać automatyczne generowanie zdarzeń lub alarmów na podstawie analizy obrazu, np. wykrywanie ruchu w określonych strefach.

12.5. **Integracja z Systemem GS1**

12.5.1. Implementacja funkcjonalności umożliwiających kodowanie i dekodowanie standardów GS1, w tym kodów kreskowych i identyfikatorów produktów, co umożliwi łatwe śledzenie i zarządzanie informacjami o produktach na różnych etapach łańcucha dostaw

12.5.2.

12.6. **Integracja z wagami przemysłowymi**

12.6.1. Integracja z wagami przemysłowymi, aby automatycznie odczytywać i rejestrować wagę produktów lub surowców w systemie. Ta funkcjonalność jest kluczowa w procesach produkcyjnych, logistycznych oraz kontroli jakości.

12.6.2. Zbierane dane w czasie rzeczywistym muszą być przesyłane do modułu biznesowych systemu

13. Infrastruktura

13.1. **Ogólne**

Z racji dużych problemów z połączeniem internetowym w zakładzie produkcyjnym (brak światłowodu) należy zapewnić infrastrukturę sprzętową na miejscu w firmie oraz zapewnić mechanizmy synchronizacji danych do innych lokalizacji

13.2. Prywatny Klaster - Wdrożenie klastra Kubernetes na prywatnej infrastrukturze, z zapewnieniem odpowiedniej redundancji kluczowych komponentów.

13.3. Konfiguracja klastra w taki sposób, aby zapewnić wysoką dostępność (HA) i odporność na awarie, z automatycznym failoverem i możliwością szybkiego przywracania usług

13.4. Implementacja funkcji automatycznego skalowania zasobów (HPA - Horizontal Pod Autoscaler) w oparciu o obciążenie i inne metryki wydajności

13.5. Konfiguracja odpowiednich polityk sieciowych dla klastra, w tym izolacja sieciowa między różnymi mikroserwisami oraz kontrola dostępu do zasobów sieciowych

13.6. Konfiguracja regularnych kopii zapasowych ważnych danych i konfiguracji klastra, w tym etcd i danych aplikacji.

13.7. Regularne tworzenie kopii zapasowych bazy danych PostgreSQL, aby zapewnić ochronę danych i możliwość szybkiego przywrócenia systemu po awarii.

13.8. Implementacja strategii odzyskiwania po awarii (Disaster Recovery), w tym automatyczne przełączanie na serwery zapasowe i replikacja danych.

13.9. Opracowanie i testowanie procedur odtwarzania po awarii, aby zapewnić ciągłość działania usług

- 13.10. Zapewnienie bezpiecznego zdalnego dostępu do klastra Kubernetes oraz zasobów platformy wyłącznie przez sieć VPN (Virtual Private Network).
- 13.11. Wdrożenie silnych mechanizmów uwierzytelniania i autoryzacji dla użytkowników korzystających z VPN, w tym wykorzystanie wieloskładnikowej autentykacji (MFA) dla dodatkowego poziomu zabezpieczeń.
- 13.12. Konfiguracja zasad sieci VPN w taki sposób, aby ograniczać dostęp tylko do niezbędnych zasobów i usług, zgodnie z zasadami najmniejszych uprawnień (principle of least privilege)

RFID

1. Bramka RFID Wieloantenowa na Wjeździe do Magazynu Surowca (x1):

- **Lokalizacja i Ilość:** Dwukrotna instalacja bramek wieloantenowych na wjeździe do magazynu surowca.
- **Funkcjonalność:** Automatyczne skanowanie i identyfikacja tagów RFID UHF na produktach wjeżdżających do magazynu, zapis do bazy danych wyłącznie tagów własnych. Możliwością odczytu wielu tagów jednocześnie. Bramka musi pozwalać na weryfikację poprawności działania i wprowadzanie korekt na bieżąco.
- **Wydajność:** Bramka musi zapewniać wysoką dokładność i szybkość odczytu, nawet w warunkach intensywnego ruchu surowców,

2. Bramka RFID Wieloantenowa na Wjeździe do Magazynu Produktu Gotowego (x2):

- **Lokalizacja i Ilość:** Trzykrotna instalacja bramek wieloantenowych na wjeździe do magazynu produktu gotowego.
- **Funkcjonalność:** Automatyczne skanowanie i weryfikacja produktów gotowych wjeżdżających do magazynu, z równoczesnym odczytem wielu tagów. Bramka musi pozwalać na weryfikację poprawności działania i wprowadzanie korekt na bieżąco.
- **Integracja:** Bramka powinna być zintegrowana z systemem zarządzania magazynem, aktualizując na bieżąco informacje o stanie zapasów.

3. Bramka RFID w Szybie Windy (x1):

- **Lokalizacja:** Instalacja bramki RFID w szybie windy służącej do transportu między piętrami.
- **Funkcjonalność:** Monitorowanie przemieszczania się towarów między poziomami, w tym automatyczne skanowanie towarów przewożonych windą.
- **Dane:** System powinien rejestrować czas i kierunek przemieszczenia każdego towaru, wspomagając tym zarządzanie logistyczne.

4. Kolektor Danych Chainway RFID UHF 2D Android 11: (x3)

- **Specyfikacja:** Urządzenie mobilne z systemem Android 11, zdolne do odczytu tagów RFID UHF i kodów kreskowych 2D.
- **Zastosowanie:** Przenośne urządzenie do zbierania danych o ważonych workach oraz potwierdzania zużycia produktu na stanowisku homogenizatora.
- **Mobilność:** Kolektor musi być wytrzymały i ergonomiczny, zapewniając łatwość użytkowania w ruchu i różnych warunkach przemysłowych.

5. Licencja na Oprogramowanie:

- **Rodzaj Oprogramowania:** Oprogramowanie do zarządzania danymi zebranymi z systemów RFID oraz integracji danych z głównym systemem ERP.
- **Funkcjonalności:** Oprogramowanie powinno oferować funkcje gromadzenia, wyszukiwania i sortowania zebranych danych, możliwość ich importu z zewnętrznych systemów oraz eksportu do zewnętrznych systemów zarządzających. Odzwierciedlenia faktycznych przemieszczeń towarów zgodnie z zaprogramowanym schematem organizacyjnym firmy. Generowania etykiet RFID UHF i tworzenia powiązanych z nimi kartotek.

6. Drukarka Kodów Kreskowych Zebra ZD621R RFID:

- **Funkcjonalność:** Drukowanie etykiet RFID oraz kodów kreskowych wysokiej jakości. Weryfikacja sprawności etykiety. Automatyczne ponawianie wydruku w przypadku nieprawidłowej weryfikacji sprawności etykiety.
- **Zastosowanie:** Używana do etykietowania nowych zapasów i produktów, zapewniając ich śledzenie w całym łańcuchu dostaw.
- **Wydajność:** Drukarka musi być szybka, niezawodna i łatwa w obsłudze, umożliwiającą łatwą wymianę materiałów eksploatacyjnych.

Każdy z tych elementów wymaga starannego doboru, aby zapewnić pełną funkcjonalność i integrację z obecnymi systemami operacyjnymi w przedsiębiorstwie, co jest kluczowe dla poprawy efektywności i monitorowania zasobów.

Wymagania нефункционалне

1. Zakres Czytania:

System RFID powinien posiadać zdolność do czytania tagów RFID na odpowiednich dla aplikacji odległościach, zarówno w warunkach bliskich, jak i na większych dystansach, w zależności od zastosowanej technologii UHF.

2. Szybkość Czytania:

System powinien być w stanie szybko i efektywnie rejestrować dane z wielu tagów jednocześnie, minimalizując czas potrzebny do zidentyfikowania i przetworzenia informacji z każdego tagu.

3. Pamięć i Przetwarzanie Danych:

Tagi RFID muszą mieć odpowiednią pojemność pamięci do przechowywania identyfikatora. Szczegółowe dane, takie jak unikalne ID, dane o produkcie, data wygaśnięcia, itp. przechowywane są w serwerowej bazie danych.

4. Odporność na Warunki Środowiskowe:

Tagi i czytniki RFID powinny być odporne na warunki środowiskowe takie jak temperatura, wilgotność, zanieczyszczenia, czy wpływ substancji chemicznych, zgodnie z wymaganiami specyficznymi dla środowiska, w którym są stosowane.

5. Integracja z Istniejącymi Systemami:

System RFID musi być kompatybilny i możliwy do integracji z istniejącymi systemami zarządzania zapasami, systemami ERP, a także innymi technologiami używanymi w przedsiębiorstwie.

6. Bezpieczeństwo Danych:

Wdrożenie odpowiednich środków bezpieczeństwa dla zapewnienia ochrony danych zapisanych na tagach RFID przed nieautoryzowanym dostępem, w tym szyfrowanie danych.

7. Zasięg i Wydajność Systemu:

System RFID powinien oferować wystarczający zasięg operacyjny i wydajność, aby zapewnić nieprzerwane i dokładne śledzenie zasobów na całym obszarze jego zastosowania.

8. Wsparcie Techniczne i Serwis:

Dostawca systemu RFID powinien zapewnić pełne wsparcie techniczne oraz serwis w razie awarii, problemów technicznych czy potrzeby aktualizacji systemu.

9. Elastyczność Konfiguracji:

System powinien oferować elastyczne opcje konfiguracji, aby móc dostosować funkcjonalność do specyficznych potrzeb operacyjnych użytkownika, w tym możliwość modyfikacji parametrów pracy tagów i czytników.

10. Rozszerzalność Systemu:

System RFID powinien być projektowany z myślą o przyszłych rozszerzeniach i skalowaniu, pozwalając na łatwe dodawanie nowych komponentów i funkcjonalności w miarę rozwoju potrzeb użytkownika.

Kolektory mają możliwość funkcjonowania aplikacji w trybie offline, z wykorzystaniem lokalnego przechowywania danych, co jest kluczowe w miejscach z ograniczonym dostępem do internetu.

Mechanizmy synchronizacji danych między urządzeniem mobilnym a serwerem, zapewniające aktualność danych po wznowieniu połączenia internetowego

Wykorzystanie funkcjonalności urządzeń, takich jak aparat fotograficzny, GPS, akcelerometr do realizacji funkcji specyficznych dla aplikacji.

Wymóg ten ma na celu zapewnienie, że wybrany dostawca posiada nie tylko odpowiednią wiedzę techniczną, ale również doświadczenie praktyczne, które jest kluczowe dla efektywnego i bezproblemowego wdrożenia systemu RFID, minimalizując ryzyko oraz maksymalizując potencjalne korzyści z jego użytkowania.