Specyfikacja wymagań

Software Requirements Specification

# Wersja

|  |  |  |  |
| --- | --- | --- | --- |
| Wersja | Data | Autor | Zakres zmian |
| 0.1.0 | 2022-10-31 | Artur Karczmarczyk | Inicjalizacja szablonu dokumentu |

# 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

Dokumentacja opisuje realizację projektu serwisu internetowego ‘ ¿Donde? ’. Serwis ma na celu wyszukiwanie miejsca pobytu pracowników Wydziału Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego w Szczecinie oraz lokalizację pomieszczeń w budynkach Wydziału Informatyki. Niniejsza dokumentacja stanowi kluczowy element procesu wdrożenia serwisu w życie.

## 1.1 Cel

Celem serwisu internetowego ¿Donde? jest ułatwianie lokalizowania pracowników oraz pomieszczeń WI ZUT

w określonym przedziale czasowym.

Dokument jest przeznaczony dla zespołu projektowego, dla użytkowników aplikacji oraz dla szanownego grona dydaktycznego sprawdzającego pracę.

## 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.

## Główne funkcje systemu:

## Ekran startowy z odnośnikami do najważniejszych części systemu.

## Sekcja MAPA, która wyświetla dwa budynki WI ZUT wraz ze zdjęciami, adresami i mapami z zaznaczonymi budynkami.

## Sekcja PLAN, która prezentuje wszystkie piętra danego budynku z numerami pomieszczeń. Kliknięcie lub podświetlenie pomieszczenia spowoduje wyświetlenie rodzaju pomieszczenia (sala wykładowa, laboratorium, audytoria), link do planu zajęć, obecnych lub najbliższych zajęć oraz danych osoby pracującej w pomieszczeniu (tytuł, imię, nazwisko, link do planu zajęć, miejsca, w których zazwyczaj można ją znaleźć).

## Sekcja SZUKAJ, która umożliwia wyszukiwanie pracownika lub pomieszczenia o określonym czasie lub dacie. Po wypełnieniu formularza wyświetlają się potencjalni pracownicy przebywający w danym pomieszczeniu, plan z zaznaczonym pomieszczeniem i mapa, lub informacje o pomieszczeniu, w którym może być pracownik o określonym czasie lub dacie (rodzaj pomieszczenia, adres, plan, mapa).

## 1.4 Odwołania

Dokumentacja uzupełniająca:

Dokument - ‘przypadki\_użycia\_WBS’, w którym znajdują się:

- przypadki użycia

- WBS

Dokument – ‘WBS-Delphi’ zawiera wycenę czasu pracy metodą Delphi

# 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.

System Donde jest nowym produktem, jest dostępny na każdą przeglądarkę internetową wspierającą HTML5

i JavaScript oraz wymaga połączenia z Internetem. System jest zależny od dostępu do bazy danych zawierającej informacje o pracownikach i pomieszczeniach wydziału.

Opiera swoją funkcjonalność o API:

* Leaflet - do wyświetlania mapy
* plan.zut.edu.pl - do pobierania informacji o planie pracownika.

Główne interfejsy - system składa się z siedmiu stron (7 plików html):

1. index.html – strona początkowa z możliwością wyboru przejścia na stronę wyszukiwarki, stronę wyboru budynku WI ZUT lub stronę logowania.
2. wyszukiwarka.html – strona z polem wyszukiwania służąca do znajdowania pracowników WI ZUT w określonych przez użytkownika godzinach lub do znajdowania pomieszczeń WI ZUT wraz z przypisanymi do nich informacjami (nazwa / numer pomieszczenia, zastosowanie pomieszczenia, jeśli jest to gabinet to kogo). Po wyszukaniu poniżej pola wyszukiwarki pojawia się wynik albo przekierowanie na stronę budynku z odpowiednio zaznaczonym pomieszczeniem.
3. mapa.html - strona z wyborem budynku WI ZUT (WI1 lub WI2).
4. plan\_wi1.html - strona z interaktywnymi planami budynku WI1 oraz informacjami.
5. plan\_wi2.html - strona z interaktywnymi planami budynku WI2 oraz informacjami.
6. login\_page.html - strona logowania dla admina.
7. admin.html – panel dla admina.

## 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.

user - osoba, która posługuje się oprogramowaniem do lokalizacji pracowników WI ZUT w godzinach pracy lub znaleźć lokalizację konkretnego pomieszczenia. user chce znaleźć miejsce pobytu pracownika WI ZUT w określonym czasie. Może to być z różnych względów zarówno student, wykładowca, dziekan, lub inny pracownik ZUT.

admin - osoba która będzie odpowiedzialna za wprowadzanie zmian w bazie danych lub zmian informacji o pomieszczeniach, np. oznaczanie sal w remoncie i innych sytuacjach.

## 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.

Aplikacja może być uruchomiona na systemach operacyjnych Windows, Linux lub MacOS przy pomocy przeglądarki wspierającej HTML5 i JavaScript (np. Chrome, Firefox, Opera, Edge). Będzie dostępna dla użytkowników niezależnie od ich lokalizacji.

Do poprawnego działania potrzebna jest baza danych np. MySQL, w której będą przechowywane informacje dotyczące planu, nauczycieli i sal. Aplikacja będzie współpracować z API plan.zut.edu.pl oraz biblioteką leaflet.

## 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.

Ograniczeniem są wymagania niefunkcjonalne zawarte w dokumencie ‘AI1-LAB10-siwz-donde’

Obraz zawierający tekst

Opis wygenerowany automatycznie

Uzasadnienie: taka jest wola prowadzących przedmiot.

## 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.

Aplikacja:

* Wyświetla informacje znajdujące się w bazie danych dla użytkowników, w tym informacje o planie, nauczycielach czy salach
* Pobiera informacje o planie z plan.zut.edu.pl
* Wyświetla mapę dzięki Leaflet API
* Obsługuję formularze do wpisywania nowych danych czy wyszukiwania informacji dzięki PHP.
* Czynniki zewnętrzne lub komponenty pozostające poza kontrolą systemu:
  + leaflet API
  + plan.zut.edu.pl

# 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 Wyszukiwanie pracownika lub pomieszczenia (publiczne)

Należy podać nazwę funkcji w kilku słowach, na przykład „3.1 Zarządzanie produktami”.

### 3.1.1 Opis

Należy podać krótki opis funkcjonalności i wskazać, czy ma ona wysoki, średni czy niski priorytet.

Wyszukiwanie przy pomocy formularzy pracowników lub pomieszczeń (dokument: wyszukiwarka.html). Funkcjonalność ma wysoki priorytet.

### 3.1.2 Wymagania funkcjonalne

Należy wymienić szczegółowe wymagania funkcjonalne związane z tą funkcjonalnością. Są to możliwości oprogramowania wymagane, aby użytkownik mógł wykonać dany przypadek użycia. Opisać, jak produkt powinien reagować na przewidywane błędy oraz nieprawidłowe dane wejściowe i zachowania. Każde wymaganie funkcjonalne powinno zostać unikalnie oznaczone.

## 3.2 Wybór budynku (publiczne)

Należy podać nazwę funkcji w kilku słowach, na przykład „3.1 Zarządzanie produktami”.

### 3.2.1 Opis

Należy podać krótki opis funkcjonalności i wskazać, czy ma ona wysoki, średni czy niski priorytet.

Wybór budynku WI ZUT (dokument: mapa.html).

Funkcjonalność ma wysoki priorytet, jest niezbędna do przejścia do funkcjonalności ‘Plan’.

### 3.2.2 Wymagania funkcjonalne

1. Użytkownik wybiera budynek, którego plany chce obejrzeć.
2. Następnie klika w zdjęcie budynku lub pinezkę na mapie Leaflet i przenosi się na stronę wybranego budynku. (plan\_wi1.html lub plan\_wi2.html)

## 3.3 Plan budynku (publiczne)

Należy podać nazwę funkcji w kilku słowach, na przykład „3.1 Zarządzanie produktami”.

### 3.3.1 Opis

Należy podać krótki opis funkcjonalności i wskazać, czy ma ona wysoki, średni czy niski priorytet.

Użytkownik może przełączać plany pięter, które go interesują oraz wybrać konkretne pomieszczenie.

Funkcjonalność ma wysoki priorytet.

### 3.3.2 Wymagania funkcjonalne

1. Po kliknięciu w interesujący nas budynek, zostajemy przekierowani do strony, na której jest interaktywny plan wybranego budynku.
2. Następnie użytkownik wybiera piętro i klika na interesujące go pomieszczenie.
3. Po kliknięciu na pomieszczenie zostaną wyświetlone informacje o pomieszczeniu.

## 3.4 Dodawanie / edycja / kasowanie pomieszczenia (ograniczony dostęp)

Należy podać nazwę funkcji w kilku słowach, na przykład „3.1 Zarządzanie produktami”.

### 3.4.1 Opis

Należy podać krótki opis funkcjonalności i wskazać, czy ma ona wysoki, średni czy niski priorytet.

Z pozycji admina (zalogowany użytkownik) można dodać nowe pomieszczenia oraz edytować lub skasować istniejące.

Funkcjonalność ma średni priorytet.

### 3.4.2 Wymagania funkcjonalne

## 3.5 Import pracownika (ograniczony dostęp)

Należy podać nazwę funkcji w kilku słowach, na przykład „3.1 Zarządzanie produktami”.

### 3.5.1 Opis

Należy podać krótki opis funkcjonalności i wskazać, czy ma ona wysoki, średni czy niski priorytet.

Dodanie nowego pracownika z pozycji admina.

Funkcjonalność ma średni priorytet.

### 3.5.2 Wymagania funkcjonalne

## 3.6 Wgranie nowego planu

Należy podać nazwę funkcji w kilku słowach, na przykład „3.1 Zarządzanie produktami”.

### 3.6.1 Opis

Należy podać krótki opis funkcjonalności i wskazać, czy ma ona wysoki, średni czy niski priorytet.

Możliwość podmiany oraz importu planów budynków.

Funkcjonalność ma niski priorytet.

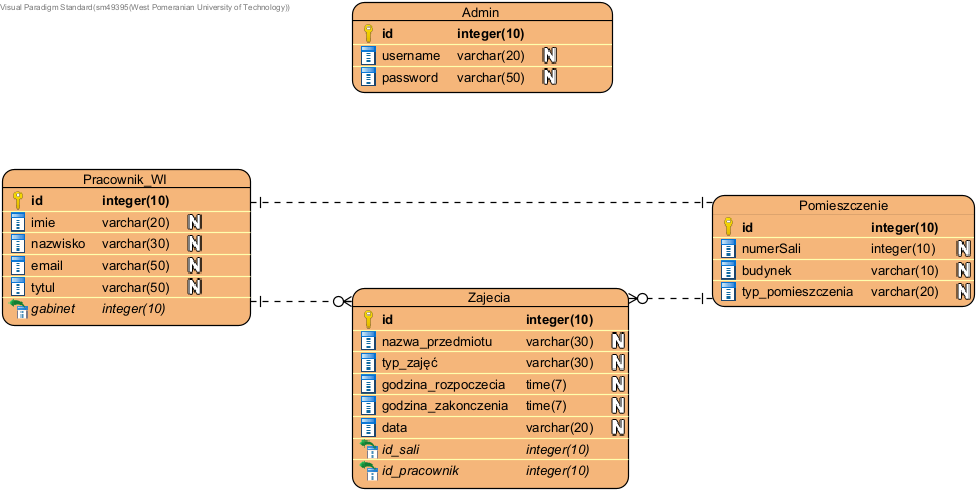
### 3.6.2 Wymagania funkcjonalne

# 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.



## 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.

System nie generuje raportów.

## 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.

* Dane pozyskiwane do bazy pobierane są z plan.zut.edu.pl.
* Dane przechowywane będą na uczelnianych serwerach zabezpieczonych kopiami zapasowymi w przypadku jakichkolwiek awarii, co zapewni integralność danych i zdolność funkcjonowania całego systemu w sposób bezawaryjny.
* Techniki wykonywania kopii zapasowych określone zostaną przez administratorów serwerowni gdzie będą przechowywane dane.
* Usuwanie danych będzie operacją wykonywaną jedynie z polecenie władz uczelni.

# 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.

## 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.

Brak

## 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.

System jest zgodny ze standardem HTML5. Zabezpieczenia są zapewniane przez szyfrowanie SSL podczas komunikacji przez Internet. System będzie również umożliwiał (adminom) zalogowanie się co pozwoli na wysyłanie zapytań i poleceń SQL.

# 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.

* **Łatwość uczenia się**: system jest łatwy w obsłudze dla nowych użytkowników, interfejs jest intuicyjny i zrozumiały.
* **Łatwość użytkowania**: system jest prosty w obsłudze, co pozwala użytkownikom szybko odnaleźć potrzebne im informacje. Czcionka jest duża i czytelna, aby nawet osoby ze słabszym wzrokiem nie miały problemu z odczytaniem zawartości.
* **Wydajność interakcji:**  system reaguje szybko na działania użytkowników, bez opóźnień i przestojów. Czas na odpowiedź z serwera jest ograniczony do 2 sekund.
* **Dostępność**: system jest dostępny dla wszystkich.

## 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.

* System ma działać bez przerw oraz żadne aktualizacje nie mogą przyczynić się do jej destabilizacji.
* System ma obsługiwać jednocześnie do 1000 użytkowników.
* System działa na popularnych przeglądarkach: [Chrome, Opera, Firefox, Edge itp.].
* Maksymalna odpowiedź na żądanie/zapytanie użytkownika to 2 sekundy.
* Aplikacja korzysta z obliczeń w chmurze, dlatego nie korzysta ze sprzętu klienta końcowego do działań obliczeniowo-zbiorczych.

## 6.3 Zabezpieczenia

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

* **Bezpieczeństwo danych**: System chroni przechowywane w nim dane przed utratą, uszkodzeniem, nieuprawnionym dostępem czy modyfikacją. W tym celu są stosowane techniki szyfrowania, backup`y oraz autoryzacja dostępu.
* **Bezpieczeństwo oprogramowania**: System jest zabezpieczony przed atakami hakerskimi, wirusami czy innymi zagrożeniami z sieci. Ponadto oprogramowanie jest aktualizowane, aby zapewnić stabilność i niezawodność działania.
* **Ochrona przed szkodami**: System ogranicza możliwość wystąpienia szkód, które mogą wyniknąć z jego użytkowania. Może to obejmować zabezpieczenia przed błędami użytkowników, awariami czy złym działaniem oprogramowania.

## 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.

* **Działania zapobiegawcze**: System informuje użytkowników o potencjalnie niebezpiecznych działaniach lub sytuacjach, a także udostępnia im narzędzia do ich uniknięcia. Może to obejmować ostrzeżenia, podpowiedzi czy instrukcje bezpieczeństwa.
* **Ochrona przed nieuprawnionym dostępem**: System blokuje lub uniemożliwia nieuprawnionym osobom dostęp do jego funkcji lub danych. W tym celu stosowane są metody autoryzacji i uwierzytelniania użytkowników.

## 6.5 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.

* **Skalowalność**: System jest zaprojektowany tak, aby był w stanie radzić sobie z coraz większą liczbą użytkowników i danych. Może on być łatwo rozbudowywany o nowe funkcje lub integrowany z innymi systemami.
* **Łatwość rozwoju**: System jest zbudowany z wykorzystaniem nowoczesnych technologii i dobrych praktyk programistycznych, co ułatwia jego rozwój i utrzymanie.
* **Elastyczność**: System jest elastyczny i umożliwia dostosowywanie go do różnych potrzeb i oczekiwań użytkowników. Możliwe jest również integrowanie go z innymi systemami lub dodawanie nowych funkcji.

# 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.

Jedynym językiem dostępnym w systemie jest język polski.

Czas w systemie jest ustawiony na

* UTC+01:00 (czas środkowoeuropejski /CET/ w okresie zimowym od października do marca)
* UTC+02:00 (czas środkowoeuropejski letni /CEST/, w okresie letnim od marca do października)

i nie podlega zmianie niezależnie od lokalizacji użytkownika systemu.

# 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.

Brak.

# Suplement A: Słownictwo

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

Wydział Informatyki Zachodniopomorskiego Uniwersytetu Technologicznego – WI ZUT.

# 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.

Brak, system jest wyjątkowo intuicyjny i prosty w użyciu.