Specyfikacja wymagań

Software Requirements Specification

# Wersja

|  |  |  |  |
| --- | --- | --- | --- |
| Wersja | Data | Autor | Zakres zmian |
| 0.1.0 | 2020-10-07 | Łukasz Obłąk  Wojciech Barłowski  Nikola Kowalik  Kamil Pawłowski  Michał Szczepaniak  Kacper Wielągowski |  |

# Spis treści

[Wersja 1](#_Toc52991313)

[Spis treści 1](#_Toc52991314)

[1. Wprowadzenie 2](#_Toc52991315)

[1.1 Cel 2](#_Toc52991316)

[1.2 Przyjęte konwencje 2](#_Toc52991317)

[1.3 Zakres projektu 2](#_Toc52991318)

[1.4 Odwołania 2](#_Toc52991319)

[2. Opis ogólny 2](#_Toc52991320)

[2.1 Perspektywa produktu 2](#_Toc52991321)

[2.2 Klasy i charakterystyki użytkowników 2](#_Toc52991322)

[2.3 Środowisko działania 3](#_Toc52991323)

[2.4 Ograniczenia projektowe i wykonawcze 3](#_Toc52991324)

[2.5 Założenia i zależności 3](#_Toc52991325)

[3. Funkcjonalności systemu 3](#_Toc52991326)

[3.x Funkcjonalność XYZ 3](#_Toc52991327)

[3.x.1 Opis 3](#_Toc52991328)

[3.x.2 Wymagania funkcjonalne 3](#_Toc52991329)

[4. Wymagania dotyczące danych 3](#_Toc52991330)

[4.1 Logiczny model danych 3](#_Toc52991331)

[4.2 Raporty 3](#_Toc52991332)

[4.3 Pozyskiwanie, integralność przechowywanie I usuwanie danych 4](#_Toc52991333)

[5. Wymagania dotyczące interfejsu zewnętrznego 4](#_Toc52991334)

[5.1 Interfejsy użytkownika 4](#_Toc52991335)

[5.2 Interfejsy programowe 4](#_Toc52991336)

[5.3 Interfejsy sprzętowe 4](#_Toc52991337)

[5.4 Interfejsy komunikacyjne 4](#_Toc52991338)

[6. Cechy jakości 5](#_Toc52991339)

[6.1 Użyteczność 5](#_Toc52991340)

[6.2 Wydajność 5](#_Toc52991341)

[6.3 Zabezpieczenia 5](#_Toc52991342)

[6.4 Bezpieczeństwo użytkowania 5](#_Toc52991343)

[6.x Inne 5](#_Toc52991344)

[7. Wymagania internacjonalizacji i lokalizacji 5](#_Toc52991345)

[8. Inne wymagania 5](#_Toc52991346)

[Suplement A: Słownictwo 5](#_Toc52991347)

[Suplement B: Modele analityczne 5](#_Toc52991348)

# 1. Wprowadzenie

We wprowadzeniu znajduje się przegląd, który ma pomóc czytelnikowi zrozumieć, jak zorganizowany jest ten dokument i jak z niego korzystać.

## 1.1 Cel

Zidentyfikuj produkt lub aplikację, których wymagania są określone w tym dokumencie, w tym numer wersji lub wydania.

Jeśli ten SRS dotyczy tylko części złożonego systemu, zidentyfikuj tę część lub podsystem.

Opisz odbiorców, dla których ten dokument jest przeznaczony (programiści, kierownicy projektów, marketing, użytkownicy, testerzy, dział wdrożeń itp.)

Dostarczenie wspólnego zrozumienia wymagań funkcjonalnych i niefunkcjonalnych systemu IWP dla wszystkich interesariuszy.

Celem projektu jest stworzenie aplikacji webowej służącej do inwentaryzacji i ewidencji uposażenia domowego z podziałem na kategorie.

Dokument jest przeznaczony dla programistów i testerów.

## 1.2 Przyjęte konwencje

Opisz tu przyjęte standardy I konwencje, znaczenie formatowania.

# 1.3 Zakres projektu

Podaj krótki opis planowanego oprogramowania i jego przeznaczenie. Powiąż oprogramowanie z celami użytkownika lub firmy oraz celami biznesowymi.

Jeśli dokument wizji / zakresu istnieje osobno, należy go tutaj przywołać, zamiast powielać jego treść.

W przypadku projektów iteracyjnych, SRS powinien zawierać własne określenie zakresu jako podzbiór długoterminowej strategii rozwoju produktu.

Można podać ogólne podsumowanie głównych funkcji produktu.

Produkt ma na celu obsługę wszystkich aspektów inwentaryzacyjnych, z czego najważniejsze aspekty to:

* Modyfikacja, wyświetlanie, dodawanie i usuwanie przedmiotów inwentarza
* Zmiana możliwych typów i lokalizacji produktów inwentaryzowanych
* Wyszukiwanie
* Wyświetlanie powiadomień o brakujących i przeterminowanych produktach

## 1.4 Odwołania

Odwołania do wszystkich przydatnych dokumentów, do których odwołuje się SRS.

# 2. Opis ogólny

Ta sekcja przedstawia ogólny przegląd produktu i środowiska, w którym będzie używany, przewidywanych użytkowników oraz znanych ograniczeń, założeń i zależności

## 2.1 Perspektywa produktu

Opisz kontekst produktu. Czy to nowy produkt, czy kolejny fragment większego systemu? W drugim przypadku, należy określić, w jaki sposób oprogramowanie odnosi się do całego systemu i zidentyfikować główne interfejsy. Należy rozważyć użycie diagramów wizualnych do ilustracji tych zależności.

Projekt zakłada utworzenie całkowicie nowego produktu.

## 2.2 Klasy i charakterystyki użytkowników

Należy zidentyfikować i scharakteryzować przewidywane klasy użytkowników końcowych tego produktu.

Opisy klas użytkowników mogą być używane wielokrotnie. Jeśli dostępny jest katalog głównych klas użytkowników, można zamiast powielać treści wstawić tutaj odwołanie do zewnętrznych dokumentów.

Użytkownik – korzysta z aplikacji.

## 2.3 Środowisko działania

Należy opisać środowisko, w którym będzie działać oprogramowanie, w tym platformę sprzętową, systemy operacyjne, geograficzne lokalizacje użytkowników, serwerów i baz danych (NA, EU, APAC) itp.

Należy wymienić wszelkie inne składniki oprogramowania lub aplikacje, z którymi system musi współpracować / współistnieć.

Przeglądarki internetowe:

* Google Chrome
* Opera
* Edge
* Mozilla Firefox

## 2.4 Ograniczenia projektowe i wykonawcze

W niektórych przypadkach język programowania lub biblioteki do wykorzystania są z góry określone. Należy opisać wszelkie czynniki, które ograniczą opcje dostępne dla programistów, oraz uzasadnienie każdego ograniczenia.

## 2.5 Założenia i zależności

Należy wymienić tutaj założenia dotyczące funkcjonalności systemu. Ponadto należy zidentyfikować wszelkie zależności budowanego produktu od czynników zewnętrznych lub komponentów pozostających poza jego kontrolą.

Przykładowo jeśli przed uruchomieniem produktu należy zainstalować system kontenerów Docker lub serwer web NGINX / Apache, należy taką zależność wskazać w tej sekcji.

Java + Spring

MySQL + MySQL Workbench 8.0

Git + Github

Docker

Vue + Vite

# 3. Funkcjonalności systemu

Wymagania funkcjonalne można zorganizować według obszaru funkcjonalnego, przebiegu procesu, przypadku użycia, trybu działania, klasy użytkownika itp. Możliwe są również hierarchiczne kombinacje tych elementów, takie jak przypadki użycia w klasach użytkowników. Należy wybrać taką metodę organizacji, która ułatwi czytelnikom zrozumienie zamierzonych możliwości produktu.

## 3.1: Konto użytkownika

### 3.1.1 Opis

Priorytet: wysoki

Ogół wymagań umożliwiających użytkownikowi tworzenie i zarządzanie kontem.

### 3.1.2 Wymagania funkcjonalne DO UZUPELNIENIA

USR-1: Użytkownik musi mieć możliwość rejestracji w systemie.

USR-2: Użytkownik musi mieć możliwość zalogowania się w systemie.

USR-3: Użytkownik musi mieć wylogowania się z systemu.

USR-4: Użytkownik musi mieć możliwość odzyskania konta po utracie hasła metodą potwierdzenia poprzez email.

USR-5: Użytkownik musi mieć możliwość usunięcia konta.

USR-6: Użytkownik musi mieć możliwość zmiany hasła.

## 3.2 Inwentaryzacja

### 3.2.1 Opis

Priorytet: wysoki

Ogół wymagań umożliwiających użytkownikowi zarządzanie inwentarzem.

### 3.2.2 Wymagania funkcjonalne

INW-1: System musi umożliwić użytkownikowi wyświetlanie wszystkich przedmiotów.

INW-2: System musi umożliwić użytkownikowi dodawanie przedmiotów.

INW-3: System musi umożliwić sortowanie przedmiotów po nazwie, kategorii, dacie ważności.

INW-4: Użytkownik musi mieć możliwość utworzenia nowej kategorii przedmiotu.

INW-5: Użytkownik musi mieć możliwość usuwania przedmiotu z inwentarza.

INW-6: Użytkownik musi mieć możliwość edycji inwentarza.

INW-7: Użytkownik musi mieć możliwość wyszukiwania przedmiotu o określonej nazwie.

INW-8: Użytkownik musi mieć możliwość wyświetlenia powiadomień o brakujących i przeterminowanych produktach.

INW-9: Użytkownik musi mieć możliwość wprowadzenia limitu ilości danego przedmiotu od którego wyskakuje powiadomienie o niskiej ilości.

# 4. Wymagania dotyczące danych

W tej sekcji szablonu należy opisać różne aspekty danych, które system będzie wykorzystywał jako dane wejściowe, przetwarzał w jakiś sposób lub produkował jako dane wyjściowe.

## 4.1 Logiczny model danych

Model danych jest wizualną reprezentacją obiektów danych i zbiorów, które system będzie przetwarzał, oraz relacji między nimi. Istnieje wiele notacji do modelowania danych, w tym diagramy ERD. To nie to samo, co implementacyjny model danych, który zostanie zrealizowany w postaci projektu bazy danych.

<https://online.visual-paradigm.com/share.jsp?id=323231313636372d31> link do ERD w visual paradigm online

## 4.2 Raporty

Jeśli system będzie generować raporty, należy je tutaj zidentyfikować i opisać ich cechy. Jeśli raport musi być zgodny z określonym, wstępnie zdefiniowanym układem, można określić to tutaj jako ograniczenie lub podać przykład. Alternatywnie, można opisać zawartości raportu, kolejności sortowania, sumowaniu poziomów itd.

Np. wydruk tabeli

## 4.3 Pozyskiwanie, integralność przechowywanie I usuwanie danych

W stosownych przypadkach opisz, w jaki sposób dane są gromadzone i utrzymywane. Określić wszelkie wymagania dotyczące potrzeby ochrony integralności danych systemu. Zidentyfikować niezbędne techniki, takie jak kopie zapasowe, punkty kontrolne, kopie lustrzane.

Backup, export danych, możliwość expotu części danych

# 5. Wymagania dotyczące interfejsu zewnętrznego

Ta sekcja zawiera informacje zapewniające prawidłową komunikację systemu z użytkownikami oraz z zewnętrznymi elementami sprzętu lub oprogramowania. Złożony system z wieloma mikroserwisami powinien tworzyć oddzielną specyfikację interfejsu lub specyfikację architektury systemu. Dokumentacja interfejsu może zawierać materiały z innych dokumentów poprzez odwołania.

## 5.1 Interfejsy użytkownika

Opisz logiczną charakterystykę każdego interfejsu użytkownika, którego potrzebuje system. Niektóre specyficzne cechy interfejsów użytkownika mogą pojawić się w 6.1 Użyteczność. Należy zwrócić uwagę na:

* Odniesienia do standardów interfejsu użytkownika, których należy przestrzegać
* Standardy dotyczące czcionek, ikon, etykiet przycisków, obrazów, schematów kolorów, sekwencji zakładek w polach, powszechnie używanych elementów sterujących, grafiki marki, informacji o prawach autorskich i prywatności itp.
* Ograniczenia rozmiaru ekranu, układu lub rozdzielczości
* Standardowe przyciski, funkcje lub łącza nawigacyjne, które pojawią się na każdym ekranie, na przykład przycisk pomocy
* Skróty klawiszowe
* Wyświetlanie wiadomości i konwencje budowy / składni komunikatów
* Wytyczne dotyczące walidacji danych (takie jak ograniczenia wartości wejściowych i kiedy sprawdzać zawartość pól)
* Standardy układu ułatwiające lokalizację (tłumaczenie) oprogramowania
* Udogodnienia dla użytkowników niedowidzących, daltonistów lub mających inne ograniczenia

## 5.2 Interfejsy programowe

Opisz połączenia między tym produktem a innymi składnikami oprogramowania (zidentyfikowanymi według nazwy i wersji), w tym innymi aplikacjami, bazami danych, systemami operacyjnymi, narzędziami, bibliotekami, witrynami internetowymi i zintegrowanymi komponentami komercyjnymi.

Określić cel, formaty i treść komunikatów, danych i wartości kontrolnych wymienianych między komponentami oprogramowania.

Określ odwzorowania danych wejściowych i wyjściowych między systemami oraz wszelkie tłumaczenia, które muszą zostać wykonane, aby dane mogły zostać przesłane z jednego systemu do drugiego.

Aplikacja podzielona na backend i frontend komunikuje się pomiędzy tymi warstwami za pomocą fetch + json. Aplikacja łączy się z bazą danych mysql w celu przechowywania danych.

## 5.3 Interfejsy sprzętowe

Opisz charakterystykę każdego interfejsu między komponentami oprogramowania i komponentami sprzętowymi systemu, jeśli takie istnieją. Opis ten może obejmować obsługiwane typy urządzeń, dane i interakcje sterowania między oprogramowaniem a sprzętem oraz używane protokoły komunikacyjne. Dane wejściowe i wyjściowe, ich formaty, prawidłowe wartości lub zakresy oraz wszelkie problemy z synchronizacją, o których programiści powinni wiedzieć.

Użytkownik porusza się po aplikacji używając klawiatury i myszki.

## 5.4 Interfejsy komunikacyjne

Określić wymagania dotyczące wszelkich funkcji komunikacyjnych, których będzie używać produkt, w tym poczty e-mail, przeglądarki internetowej, protokołów sieciowych i formularzy elektronicznych.

Zdefiniować odpowiednie formaty komunikatów. Określ zasady bezpieczeństwa i szyfrowania komunikacji, szybkości przesyłania danych, mechanizmy synchronizacji.

Gdy użytkownik zaloguje się ze swojego konta na inne urządzenie, nadal będzie widział te same dane w aplikacji.

# 6. Cechy jakości

W tej sekcji określa się wymagania niefunkcjonalne inne niż ograniczenia, które są zapisane w sekcji 2.4 oraz 5.1.

Te wymagania jakościowe powinny być szczegółowe, ilościowe i weryfikowalne.

Należy wskazać względne priorytety różnych atrybutów, np. bezpieczeństwo ponad wydajnością.

## 6.1 Użyteczność

Wymagania dotyczące użyteczności dotyczą łatwości uczenia się, łatwości użytkowania, wydajności interakcji i dostępności. Określone tutaj wymagania dotyczące użyteczności pomogą projektantowi interfejsu użytkownika stworzyć optymalne user experience.

* Strona responsywna dostosowująca się do wymiarów wyświetlacza
* Możliwość wpisywania danych w formie szybkiej (w formie arkusza)

## 6.2 Wydajność

Określić wymagania dotyczące wydajności dla różnych operacji systemowych. Jeśli różne wymagania funkcjonalne lub cechy mają różne wymagania dotyczące wydajności, lepiej jest określić te cele wydajnościowe razem z odpowiednimi wymaganiami funkcjonalnymi, zamiast zbierać je w tej sekcji.

## 6.3 Zabezpieczenia

Określ wszelkie wymagania dotyczące kwestii bezpieczeństwa lub prywatności. Mogą one odnosić się do bezpieczeństwa fizycznego, danych lub oprogramowania.

* Weryfikacja użytkownika poprzez wysłanie czasowego tokena drogą mailową.
* Podczas logowania używa się jwt
* Użycie bezpiecznych szyfrów do szyfrowania danych wrażliwych
* Użycie architektury REST

## 6.4 Bezpieczeństwo użytkowania

Określ wymagania dotyczące możliwych szkód, które mogą wynikać z użytkowania produktu. Zdefiniuj wszelkie zabezpieczenia lub działania, które należy podjąć, a także potencjalnie niebezpieczne działania, którym należy zapobiec.

## 6.x Inne

Utwórz osobną sekcję w SRS dla każdego dodatkowego atrybutu jakości produktu, aby opisać cechy, które będą ważne zarówno dla klientów, jak i dla programistów.

# 7. Wymagania internacjonalizacji i lokalizacji

Wymagania dotyczące umiędzynarodowienia i lokalizacji zapewniają, że produkt będzie odpowiedni do użytku w innych krajach, kulturach i lokalizacjach geograficznych niż te, w których został stworzony. Przykładowo różnice walutowe; formatowanie dat, numerów, adresów czy numerów telefonów. Wymagania dotyczące umiędzynarodowienia i lokalizacji mogą być ponownie wykorzystywane w różnych projektach.

Angielski, Polski

# 8. Inne wymagania

Opcjonalnie można zdefiniować dowolne inne wymagania nie zdefiniowane wcześniej w tym SRS. Przykładowo wymagania prawne, finansowe; wymagania instalacji produktu, jego konfiguracji, uruchomienia czy wyłączenia.

# Suplement A: Słownictwo

Należy zdefiniować tutaj wszelkie skróty i specjalistyczne słownictwo używane w tym dokumencie.

# Suplement B: Modele analityczne

W tej opcjonalnej sekcji należy umieścić wszelkie modele analityczne które pomogą czytelnikowi zrozumieć system. Przykładowo drzewa funkcjonalności, diagramy ERD, procesy BPMN. Czasem czytelniejsze może być umieszczenie tych diagramów bezpośrednio w treści SRSa.