Specyfikacja wymagań

Software Requirements Specification

# Wersja

|  |  |  |  |
| --- | --- | --- | --- |
| Wersja | Data | Autor | Zakres zmian |
| 0.1.2 | 2023-01-16 | Grzegorz Lach | Punkt 4. |

# 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

Niniejszy dokument opisuje podstawową wizję systemu ¿Donde?. Idea projektu narodziła się na wydziale Chemii, gdzie pracownicy mają przypisane do siebie swoje sale laboratoryjne i gabinety, natomiast nawet doświadczeni pracownicy WCh mają problem ze znalezieniem niektórych pomieszczeń, tym bardziej, że WCh ma aż 3 budynki! Podobnie WI ma 2 budynki, a pracownicy mają swoje gabinety. System powinien ułatwić lokalizowanie pracowników o określonej porze.

## Cel

Aplikacja webowa „Donde”, której celem jest lokalizacja pracowników wybranego wydziału ZUT o określonej porze dnia. W dodatku lokalizacja wybranych pomieszczeń w zadanym budynku infrastruktury ZUT wraz z ich danymi adresowanymi. Dokument ten jest przeznaczony dla programistów, product owner’a oraz dla wszystkich pozostałych zainteresowanych budową oraz funkcjonalnością aplikacji.

## 1.2 Przyjęte konwencje

Opisz tu przyjęte standardy I konwencje, znaczenie formatowania.

**Podmiot –** pracownik ZUT’u, bądź pomieszczenie w zadanym budynku

# Zakres projektu

Aplikacja jest przeznaczona dla studentów ZUT, pracowników ZUT oraz pozostałych osób zainteresowanych rozmieszczeniem pomieszczeń bądź, miejscem przebywania pracowników ZUT. Aplikacja ułatwi życie wyżej wymienionym grupą w szybszych i łatwiejszym znajdowaniem **podmiotów**.

Aplikacja pozwoli na wybór wydziału ZUT, w przypadku kiedy dany wydział posiada więcej niż 1 budynek pozwoli też na wybór interesującego budynku. W tym momencie będzie wyświetlona mapka podziału pomieszczeń, po kliknięciu danego pomieszczenia użytkownik otrzyma wszystkie informacje na jego temat.

## 1.4 Odwołania

* AI1-LAB10-swiz-donde

# 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

Jest to nowy produkt, który będzie działał na API w „plan lekcji ZUT”

## 

## 2.2 Klasy i charakterystyki użytkowników

Inni – Osoba nie zalogowana, posiada ograniczone funkcjonalności. Będzie w stanie jedynie przeglądać rozkład pomieszczeń wybranego wydziału ZUT.

Student – Posiada to co Inni. Jest to użytkownik zalogowany, który ma dodatkowo dostęp do informacji który pracownik ZUTU jest właścicielem danego pomieszczenia oraz gdzie aktualnie można znaleźć wybranego pracownika ZUT.

Pracownik ZUT – Posiada to co Student.

Administrator – Posiada to co Pracownik ZUT. Dodatkowo może modyfikować dane pomieszczeń, dodawać nowe plany oraz budynki.

## 2.3 Środowisko działania

Oprogramowanie będzie działać na terenie Polski szczególnie w miejscowości Szczecin gdzie znajduje się najwięcej użytkowników tej aplikacji.

*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.*

## 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

* Instalacja apache
* Instalacja MySql
* Połączenie z API plan zajęć ZUT

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.

# 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 Ekran startowy

### 3.1.1 Opis

Pierwszy ekran po uruchomieniu aplikacji. Zawiera odnośniki przekierowujące do kolejnych części systemu - WYSOKI

### 3.1.2 Wymagania funkcjonalne

### 3.1.2.1 Logowanie

### 3.1.2.1.1 Poprawne działanie

Użytkownik podaje poprawne dane logowania, login i hasło. Dzięki temu odblokowuje kolejne funkcjonalności systemu w zależności od tego na jakim poziomie użytkownik jest zalogowany.

### 3.1.2.1.2 Użytkownik podął złe dane/ brak danych

Wyświetlenie komunikatu o błędnym logowaniu. Pozostanie na danej stronie i pozwolenie na ponowne wysłanie danych.

### 3.1.2.1.3 Problem z połączeniem

Wyświetlenie komunikatu o problemie połącznia się z aplikacją. Pozostanie na danej stronie i pozwolenia na ponowną próbę zalogowania.

### 3.1.2.2 Przejście do sekcji MAPA

### 3.1.2.3 Przejście do sekcji PLAN

### 3.1.2.4 Przejście do sekcji SZUKAJ

### 3.1.2.5 Dodaj pracownika ZUT

### 3.1.2.6 Import pracowników

### 3.1.2.7 Edycja informacji o pomieszczeniach

### 3.1.2.8 Zarządanie budnkiem

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.

# 4. Wymagania dotyczące danych

Dane wejściowe:

-Mapy SVG planów pomieszczeń danych pięter

-Plan zajęć ZUT

-Baza pracowników ZUT

Dane wyjściowe:

-Tabela danych (Wydział, budynek, piętro, pomieszczenie, właściciel, aktualny, plan zajęć )

## 4.1 Logiczny model danych

## 4.2 Raporty

Brak

## 4.3 Pozyskiwanie, integralność przechowywanie I usuwanie danych

Dane będą przechowywane na zabezpieczonym serwerze uczelnianym uczelni ZUT. Aby zapewnić integralność i bezpieczeństwo danych będzie potrzebne tworzenie kopii zapasowych na osobnym urządzeniu fizycznym, w razie awarii/pożaru jednego z dwóch serwerów dane pozostaną bezpieczne na drugim. Z racji logicznego działania aplikacji nie potrzeba częstego wykonywania kopii danych, wystarczy wykonywać je raz w miesiącu. Dane te to użytkownicy którymi są studenci oraz pracownicy ZUT oraz budynki. Pierwsza grupa danych zazwyczaj zmienia się znacząco dopiero podczas rozpoczęcia nowego semestru, natomiast druga (budynki) jest sytuacją wyjątkową i dane te rzadko będą się zmieniać.

# 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 dot. szerokości ekranu:
  + 480px – desktop telefon
  + 786plx – desktop tablet
  + Reszta – wersja desktopowa
* Standardowe przyciski:
  + Powrót
* Konwencja komunikatów:
  + Posiada krótki tytuł np.: Błąd, Uwaga, Powiadomienie…
  + Wyświetlany na odpowiednim kolorze powiązanym z jego znaczeniem tzn.: komunikat błędu pojawi się na czerwonym tle, zwykłe powiadomienie na niebieskim itp.
* Dane wejściowe powinny mieć ograniczenie maksymalnie tak duże jak maksymalna wartość ustalona w bazie danych. Walidacja danych powinny odbyć się dwuetapowo. Wstępnie po stronie użytkownika podczas wypełniania formularzy, kolejne po stronie serwera.
* Dostępność:
  + Dla osób z problemami wzrokowymi: Większa czcionka, audiodeskrypcja
  + Dla osób zagranicznych: Wybór języka

## 5.2 Interfejsy programowe

Połączenie aplikacji z Planem zajęć:

Celem jest pobranie planu zajęć dla danego pomieszczenia, aby wyświetlić dane użytkownikowi. Dane przesyłane są w formacie JSON, więc trzeba będzie je zrzutować na wcześniej przygotowany model klasy.

## 5.3 Interfejsy sprzętowe

Aplikacja nie przewiduje takich interfejsów.

## 5.4 Interfejsy komunikacyjne

Aplikacja nie przewiduje takich interfejsów.

# 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

Wymagana jest responsywność aplikacji, aby zapewnić dostęp i wygodę w użyciu na każdym urządzeniu mobilnym.

Kolorystyka aplikacji powinna obejmować jedynie kolory takie jak: CMYK 100/0/95/20, 100/75/0/20, 0/0/0/80 oraz odcienie bieli i czerni.

Używane czcionki powinny być z rodziny: Franklin Gothic oraz jedynie w 4 wariantach rozmiarowych

## 6.2 Wydajność

Maksymalna liczba użytkowników korzystających z aplikacji powinna zależeć od ilości studentów. Czyli 2/3 ilości studentów jest to minimum, aby aplikacja działała wydajnie.

Aplikacja przewiduje, że cały ruch internetowy będzie pochodził z Polski.

Działanie aplikacji powinno być dynamiczne i każde zapytanie powinno zostać zrealizowane maksymalnie do 1 sekundy.

## 6.3 Zabezpieczenia

Najważniejsze dla nas jest zabezpieczenie danych wrażliwych, RODO, w tym celu dostęp do bazy danych jest ograniczony oraz kontrolowany. Każde zapytanie z którego aplikacja korzysta musi zostać zaprojektowane w sposób taki aby pobierać jedynie dane potrzebne, a nie wszystkie.

Sama aplikacja musi być zabezpieczona na wielu poziomach tak, aby żadne dane z niej nie zostały wykradzione. Weryfikacja poprzez login/ hasło. Szyfrowanie danych oraz samej aplikacji – minifikacja.

## 6.4 Bezpieczeństwo użytkowania

# 7. Wymagania internacjonalizacji i lokalizacji

Nie dotyczy

# 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.