

# OutsideJob

## Specyfikacja Wymagań Funkcjonalnych i Niefunkcjonalnych

### Aplikacja do Zarządzania Raportami Terenowymi

Dokumentacja Projektowa

1 grudnia 2025

## Spis treści

<b>1 Wstęp</b>	<b>4</b>
1.1 Cel projektu . . . . .	4
1.2 Zakres aplikacji . . . . .	4
1.3 Adresaci i interesariusze . . . . .	4
1.4 Dokumenty referencyjne . . . . .	4
<b>2 Wymagania funkcjonalne</b>	<b>4</b>
2.1 REQ-F1: Autentykacja i logowanie . . . . .	4
2.2 REQ-F2: Zapamiętanie loginu i hasła . . . . .	5
2.3 REQ-F3: Lista zadań (Tasks) po logowaniu . . . . .	5
2.4 REQ-F4: Filtrowanie listy zadań . . . . .	6
2.5 REQ-F5: Szczegóły zadania . . . . .	6
2.6 REQ-F6: Rozpoczęcie dojazdu . . . . .	6
2.7 REQ-F7: Rozpoczęcie pracy nad zadaniem . . . . .	7
2.8 REQ-F8: Wstrzymanie zadania (SUSPENDED) . . . . .	7
2.9 REQ-F9: Rozliczenie pracy (Completion) . . . . .	8
2.10 REQ-F10: Dodawanie zadania (jeśli wymaga klient) . . . . .	8
2.11 REQ-F11: Logout . . . . .	8
2.12 REQ-F12: Dodawanie załączników (Zdjęcia, Notatki) . . . . .	9
2.13 REQ-F13: Interfejs użytkownika (UI Layout) . . . . .	9
2.14 REQ-F14: Synchronizacja danych . . . . .	9
<b>3 Wymagania niefunkcjonalne</b>	<b>10</b>
3.1 NFR-1: Wydajność i skalowalność . . . . .	10
3.2 NFR-2: Bezpieczeństwo . . . . .	10
3.3 NFR-3: Dostępność i niezawodność . . . . .	10
3.4 NFR-4: Offline-first . . . . .	10
3.5 NFR-5: Kompatybilność . . . . .	11
3.6 NFR-6: Responsywność i UX . . . . .	11
3.7 NFR-7: Observability i monitoring . . . . .	11

<b>4 Przypadki użycia (Use Cases)</b>	<b>11</b>
4.1 UC-1: Zalogowanie pracownika . . . . .	11
4.2 UC-2: Przeglądanie listy zadań . . . . .	12
4.3 UC-3: Rozpoczęcie pracy nad zadaniem . . . . .	13
4.4 UC-4: Wstrzymanie zadania . . . . .	13
4.5 UC-5: Rozliczenie pracy . . . . .	14
<b>5 Scenariusze testowe</b>	<b>14</b>
5.1 ST-1: Logowanie online . . . . .	14
5.2 ST-2: Auto-logowanie . . . . .	14
5.3 ST-3: Praca offline . . . . .	15
5.4 ST-4: Filtrowanie zadań . . . . .	15
5.5 ST-5: Wstrzymanie z powodem . . . . .	15
5.6 ST-6: Rozliczenie z zdjęciami (offline + online) . . . . .	16
<b>6 Model domenowy i słownik pojęć</b>	<b>16</b>
6.1 Kluczowe terminy . . . . .	16
<b>7 Architektura systemu</b>	<b>17</b>
7.1 Komponenty systemu . . . . .	17
7.2 Przepływ danych . . . . .	17
<b>8 Model danych (ERD)</b>	<b>17</b>
8.1 Tabela Users . . . . .	17
8.2 Tabela Tasks . . . . .	18
8.3 Tabela Subtasks . . . . .	18
8.4 Tabela Reports . . . . .	19
8.5 Tabela Attachments . . . . .	19
8.6 Tabela Suspensions . . . . .	19
<b>9 Interfejs użytkownika (UI/UX)</b>	<b>19</b>
9.1 Główne ekrany . . . . .	19
9.2 Wytyczne designu . . . . .	20
<b>10 Integracje</b>	<b>20</b>
10.1 Firebase (opcjonalnie) . . . . .	20
<b>11 Plan testów</b>	<b>20</b>
11.1 Testy funkcjonalne . . . . .	20
11.2 Testy offline . . . . .	21
11.3 Testy wydajności . . . . .	21
11.4 Testy bezpieczeństwa . . . . .	21
<b>12 Wytyczne kodowania i konwencje</b>	<b>21</b>
12.1 Język programowania . . . . .	21
12.2 Struktura projektu . . . . .	22
12.3 Konwencje nazewnictwa . . . . .	22
12.4 Commits . . . . .	22

<b>13 Harmonogram wdrożenia</b>	<b>22</b>
13.1 Fazy projektu . . . . .	22
13.2 Kamienie milowe . . . . .	23
<b>14 Podsumowanie</b>	<b>23</b>
14.1 Dalsze kroki . . . . .	23

## 1 Wstęp

### 1.1 Cel projektu

OutsideJob to platforma mobilna przeznaczona dla firm posiadających floty serwisantów i pracowników terenowych. Celem aplikacji jest umożliwienie prostego i jednolitego prowadzenia raportów z podjętych zleceń (zadań) bezpośrednio z terenu, z możliwością offline-first i automatycznej synchronizacji po powrocie do internetu.

### 1.2 Zakres aplikacji

- Zarządzanie listą przydzielonych zadań (Tasks)
- Tworzenie szczegółowych raportów z wykonanych zleceń
- Wygenerowanie raportów w formatach wymaganych przez klienta (PDF, JSON, Excel itp.)
- Praca offline z możliwością synchronizacji danych
- Filtrowanie i wyszukiwanie zadań
- Zarządzanie załącznikami (zdjęcia, dokumenty)
- Kontrola dostępu poprzez logowanie

### 1.3 Adresaci i interesariusze

- **Pracownicy terenowi** (serwisanci, elektrycy, hydraulicy itp.) – główni użytkownicy aplikacji
- **Kierownicy pracowników** – mogą przydzielać zadania
- **Administratorzy OutsideJob** – zarządzają użytkownikami i systemem
- **Firmy-klienci** – otrzymują raporty w wymaganych formatach

### 1.4 Dokumenty referencyjne

- Architektura Synchronizacji Offline (osobny dokument)
- API REST (specyfikacja endpoints)
- Wytyczne UX/UI (wireframy/mockupy)

## 2 Wymagania funkcjonalne

### 2.1 REQ-F1: Autentykacja i logowanie

**Opis:** Użytkownik musi się zalogować do aplikacji za pomocą adresu e-mail i hasła otrzymanego od administratora OutsideJob.

- REQ-F1.1: Ekran logowania zawiera pola: “Email”, “Hasło”
- REQ-F1.2: Po wpisaniu niepoprawnych danych wyświetlane jest komunikat błędu
- REQ-F1.3: Po zalogowaniu aplikacja synchronizuje listę zadań z serwerem

**Priorytet:** KRYTYCZNE

**Aktor:** Pracownik terenowy

---

## 2.2 REQ-F2: Zapamiętanie loginu i hasła

**Opis:** Użytkownik przy pierwszym logowaniu ma możliwość zapamiętania loginu i hasła dla automatycznego zalogowania przy kolejnych uruchomieniach aplikacji.

- REQ-F2.1: Checkbox “Zapamiętaj mnie” dostępny na ekranie logowania
- REQ-F2.2: Dane uwierzytelniające przechowywane lokalnie w **Android Keystore** (bezpieczne)
- REQ-F2.3: Przy wylogowaniu dane usuwane z lokalnego magazynu
- REQ-F2.4: Logowanie na innym urządzeniu powoduje wylogowanie na poprzednim

**Priorytet:** WYSOKI

**Aktor:** Pracownik terenowy

---

## 2.3 REQ-F3: Lista zadań (Tasks) po logowaniu

**Opis:** Po zalogowaniu użytkownik otrzymuje zsynchronizowaną listę zadań przydzielonych do niego.

- REQ-F3.1: Lista zadań pobierana z serwera przy logowaniu
- REQ-F3.2: Zadania sortowane po dacie utworzenia (najnowsze u góry)
- REQ-F3.3: Każde zadanie wyświetla: tytuł, krótki opis, status, datę terminu
- REQ-F3.4: Ikona statusu (kolor/ikonka) wskazuje stan: TODO, IN\_PROGRESS, COMPLETED, SUSPENDED, OVERDUE
- REQ-F3.5: Możliwość kliknięcia w zadanie dla wyświetlenia całego zadania z przyciskami rozpoczęt/rozpocznij dojazd rozpoczęt/prace

**Priorytet:** KRYTYCZNE

**Aktor:** Pracownik terenowy

---

## 2.4 REQ-F4: Filtrowanie listy zadań

**Opis:** Użytkownik ma możliwość filtrowania zadań między kategoriami.

- REQ-F4.1: Filtry dostępne w górnym menu (po prawej): ALL, TODO, IN\_PROGRESS, COMPLETED, SUSPENDED, OVERDUE
- REQ-F4.2: Wybranie filtru powoduje przeliczenie listy lokalnie
- REQ-F4.3: Aktualny filtr wyświetlany w nagłówku ekranu
- REQ-F4.4: Preferencja ostatniego wybranego filtru przechowywana lokalnie
- REQ-F4.5: Przy każdorazowym uruchomieniu aplikacji pracownik widzi filtr TODO
- REQ-F4.6: Zadanie które nigdy nie wyło wyświetcone (pracownik nigdy jeszcze w niego nie kliknął nie widział szczegółów wyróżnia się kolorem jest jako new) ale stan pozostaje TODO

**Priorytet:** WYSOKI

**Aktor:** Pracownik terenowy

---

## 2.5 REQ-F5: Szczegóły zadania

**Opis:** Po wybraniu zadania z listy użytkownik widzi ekran ze szczegółami i możliwościami wykonania.

- REQ-F5.1: Ekran zawiera: tytuł, pełny opis, termin, przydzielony pracownik, lokalizacja, załączniki
- REQ-F5.2: Przycisk “Rozpocznij dojazd” – zmienia status na IN\_TRANSIT
- REQ-F5.3: Przycisk “Rozpocznij pracę” – zmienia status na IN\_PROGRESS
- REQ-F5.4: Przycisk “Wstrzymaj” – zmienia status na SUSPENDED (wymaga opisu) tylko w stanie IN\_PROGRESS lub IN\_TRANSIT
- REQ-F5.5: Przycisk “Wznów” – powrót do IN\_PROGRESS (dostępny tylko ze stanu SUSPENDED)

**Priorytet:** KRYTYCZNE

**Aktor:** Pracownik terenowy

---

## 2.6 REQ-F6: Rozpoczęcie dojazdu

**Opis:** Pracownik może rozpoczęć dojazd do lokalizacji zadania, zmienia status na IN\_TRANSIT.

- REQ-F6.1: Przycisk “Rozpocznij dojazd” zmienia status z TODO na IN\_TRANSIT

**Priorytet:** WYSOKI

**Aktor:** Pracownik terenowy

---

## 2.7 REQ-F7: Rozpoczęcie pracy nad zadaniem

**Opis:** Pracownik rozpoczyna pracę nad zadaniem, zmienia status na IN\_PROGRESS, otwiera się checklist zadań.

- REQ-F7.1: Przycisk “Rozpocznij pracę” zmienia status na IN\_PROGRESS
- REQ-F7.2: Wyświetla się lista podzadań (cały formularz raportowania) z checkboxami
- REQ-F7.3: Każde podzadanie może mieć opis, typ (checkbox, tekstowe pole, zdjęcie)
- REQ-F7.4: Jedno kliknięcie na podzadanie otwiera klawiaturę dla szybkiego wprowadzenia tekstu
- REQ-F7.5: Pracownik nie może mieć więcej niż jedno zlecenie o statusie IN\_PROGRESS oraz IN\_TRANSIT

**Priorytet:** KRYTYCZNE

**Aktor:** Pracownik terenowy

---

## 2.8 REQ-F8: Wstrzymanie zadania (SUSPENDED)

**Opis:** Pracownik może wstrzymać zadanie (np. oczekiwanie na części, przerwa), wymaga uzasadnienia.

- REQ-F8.1: Przycisk “Wstrzymaj” dostępny ze stanu IN\_PROGRESS lub IN\_TRANSIT
- REQ-F8.2: Formularz zawiera pola dostosowane do konkretnego klienta (np. wymiana przedmiotów, wykonanie zdjęć)
- REQ-F8.3: Wyświetla się dialog z polem tekstowym “Powód wstrzymania”
- REQ-F8.4: Pole obowiązkowe – bez powodu nie można wstrzymać
- REQ-F8.5: Status zmienia się na SUSPENDED
- REQ-F8.6: Przycisk “Wznów pracę” powraca do IN\_PROGRESS
- REQ-F8.7: Historia wstrzymań przechowywana w raporcie

**Priorytet:** WYSOKI

**Aktor:** Pracownik terenowy

---

## 2.9 REQ-F9: Rozliczenie pracy (Completion)

**Opis:** Pracownik kończy pracę, podaje ostateczny raport, zmienia status na COMPLETED.

- REQ-F9.1: Przycisk “Zakończ pracę” sprawdza czy wszystkie obowiązkowe pola zostały uzupełnione jeżeli tak to zamyka zlecenie i zmienia status na COMPLETED
- REQ-F9.2: PO zakończeniu pracy brak możliwości edycji jakichkolwiek pól
- REQ-F9.3: Po potwierdzeniu – upload raportów/zdjęć i zmiana statusu na COMPLETED
- REQ-F9.4: Zadanie trafia do historii, nie można go edytować

**Priorytet:** KRYTYCZNE

**Aktor:** Pracownik terenowy

---

## 2.10 REQ-F10: Dodawanie zadania (jeśli wymaga klient)

**Opis:** Pracownik (lub kierownik) może dodać nowe zadanie do swojej listy.

- REQ-F10.1: Zakładka “+ Dodaj zadanie” w navigation drawer
- REQ-F10.2: Formularz zawiera: tytuł, opis, lokalizacja, termin, przydzielenie do pracownika
- REQ-F10.3: Jeśli pracownik – zawsze przydzielane sobie
- REQ-F10.4: Jeśli kierownik – może wybrać pracownika z listy
- REQ-F10.5: Zadanie tworzone TYLKO online
- REQ-F10.6: Po utworzeniu – serwer generuje ID, synchronizuje się na urządzenia

**Priorytet:** ŚREDNI

**Aktor:** Pracownik terenowy, Kierownik

---

## 2.11 REQ-F11: Logout

**Opis:** Użytkownik może się wylogować z aplikacji.

- REQ-F11.1: Opcja “Wyloguj” w navigation drawer (dół, po lewej)
- REQ-F11.2: Po wylogowaniu – usuwane zapamiętane dane logowania (jeśli były)
- REQ-F11.3: Powrót do ekranu logowania
- REQ-F11.4: Lokalne dane zadań pozostają (offline-first), ale logowanie na innym urządzeniu te sesję wyłącza

**Priorytet:** WYSOKI

**Aktor:** Pracownik terenowy

---

## 2.12 REQ-F12: Dodawanie załączników (Zdjęcia, Notatki)

**Opis:** Pracownik może dodawać zdjęcia, notatki głosowe, dokumenty do zadania.

- REQ-F12.1: Przycisk “+ Dodaj zdjęcie” w ekranie pracy
- REQ-F12.2: Możliwość robienia zdjęcia bezpośrednio lub wybrania z galerii
- REQ-F12.3: Zdjęcia przechowywane lokalnie, uploaded przy zatwierdzeniu zadania
- REQ-F12.4: Pole notatek dla każdego podzadania
- REQ-F12.5: Notatki głosowe (opcjonalnie) – nagranie i transkrypcja

**Priorytet:** WYSOKI

**Aktor:** Pracownik terenowy

---

## 2.13 REQ-F13: Interfejs użytkownika (UI Layout)

**Opis:** Struktura ekranów i nawigacji aplikacji.

- REQ-F13.1: Navigation Drawer po lewej stronie (“Tasks”, “+ Add Task”, “Wyloguj”)
- REQ-F13.2: Góra ekran - toolbar z: ikoną menu (hamburger), tytułem, ikoną filtrów
- REQ-F13.3: Menu filtrów po prawej w toolbarze (ALL, TODO, IN\_PROGRESS, COMPLETED, SUSPENDED, OVERDUE)
- REQ-F13.4: Bottom sheet z szybkimi akcjami (jeśli potrzebne)
- REQ-F13.5: FAB (Floating Action Button) “+” dla szybkiego dodania zadania

**Priorytet:** ŚREDNI

**Aktor:** Pracownik terenowy

---

## 2.14 REQ-F14: Synchronizacja danych

**Opis:** Dane synchronizują się automatycznie i ręcznie.

- REQ-F14.1: Automatyczna synchronizacja co 15 minut (jeśli są zmiany)
- REQ-F14.2: Ręczna synchronizacja – przyciągnięcie listy w dół (pull-to-refresh)
- REQ-F14.3: Po powrocie internetu – automatyczna synchronizacja pending zmian
- REQ-F14.4: Powiadomienie o sukcesie/błędzie synchronizacji
- REQ-F14.5: Brak konfliktów (zadania zakończone niezmienne)

**Priorytet:** KRYTYCZNE

**Aktor:** System

---

### 3 Wymagania niefunkcjonalne

#### 3.1 NFR-1: Wydajność i skalowalność

- NFR-1.1: Minimalna obsługa 50 użytkowników równolegle
  - NFR-1.2: Czas ładowania listy zadań: < 2 sekundy (gdy online)
  - NFR-1.3: Czas otwarcia szczegółów zadania: < 1 sekunda
  - NFR-1.4: Maksymalnie 100 zadań na liście bez lag'ów
  - NFR-1.5: Architektura skalowalna do 1000+ użytkowników w przyszłości
- 

#### 3.2 NFR-2: Bezpieczeństwo

- NFR-2.1: Cała komunikacja przez HTTPS (TLS 1.2+)
  - NFR-2.2: Hasła przechowywane haszowane (bcrypt/Argon2) na serwerze
  - NFR-2.3: Dane zapamiętane na urządzeniu w Android Keystore (szylfowane)
  - NFR-2.4: Logowanie z poziomu administratora możliwe dla audytu
  - NFR-2.5: Raporty zawierają dane pełne tylko dla autoryzowanych osób
- 

#### 3.3 NFR-3: Dostępność i niezawodność

- NFR-3.1: Wysoka dostępność (HA) 24/7 – SLA 99.5%
  - NFR-3.2: Backend w architekturze multi-region (ang. failover)
  - NFR-3.3: Backup danych co 6 godzin
  - NFR-3.4: Recovery Time Objective (RTO): < 1 godzina
  - NFR-3.5: Recovery Point Objective (RPO): < 15 minut
- 

#### 3.4 NFR-4: Offline-first

- NFR-4.1: Wszystkie dane przechowywane lokalnie w Room (SQLite)
  - NFR-4.2: Pracownik może pracować 100% offline
  - NFR-4.3: Synchronizacja dwukierunkowa (push/pull)
  - NFR-4.4: Brak konfliktów dla zamkniętych zadań
  - NFR-4.5: Offline-first = dane najpierw lokalnie, potem serwer
-

### 3.5 NFR-5: Kompatybilność

- NFR-5.1: Android 7.0 (API 24) minimum
  - NFR-5.2: Obsługa ekranów od 4.5” do 6.7”
  - NFR-5.3: Orientacja portrait i landscape
  - NFR-5.4: Obsługa różnych DPI (mdpi, hdpi, xhdpi, xxhdpi)
  - NFR-5.5: Testowanie na minimalnie 5 popularnych modelach urządzeń
- 

### 3.6 NFR-6: Responsywność i UX

- NFR-6.1: Szybka reakcja na interakcje użytkownika (< 300ms)
  - NFR-6.2: Animacje płynne (60 FPS)
  - NFR-6.3: Material Design 3 lub nowszy
  - NFR-6.4: Dostępność (A11y) – obsługa czytników ekranu
  - NFR-6.5: Kontrast tekstu minimum 4.5:1
- 

### 3.7 NFR-7: Observability i monitoring

- NFR-7.1: Centralne logowanie (np. ELK, Datadog)
  - NFR-7.2: Monitoring wydajności backend'u (APM)
  - NFR-7.3: Alerty na błędy i anomalie
  - NFR-7.4: Dashboards dla: liczba aktywnych użytkowników, czas odpowiedzi API, błędy
- 

## 4 Przypadki użycia (Use Cases)

### 4.1 UC-1: Zalogowanie pracownika

**Aktor:** Pracownik terenowy

**Warunki początkowe:** Aplikacja zainstalowana, użytkownik posiada e-mail i hasło od administratora

**Główny przepływ:**

1. Pracownik uruchamia aplikację
2. System wyświetla ekran logowania

3. Pracownik wpisuje e-mail i hasło
4. Pracownik opcjonalnie zaznacza “Zapamiętaj mnie”
5. Pracownik kliką “Zaloguj”
6. System łączy się z serwerem i weryfikuje dane
7. Po sukcesie – pobiera listę zadań
8. Aplikacja wyświetla listę Tasks

**Alternatywne przepływy:**

- A1: Niepoprawne hasło – wyświetlany komunikat błędu, pracownik może spróbować ponownie
- A2: Brak internetu – jeśli były poprzednie dane zapamiętane, system zaloguje offline
- A3: E-mail nie istnieje – komunikat “Użytkownik nie znaleziony”

**Warunki końcowe:** Pracownik zalogowany, lista zadań widoczna

---

## 4.2 UC-2: Przeglądanie listy zadań

**Aktor:** Pracownik terenowy**Warunki początkowe:** Pracownik zalogowany**Główny przepływ:**

1. System wyświetla listę wszystkich przydzielonych zadań
2. Pracownik widzi: tytuł, krótki opis, status, datę terminu
3. Pracownik może przewijać listę
4. Pracownik kliką na jedno z zadań
5. System otwiera szczegóły zadania

**Alternatywne przepływy:**

- A1: Brak zadań – wyświetlany komunikat “Brak przydzielonych zadań”
- A2: Pracownik filtry zadania (TODO, COMPLETED, SUSPENDED)
- A3: Pull-to-refresh – ręczna synchronizacja z serwerem

**Warunki końcowe:** Lista zadań wyświetlona lub szczegóły zadania otwarte

---

### 4.3 UC-3: Rozpoczęcie pracy nad zadaniem

**Aktor:** Pracownik terenowy

**Warunki początkowe:** Pracownik wybrał zadanie ze statusem TODO lub IN\_TRANSIT

**Główny przepływ:**

1. System wyświetla szczegóły zadania
2. Pracownik klika “Rozpocznij pracę”
3. Status zmienia się na IN\_PROGRESS
4. System wyświetla listę podzadań (checklist)
5. Pracownik wykonuje podzadania, zaznacza checkboxy
6. Pracownik dodaje notatki/zdjęcia
7. Po zakończeniu – pracownik klika “Zakończ pracę”

**Alternatywne przepływy:**

- A1: Pracownik wstrzymuje zadanie – przycisk “Wstrzymaj” z uzasadnieniem
- A2: Pracownik pracuje offline – zmiany przechowywane lokalnie

**Warunki końcowe:** Status zmieniony na IN\_PROGRESS, lub SUSPENDED, lub COMPLETED

---

### 4.4 UC-4: Wstrzymanie zadania

**Aktor:** Pracownik terenowy

**Warunki początkowe:** Pracownik w stanie IN\_PROGRESS

**Główny przepływ:**

1. Pracownik klika “Wstrzymaj”
2. System wyświetla dialog z polem “Powód wstrzymania”
3. Pracownik wpisuje powód (obowiązkowe)
4. Pracownik klika “Wstrzymaj”
5. Status zmienia się na SUSPENDED
6. System zapisuje zmianę lokalnie, odkłada do synchronizacji

**Warunki końcowe:** Status SUSPENDED, powód wstrzymania zapisany

---

## 4.5 UC-5: Rozliczenie pracy

**Aktor:** Pracownik terenowy

**Warunki początkowe:** Pracownik w stanie IN\_PROGRESS, praca skończona

**Główny przepływ:**

1. Pracownik kliką “Zakończ pracę”
2. System sprawdza formularz raportu (dostosowany do klienta) czy zawiera wszystkie obowiązkowe punkty
3. Pracownik kliką “Potwierdź”/“Wyślij”
4. System uploaduje załączniki (jeśli online) i zmienia status na COMPLETED
5. Powiadomienie “Zadanie zostało rozliczone”

**Alternatywne przepływy:**

- A1: Offline – plik raportów zapisany lokalnie, upload po powrocie internetu
- A2: Brak wymaganych pól – komunikat błędu, pracownik uzupełnia

**Warunki końcowe:** Status COMPLETED, raport zapisany na serwerze, zadanie niezmienne

---

## 5 Scenariusze testowe

### 5.1 ST-1: Logowanie online

1. Pracownik uruchamia aplikację
  2. Wpisuje e-mail i hasło
  3. Zaznacza “Zapamiętaj mnie”
  4. Kliką “Zaloguj”
  5. ✓ Aplikacja zalogowana, lista zadań pobrana
- 

### 5.2 ST-2: Auto-logowanie

1. Pracownik w poprzedniej sesji zaznaczył “Zapamiętaj mnie”
  2. Pracownik uruchamia aplikację (2. raz)
  3. ✓ Aplikacja automatycznie zalogowana bez wpisywania danych
-

### 5.3 ST-3: Praca offline

1. Pracownik zalogowany, ma listę zadań
  2. Wyłącz internet (Airplane Mode)
  3. Pracownik otwiera zadanie
  4. Zmienia status na IN\_PROGRESS
  5. Dodaje checkboxy, notatki, zdjęcia
  6. Kończy zadanie (COMPLETED)
  7. ✓ Wszystkie operacje działają offline
  8. Włącz internet
  9. ✓ Zmiany automatycznie synchronizują się
- 

### 5.4 ST-4: Filtrowanie zadań

1. Pracownik w liście zadań
  2. Kliką filtr “COMPLETED”
  3. ✓ Lista wyświetla tylko ukończone zadania
  4. Kliką filtr “SUSPENDED”
  5. ✓ Lista wyświetla tylko wstrzymane zadania
- 

### 5.5 ST-5: Wstrzymanie z powodem

1. Pracownik w IN\_PROGRESS
  2. Kliką “Wstrzymaj”
  3. Dialog pojawia się
  4. Pracownik NIE wpisuje powodu i kliką “Wstrzymaj”
  5. ✓ System wyświetla błąd “Pole obowiązkowe”
  6. Pracownik wpisuje powód
  7. ✓ Status zmienia się na SUSPENDED
-

## 5.6 ST-6: Rozliczenie z zdjęciami (offline + online)

1. Pracownik offline
  2. Kończy pracę, kliką “Zakończ pracę”
  3. Dodaje opisany raport, wymagane zdjęcia
  4. Kliką “Potwierdź”
  5. ✓ Raport zapisany lokalnie, status COMPLETED
  6. Włącz internet
  7. ✓ Zdjęcia i raport automatycznie uploadują się
- 

# 6 Model domenowy i słownik pojęć

## 6.1 Kluczowe terminy

- **Task (Zadanie)** – jednostka pracy przydzielona pracownikowi, zawiera opis, deadline, lokalizację
  - **Subtask (Podzadanie)** – element checklist'u w ramach Task'u
  - **Worker (Pracownik)** – użytkownik aplikacji, osoba terenowa
  - **Manager (Kierownik)** – przełożony pracownika, może przydzielać zadania
  - **Administrator** – osoba z OutsideJob, zarządza użytkownikami i systemem
  - **Report (Raport)** – dokument podsumowujący wykonane zadanie (PDF, JSON, Excel)
  - **Status** – stan zadania: TODO, IN\_TRANSIT, IN\_PROGRESS, SUSPENDED, COMPLETED
  - **Synchronizacja** – dwukierunkowa wymiana danych między urządzeniem a serwerem
  - **Offline-first** – dane najpierw przechowywane lokalnie, synchronizacja gdy internet dostępny
  - **Sesja** – okres zalogowania użytkownika, wygasająca po wylogowaniu lub zalogowaniu użytkownika na innym urządzeniu bądź po 7 dniach braku aktywności
-

## 7 Architektura systemu

### 7.1 Komponenty systemu

Android App (OutsideJob)

- UI Layer (Fragments, Compose)
- ViewModel Layer
- Repository Layer
- Local DB (Room/SQLite)
- WorkManager (Sync)

(HTTPS REST API)



Backend Server

- Authentication Service
- Task Management Service
- Report Generation Service
- Sync Service
- PostgreSQL Database
- File Storage (S3/GCS)

### 7.2 Przepływ danych

1. Pracownik uruchamia aplikację
  2. Loguje się (email + password)
  3. Aplikacja pobiera listę Task'ów z serwera
  4. Task'i przechowywane w Room (SQLite)
  5. Pracownik pracuje offline (zmiany lokalnie)
  6. Po powrocie internetu – WorkManager synchronizuje zmiany
  7. Raport uploadowany do serwera
  8. Serwer generuje raporty w różnych formatach
- 

## 8 Model danych (ERD)

### 8.1 Tabela Users

```

1  CREATE TABLE users (
2      id BIGSERIAL PRIMARY KEY,
3      email VARCHAR(255) UNIQUE NOT NULL,
4      password_hash VARCHAR(255) NOT NULL,
5      first_name VARCHAR(100),
6      last_name VARCHAR(100),
7      role VARCHAR(50) DEFAULT 'WORKER', -- WORKER, MANAGER,
8          ADMIN
9      company_id BIGINT NOT NULL,
10     created_at TIMESTAMP DEFAULT NOW(),
11     updated_at TIMESTAMP DEFAULT NOW(),
12     last_login TIMESTAMP,
13     FOREIGN KEY (company_id) REFERENCES companies(id)
14 );

```

## 8.2 Tabela Tasks

```

1  CREATE TABLE tasks (
2      id BIGSERIAL PRIMARY KEY,
3      display_id VARCHAR(20) UNIQUE NOT NULL,
4      title VARCHAR(255) NOT NULL,
5      description TEXT,
6      assigned_to BIGINT NOT NULL,
7      created_by BIGINT NOT NULL,
8      status VARCHAR(50) DEFAULT 'TODO',
9          -- TODO, IN_TRANSIT, IN_PROGRESS, SUSPENDED, COMPLETED
10     due_date TIMESTAMP,
11     location_lat DECIMAL(10, 8),
12     location_lng DECIMAL(11, 8),
13     created_at TIMESTAMP DEFAULT NOW(),
14     updated_at TIMESTAMP DEFAULT NOW(),
15     completed_at TIMESTAMP,
16     FOREIGN KEY (assigned_to) REFERENCES users(id),
17     FOREIGN KEY (created_by) REFERENCES users(id)
18 );

```

## 8.3 Tabela Subtasks

```

1  CREATE TABLE subtasks (
2      id BIGSERIAL PRIMARY KEY,
3      task_id BIGINT NOT NULL,
4      title VARCHAR(255) NOT NULL,
5      description TEXT,
6      type VARCHAR(50), -- CHECKBOX, TEXT, PHOTO, NOTES
7      completed BOOLEAN DEFAULT FALSE,
8      completed_at TIMESTAMP,
9      FOREIGN KEY (task_id) REFERENCES tasks(id)
10 );

```

## 8.4 Tabela Reports

```

1 CREATE TABLE reports (
2     id BIGSERIAL PRIMARY KEY,
3     task_id BIGINT UNIQUE NOT NULL,
4     worker_id BIGINT NOT NULL,
5     description TEXT,
6     duration_minutes INT,
7     signature_base64 TEXT,
8     status VARCHAR(50) DEFAULT 'DRAFT',
9     -- DRAFT, SUBMITTED, APPROVED, REJECTED
10    created_at TIMESTAMP DEFAULT NOW(),
11    submitted_at TIMESTAMP,
12    FOREIGN KEY (task_id) REFERENCES tasks(id),
13    FOREIGN KEY (worker_id) REFERENCES users(id)
14 );

```

## 8.5 Tabela Attachments

```

1 CREATE TABLE attachments (
2     id BIGSERIAL PRIMARY KEY,
3     task_id BIGINT NOT NULL,
4     file_path VARCHAR(255),
5     file_type VARCHAR(50), -- PHOTO, PDF, DOCUMENT
6     uploaded_at TIMESTAMP DEFAULT NOW(),
7     FOREIGN KEY (task_id) REFERENCES tasks(id)
8 );

```

## 8.6 Tabela Suspensions

```

1 CREATE TABLE suspensions (
2     id BIGSERIAL PRIMARY KEY,
3     task_id BIGINT NOT NULL,
4     reason TEXT NOT NULL,
5     suspended_at TIMESTAMP DEFAULT NOW(),
6     resumed_at TIMESTAMP,
7     FOREIGN KEY (task_id) REFERENCES tasks(id)
8 );

```

# 9 Interfejs użytkownika (UI/UX)

## 9.1 Główne ekrany

1. **Ekran logowania** – email, password, checkbox “Zapamiętaj mnie”, przycisk “Zaloguj”

2. **Lista zadań** – liste Tasks z filtrami (ALL, TODO, IN\_PROGRESS, COMPLETED, SUSPENDED, OVERDUE), pull-to-refresh
3. **Szczegóły zadania** – informacje, przyciski akcji (Rozpocznij dojazd, Rozpocznij pracę)
4. **Ekran pracy** – checklist podzadań, pola tekstowe, przycisk dodania zdjęcia, Zakończ, Wstrzymaj
5. **Podekran pracy** – szczegółowy ekran z polami w zależności od potrzeb klienta projektowany indywidualnie
6. **Navigation Drawer** – Tasks, + Add Task, Profile, Wyloguj

## 9.2 Wytyczne designu

- Material Design 3
  - Kolory: Primary (Teal 2196F3), Secondary (Gray), Error (Red)
  - Typography: Roboto (system font)
  - Spacing: 8px, 16px, 24px
  - Icons: Material Icons
  - Accessibility: WCAG 2.1 AA standard
- 

# 10 Integracje

## 10.1 Firebase (opcjonalnie)

- Push notifications
  - Crash reporting
  - Analytics
- 

# 11 Plan testów

## 11.1 Testy funkcjonalne

- Logowanie (poprawne i niepoprawne dane)
- Zapamiętanie hasła
- Przeglądanie listy zadań

- Filtrowanie
- Rozpoczęcie pracy
- Wstrzymanie z powodem
- Rozliczenie pracy
- Synchronizacja danych

## 11.2 Testy offline

- Praca bez internetu
- Synchronizacja po powrocie internetu
- Brak konfliktów

## 11.3 Testy wydajności

- Ładowanie listy 100 zadań
- Responsywność UI
- Zużycie baterii i pamięci
- Upload dużych zdjęć

## 11.4 Testy bezpieczeństwa

- Szyfrowanie danych w spoczynku i w ruchu
  - Wygasanie sesji
  - Logowanie do Keystore
- 

# 12 Wytyczne kodowania i konwencje

## 12.1 Język programowania

- Kotlin (główny) dla Android App
- Java / Kotlin dla Backend'u
- TypeScript (opcjonalnie) dla Admin Panel

## 12.2 Struktura projektu

```
app/
  data/
    database/
    model/
    remote/ (API calls)
    repository/
domain/
  usecase/
  entity/
presentation/
  ui/
    login/
    tasks/
    task_detail/
    work/
  viewmodel/
worker/ (sync, background jobs)
util/
```

## 12.3 Konwencje nazewnictwa

- Klasy: PascalCase (“TaskDetailFragment”)
- Funkcje: camelCase (“fetchTasks”)
- Stałe: UPPER\_SNAKE\_CASE (“MAX\_RETRIES”)
- XML ids: snake\_case (“btn\_start\_work”)

## 12.4 Commits

- Format: “[TYPE] Description”
- Typy: feat, fix, refactor, test, docs, chore
- Przykład: “[feat] Add task filtering by status”

—

# 13 Harmonogram wdrożenia

## 13.1 Fazy projektu

- **Faza 1 (Sprint 1-2):** Logowanie, autentykacja, zapamiętanie hasła
- **Faza 2 (Sprint 3-4):** Lista zadań, szczegóły, filtrowanie
- **Faza 3 (Sprint 5-6):** Praca nad zadaniem, checklist, wstrzymywanie

- **Faza 4 (Sprint 7-8):** Rozliczenie, raportowanie, upload zdjęć
- **Faza 5 (Sprint 9-10):** Synchronizacja offline-first, WorkManager
- **Faza 6 (Sprint 11-12):** Testy (QA), optymalizacja, deployment

## 13.2 Kamienie milowe

- Koniec Sprint 4: Alpha (prototyp funkcjonalny)
  - Koniec Sprint 8: Beta (pełny feature set)
  - Koniec Sprint 12: Release 1.0.0 (produkcja)
- 

# 14 Podsumowanie

Niniejsza specyfikacja stanowi pełną dokumentację wymagań funkcjonalnych i niefunkcjonalnych dla aplikacji OutsideJob. Dokument obejmuje:

- 14 głównych wymagań funkcjonalnych (REQ-F)
- 7 wymagań niefunkcjonalnych (NFR)
- 5 kluczowych przypadków użycia
- 6 scenariuszy testowych
- Model danych (ERD) z tabelami SQL
- Architekturę systemu
- Wytyczne UX/UI
- Plan testów i wdrażania

Dokument stanowi fundament dla całego projektu i powinien być zatwierdzony przez stakeholderów przed przejściem do fazy implementacji.

## 14.1 Dalsze kroki

1. Zatwierdzenie specyfikacji przez zespół i klienta
2. Przygotowanie mockup'ów UI/UX (Figma)
3. Specyfikacja API REST (OpenAPI/Swagger)
4. Przygotowanie planu architektonicznego (szczegółowo)
5. Konfiguracja środowiska (dev, staging, prod)
6. Rozpoczęcie Sprint 1