

# Dokument wymagań funkcjonalnych

**Projekt:** *Robotovo Dashboard*

**Wersja:** 1.0

**Data:** 2024-11-25

**Autorzy:** Tomasz Petrykowski, Maciej Peta, Dorota Duda, Ewa Węglewska, Paweł Żabczyński

**Zatwierdzone przez:** Powyższy zespół, dr Witold Bołt

---

## 1. Wstęp

### 1.1 Cel dokumentu

Celem tego dokumentu jest przedstawienie wymagań funkcjonalnych i нефункциональных systemu zarządzania usługą dostarczania dronem, który umożliwi efektywne zarządzanie zamówieniami i dronami w firmie Robotovo.

### 1.2 Zakres projektu panelu pracownika

System ma zapewniać:

- Integrację z systemem głównym oraz bazą danych.
- Zarządzanie pracownikami.
- Zarządzanie zamówieniami.
- Zarządzanie urządzeniami.
- System zgłoszeń problemów.
- Powiadomienia dla użytkowników.

System **nie obejmuje** procesów potrzebnych do poprawnego świadczenia usługi (wyznaczanie trasy drona itp.) oraz dodawania zamówień do systemu. Aplikacja **nie zawiera też własnej bazy danych**, przekazuje i zbiera informacje do głównej bazy danych.

### 1.3 Odbiorcy

- Pracownicy supportu technicznego.
- Technicy oraz serwisanci dronów oraz komponentów potrzebnych do ich działania.
- Pracownicy końcowi: osoby pracujące w restauracjach, magazynach, punktach odbioru paczek

## 2. Opis systemu

### 2.1 Kontekst biznesowy

System umożliwi sprawne zarządzanie wypadkami technicznymi, ułatwi serwis urządzeń oraz umożliwi ciągły kontakt między pracownikami różnych podmiotów. Usprawni również wysyłanie zamówienia do klientów końcowych.

### 2.2 Cele projektu

1. Skrócenie czasu napraw usterek urządzeń.
  2. Poprawa komunikacji między pracownikami różnych podmiotów.
  3. Możliwość sprawnej i szybkiej pomocy w razie problemów w trakcie dostarczania zamówienia.
- 

## 3. Wymagania funkcjonalne

### 3.1 Integracja z bazą danych

**FR-001:** Aplikacja musi łączyć się z centralną bazą danych przez API oraz umożliwiać automatyczną synchronizację informacji o zamówieniach, dronach i zgłoszeniach.

### 3.2 Zarządzanie użytkownikami

**FR-002:** System musi umożliwiać logowanie użytkowników za pomocą adresu e-mail i hasła.

**FR-003:** Administratorzy mogą tworzyć, edytować i usuwać konta użytkowników.

**FR-004:** System powinien wspierać role użytkowników:

- **Administrator:** pełny dostęp.
- **Pracownik supportu:** Dostęp do:
  - zamówień,
  - listy zgłoszeń (ticketów)
  - listy aktywnych dronów,
  - kontaktu do innych podmiotów
- **Pracownik techniczny:** Dostęp do:
  - listy wszystkich dronów,
  - listy zgłoszeń (ticketów)
  - kontaktu do innych podmiotów-
- **Pracownik końcowy:** Dostęp do:
  - listy zamówień ze swojego punktu odbioru (restauracja, magazyn)

- kontaktu do pracowników supportu

### 3.3 Zarządzanie zamówieniami

**FR-005:** Pracownik końcowy może szybko potwierdzić gotowość zamówienia, jak i gotowość drona do odlotu:

**FR-006:** Pracownik supportu może podejrzeć listę zamówień przypisanych do danego punktu odbioru aby móc sprawnie pomóc punktom odbioru w razie potencjalnych problemów.

### 3.4 Zgłoszenia

**FR-007:** Pracownicy końcowi powinni mieć do dyspozycji opcję zgłoszenia problemu technicznego związanego z urządzeniem lub działaniem systemu.

**FR-008:** Pracownicy supportu mogą zarządzać listą zgłoszeń im przydzielonych.

**FR-009:** Pracownicy supportu w ramach zgłoszenia mogą komunikować się ze zgłaszającym lub przesłać zgłoszenie do odpowiedniego działu, np. technicznego. Mogą też zamknąć dane zgłoszenie jako rozwiązane.

**FR-010:** Technicy również mają dostęp do zarządzania listą przydzielonych im zgłoszeń.

### 3.5 Komunikacja między pracownikami

**FR-011:** Pracownik końcowy ma do dyspozycji odpowiedni moduł czatu w celu komunikacji z pracownikami supportu

**FR-012:** Pracownik techniczny ma do dyspozycji moduł czatu w celu komunikacji oraz konsultacji z pracownikami supportu.

**FR-013:** Pracownik supportu posiada odpowiedni moduł komunikacji pogrupowany na obsługiwane zgłoszenia oraz moduł czatu, który pozwala na konsultację z innymi pracownikami.

### 3.6 Monitorowanie dronów

**FR-014:** Pracownik supportu oraz pracownik techniczny ma możliwość sprawdzenia lokalizacji drona oraz jego statusu.

**FR-015:** Pracownik supportu oraz pracownik techniczny mogą zmienić status urządzenia na:

- Sprawne
- Zajęte
- Niesprawne

- W trakcie naprawy

**FR-016:** Historia zmian statusów powinna być widoczna w szczegółach parametrów urządzenia.

### 3.7 Przejęcie kontroli manualnej nad dronem

**FR-017:** Aplikacja ma łączyć się z dronem poprzez API, przysyłać obraz z kamery oraz wysyłać powiadomienia w razie problemów z urządzeniem.

**FR-018:** System ma umożliwiać przejęcie zdalnej kontroli nad zachowaniem drona w razie problemów pogodowych lub innej usterki.

### 3.8 Powiadomienia

**FR-019:** System wysyła alerty do systemów do pracowników o nowym zamówieniu / zgłoszeniu / usterce / wiadomości.

**FR-020:** Każdy z pracowników otrzymuje powiadomienie w systemie o zmianie statusu zgłoszenia / zamówienia / usterki.

### 3.9 Wyświetlanie historii

**FR-021:** Aplikacja wyświetla historię zgłoszeń, usterek oraz statusów.

---

## 4. Wymagania niefunkcjonalne

- **NFR-001 - Dostępność:** System musi działać 24/7 z maksymalnym czasem przestoju 1h miesięcznie.
- **NFR-002 - Wydajność:** System powinien obsługiwać do 200 jednoczesnych użytkowników.
- **NFR-003 - Bezpieczeństwo:** Dane zamówień, pracowników oraz urządzeń muszą być przechowywane w formie zaszyfrowanej oraz powinny być chronione przed nieautoryzowanym dostępem.
- **NFR-004 - Wygoda użytkowania:** Interfejs aplikacji musi być intuicyjny i łatwy w obsłudze oraz dostosowany do potrzeb różnych grup użytkowników.

---

## 5. Załączniki

- **Glosariusz:**

- *Aplikacja / System*: System pozwalający pracownikom zarządzać zamówieniami, urządzeniami oraz zgłoszeniami.
- *Urządzenie*: Dron lub inne urządzenie potrzebne do jego pracy, np. stacja ładowania.
- *Zamówienie*: transakcja wprowadzona do systemu przez system.
- *Status urządzenia*: aktualny stan techniczny urządzenia.
- *Usterka*: Błąd w działaniu urządzenia.

- **Lista skrótów:**

- *FR*: Wymaganie funkcjonalne (Functional Requirement).
- *NFR*: Wymaganie нефunkcjonalne (Non-Functional Requirement).
- *API*: Interfejs aplikacji zewnętrznej, pozwalający na łączenie się z nią (Application User Interface).