

# Inżynieria Oprogramowania

III - Faza konstrukcji - Grudzień

Dominik Walecko, Wojciech Woźniak

1. Dla 3 funkcji ( przypadków użycia, historyjek użytkownika) opracowanie dokładnych przypadków testowych, tzw przypadków testowych niższego rzędu (z podaniem konkretnych wartości danych i wyników oraz opisami stanu początkowego i końcowego).
- 

**1. Historyjka: Jako osoba zainteresowana kupnem psa, chciałbym móc przeglądać jak największą bazę ogłoszeń, abym miał pewność, że wybiorę najlepszego kandydata.**

#### **Przypadek testowy 1**

- **ID:** TC001
- **Opis:** Użytkownik przegląda listę ogłoszeń w aplikacji.
- **Stan początkowy:** Użytkownik zalogowany w aplikacji i znajduje się na ekranie głównym z możliwością przeglądania ogłoszeń.
- **Dane wejściowe:**
  - Wybór kategorii ogłoszeń "Psy do adopcji".
  - Wprowadzenie filtrów: "Wielkość psa" = "Średni", "Wiek psa" = "1-3 lata".
- **Kroki:**
  - Użytkownik wybiera kategorię ogłoszeń "Psy do adopcji".
  - Użytkownik ustawia filtry na "Wielkość psa" = "Średni" i "Wiek psa" = "1-3 lata".
  - Użytkownik przegląda wyniki.
- **Oczekiwany wynik:**
  - Lista ogłoszeń zawiera tylko psy średniej wielkości w wieku od 1 do 3 lat.
  - Ogłoszenia są posortowane według daty dodania (od najnowszego).
- **Stan końcowy:**
  - Użytkownik ma dostęp do listy pasujących ogłoszeń, które spełniają określone kryteria.

---

**2. Historyjka: Jako pracownik schroniska, chciałbym móc wystawiać swoje ogłoszenia tak, abym miał pewność, że zobaczy je jak największa liczba zainteresowanych osób, aby zwiększyć szanse na znalezienie nabywcy.**

**Przypadek testowy 2**

- **ID:** TC002
- **Opis:** Pracownik schroniska wystawia ogłoszenie o psie.
- **Stan początkowy:** Użytkownik (pracownik schroniska) zalogowany w aplikacji i znajduje się na ekranie "Dodaj ogłoszenie".
- **Dane wejściowe:**
  - Tytuł ogłoszenia: "Pies do adopcji – Max, 2 lata, średni".
  - Opis ogłoszenia: "Max to pies o średnim wzroście, uwielbia spacerować, świetny z dziećmi".
  - Kategoria: "Psy do adopcji".
  - Wybór filtrów: "Wiek" = "2 lata", "Wielkość" = "Średni".
- **Kroki:**
  - Użytkownik wybiera opcję "Dodaj ogłoszenie".
  - Wypełnia formularz ogłoszenia (tytuł, opis, kategoria, filtry).
  - Użytkownik zatwierdza ogłoszenie.
- **Oczekiwany wynik:**
  - Ogłoszenie zostaje dodane do bazy ogłoszeń.
  - Ogłoszenie pojawia się na stronie głównej aplikacji i jest widoczne dla wszystkich użytkowników, którzy przeglądają psy do adopcji w kategorii "Średni" i "2 lata".
- **Stan końcowy:**
  - Ogłoszenie jest aktywne i widoczne dla użytkowników.

---

### 3. Historyjka: Jako osoba chcąca pełnić funkcję wolontariusza, chciałbym móc zgłaszać swoją chęć udziału w wolontariacie w sposób jak najbardziej przystępny, aby zaoszczędzić swój czas.

#### Przypadek testowy 3

- **ID:** TC003
- **Opis:** Użytkownik zgłasza chęć udziału w wolontariacie.
- **Stan początkowy:** Użytkownik zalogowany w aplikacji i znajduje się na ekranie "Wolontariat".
- **Dane wejściowe:**
  - Wybór wolontariatu: "Pomoc w schronisku - weekendy".
  - Wybór terminu: "Sobota 10:00".
  - Dodanie komentarza: "Chciałbym pomóc w opiece nad psami".
- **Kroki:**
  - Użytkownik wybiera kategorię wolontariatu "Pomoc w schronisku - weekendy".
  - Użytkownik wybiera preferowany termin (np. "Sobota 10:00").
  - Użytkownik dodaje opcjonalny komentarz.
  - Użytkownik klikając przycisk "Zgłoś się", zatwierdza zgłoszenie.
- **Oczekiwany wynik:**
  - Zgłoszenie wolontariusza jest zapisane w systemie.
  - Użytkownik otrzymuje potwierdzenie na ekranie oraz email z informacją o pomyślnym zgłoszeniu.
- **Stan końcowy:**
  - Użytkownik jest zapisany na wolontariat na wybrany dzień i godzinę, a aplikacja zaktualizowana o zgłoszenie.

2. Na podstawie norm ISO/IEC 9126 i 25 000 (przesłane materiały) opracowanie pomiarów mających na celu sprawdzenie czy system spełnia opisane w poprzednich dokumentach wymagania niefunkcjonalne.
- 

### 1. Interfejs przyjazny w obsłudze

- **Cel:** Sprawdzić, czy interfejs aplikacji jest intuicyjny i czy najważniejsze funkcje są w stałych, widocznych miejscach.
- **Pomiar:**
  - **M1:** Czas, w którym użytkownik znajduje i korzysta z najważniejszych funkcji (np. czas przejścia do opcji dodania ogłoszenia).
  - **M2:** Liczba przypadków, w których użytkownik nie wie, jak skorzystać z danej funkcji (poziom trudności).
  - **Metoda:** Przeprowadzenie testów użyteczności z użytkownikami. Zbieranie danych na temat czasu realizacji zadań i liczby pomyłek.
  - **Norma:** Czas użycia funkcji powinien wynosić mniej niż 5 minut, a liczba błędnych działań powinna być poniżej 5%.

### 2. Niezależność od systemu

- **Cel:** Sprawdzić, czy aplikacja działa poprawnie na najnowszych przeglądarkach i jest niezależna od wielkości ekranu.
- **Pomiar:**
  - **M1:** Czas ładowania aplikacji w różnych przeglądarkach (np. Chrome, Firefox, Safari, Edge).
  - **M2:** Sprawdzenie responsywności aplikacji na różnych rozdzielczościach ekranu (od smartfonów po duże ekrany).
  - **Metoda:** Testy wydajnościowe w różnych przeglądarkach i na urządzeniach o różnych rozdzielczościach.
  - **Norma:** Czas ładowania aplikacji powinien wynosić maksymalnie 3 sekundy, a aplikacja powinna być w pełni funkcjonalna na ekranach od 320px do 1920px.

### 3. Aktualizacja ogłoszeń

- **Cel:** Sprawdzić, czy ogłoszenia są aktualizowane na bieżąco i czy dane są spójne.
- **Pomiar:**
  - **M1:** Czas, w jakim ogłoszenia są aktualizowane po ich edytowaniu lub dodaniu.

- **M2:** Liczba przypadków, w których dane ogłoszenia zostały zaktualizowane w sposób niepoprawny.
- **Metoda:** Monitorowanie czasu aktualizacji ogłoszeń oraz testy spójności danych w bazie.
- **Norma:** Czas aktualizacji ogłoszenia nie powinien przekraczać 5 sekund, a spójność danych powinna wynosić 100%.

#### 4. Informacje o błędach operacyjnych

- **Cel:** Sprawdzić, czy użytkownik otrzymuje jasną informację o błędach w operacjach takich jak dodawanie ogłoszenia czy tworzenie konta.
- **Pomiar:**
  - **M1:** Czas potrzebny użytkownikowi na zrozumienie komunikatu o błędzie.
  - **M2:** Liczba przypadków, w których użytkownik nie rozumie komunikatu o błędzie.
  - **Metoda:** Testy użyteczności, w których użytkownicy są proszeni o wykonanie operacji i zgłoszenie błędów.
  - **Norma:** 95% użytkowników powinno poprawnie zrozumieć komunikat w ciągu 10 sekund.

#### 5. Wsparcie dla języków

- **Cel:** Sprawdzić, czy aplikacja wspiera język polski i angielski w jej wczesnych wersjach.
- **Pomiar:**
  - **M1:** Liczba przetłumaczonych słów w aplikacji w języku angielskim.
  - **M2:** Liczba błędów tłumaczeń w aplikacji.
  - **Metoda:** Przeprowadzenie testów językowych z użytkownikami mówiącymi w języku angielskim.
  - **Norma:** Przetłumaczone co najmniej 95% tekstów w aplikacji, a liczba błędnych tłumaczeń nie powinna przekroczyć 2%.

### 3. Opracowanie planu beta testowania

---

#### Faza 1: Wstępne testy funkcjonalności (3 pierwsze tygodnie)

##### **Cel:**

Testowanie podstawowych funkcji aplikacji na wczesnym etapie, aby upewnić się, że kluczowe mechanizmy, takie jak rejestracja użytkowników, dodawanie ogłoszeń oraz wyszukiwanie psów działają poprawnie.

##### **Zadania:**

- **Testowanie podstawowych funkcji:**
  - Rejestracja użytkowników (schronisk i osób poszukujących psów).
  - Dodawanie ogłoszeń o psach do adopcji.
  - Sprawdzanie poprawności formularzy oraz opcji filtrowania psów (rasa, wiek, lokalizacja).
- **Testowanie systemu logowania i rejestracji:**
  - Testowanie procesu rejestracji i logowania dla schronisk oraz użytkowników.
- **Testowanie procesu dodawania psów do bazy danych:**
  - Weryfikacja poprawności zapisywanych danych o psach (rasa, wiek, wielkość).
  - Walidacja wprowadzonych danych.

---

## **Faza 2: Testowanie z użytkownikami (po wdrożeniu części bazy psów - kilka schronisk z okolicy w ramach współpracy)**

### **Cel:**

Testowanie aplikacji przez zaproszoną grupę użytkowników.

### **Zadania:**

- **Udostępnienie aplikacji wybranej grupie testerów:**
  - Użytkownicy będą mogli rejestrować się jako wolontariusze, przeglądać ogłoszenia i dodawać swoje dane do bazy.
  - Sprawdzanie, czy wszystkie funkcjonalności są dostępne i działają poprawnie.
- **Testowanie komunikacji między użytkownikami:**
  - Testowanie systemu czatu pomiędzy osobami wystawiającymi ogłoszenia a zainteresowanymi adopcją.
- **Zbieranie feedbacku od użytkowników:**
  - Testowanie, jak łatwo użytkownicy znajdują psy, które ich interesują.
  - Uzyskanie opinii na temat procesu rejestracji, dodawania ogłoszeń i interfejsu aplikacji.
- **Testowanie funkcji wolontariatu:**
  - Rejestracja wolontariuszy i sprawdzanie kalendarza dostępności.



---

### Faza 3: Testy przedpremierowe (2 tygodnie przed premierą)

#### **Cel:**

Testowanie aplikacji w warunkach zbliżonych do rzeczywistego użytkowania, sprawdzenie wszystkich kluczowych funkcji i upewnienie się, że aplikacja działa bezawaryjnie przed wydaniem jej na szeroką skalę.

#### **Zadania:**

- **Testowanie funkcji reklam:**
  - a. Sprawdzenie, czy system reklam działa zgodnie z oczekiwaniami i nie zakłóca działania aplikacji.
- **Testowanie funkcji donacji i czatu (jeśli zostaną zaimplementowane):**
  - b. Testowanie systemu donacji dla schronisk oraz czatu pomiędzy użytkownikami.
- **Testowanie skalowalności aplikacji:**
  - c. Testowanie działania aplikacji przy większym obciążeniu (więcej użytkowników, więcej ogłoszeń).
- **Powtórne testowanie wszystkich funkcji:**
  - d. Testowanie procesów rejestracji, dodawania ogłoszeń, filtrowania psów oraz interakcji z użytkownikami.

#### 4. Napisanie planu zarządzania jakością oprogramowania.

---

- a) Testowanie wszystkich funkcji tworzonych przez programistów na bieżąco. Każda nowa funkcjonalność będzie natychmiast sprawdzana pod kątem poprawności działania.
- b) Testowanie aplikacji i bazy danych w celu zapewnienia ich bezpieczeństwa. Przeprowadzimy audyty zabezpieczeń, aby upewnić się, że aplikacja jest odporna na potencjalne zagrożenia.
- c) Pisanie aplikacji zgodnie z ustalonymi standardami kodowania. Cały proces kodowania będzie opierał się na najlepszych praktykach, co zapewni łatwiejszą konserwację i rozwój aplikacji w przyszłości.
- d) Regularne raportowanie postępów przez zespół. Programiści będą codziennie przekazywać raporty z realizowanych zadań, co umożliwi bieżące monitorowanie postępu prac i dotrzymywanie ustalonych terminów.
- e) Użycie systemu Git do kontroli wersji. Cały kod źródłowy będzie przechowywany w systemie kontroli wersji, co ułatwi śledzenie zmian i poprawi współpracę w zespole.
- f) Dołączenie dokumentacji na etapie końcowym projektu. Po zakończeniu prac nad aplikacją, do projektu zostanie dołączona szczegółowa dokumentacja opisująca jego działanie i architekturę.

**5. Sporządzenie dokładniejszego planu wykonania produktu, dokładniejszej oceny pracochłonności, dokładniejszego harmonogramu.**

---

Nr	Nazwa zadania	Czas	Początek	Koniec
1.	Przygotowanie tablicy koncepcyjnej	3 dni	29.10.2024	31.10.2024
2.	Opracowanie architektury systemu	4 dni	01.11.2024	04.11.2024
3.	Rekrutacja pracowników, podział pracy	14 dni	05.11.2024	19.11.2024
4.	Implementacja podstawowych funkcjonalności - osiągnięcie poziomu pozwalającego bardzo bazowe korzystanie z aplikacji	33 dni	20.11.2024	23.12.2024
5.	Przerwa świąteczna	3 dni	24.12.2024	26.12.2024
6.	Stworzenie bazy testowej, rekrutacja beta-testerów	6 dni	27.12.2024	02.01.2025
7.	Implementacja bardziej zaawansowanych funkcji, poprawki błędów znalezionych przez beta-testerów	21 dni	03.01.2025	24.01.2025
8.	Testowanie bezpieczeństwa i jakości	7 dni	25.01.2025	31.01.2025
9.	Testy przedpremierowe, ostatnie poprawki	7 dni	01.02.2025	08.02.2025
10.	Opublikowanie aplikacji w sklepach Google/Apple	1 dzień	09.02.2025	10.02.2025

## **6. Ocena zgodności wykonanych prac z wizją systemu i specyfikacją wymagań.**

---

Dotychczasowe wyniki prac nad Psinderem są zadowalające, a postępy w projekcie są widoczne. Istnieje niestety możliwość, że czas realizacji projektu zostanie przedłużony o około dwa tygodnie. Jest to związane z dłuższym niż przewidywany czasem rekrutacji pracowników, oraz chęcią zapewnienia lepszej jakości przy premierze (co zapewni lepszą retencję użytkowników).