

Raport – Terminarz Kliniki Weterynaryjnej

Część 1: Opis aplikacji

1.1 Cel

Aplikacja służy do prowadzenia terminarza wizyt w małej klinice weterynaryjnej. Umożliwia dodawanie wizyt, przeglądanie ich w formie listy, edycję, archiwizację oraz ustawienie przypomnienia przed wizytą.

1.2 Grupa docelowa

- pracownicy rejestracji, personel kliniki weterynaryjnej,
- małe gabinety, które chcą prostego terminarza offline na telefonie.

1.3 Główne funkcjonalności

- lista wizyt (aktualne, archiwum),
- wyszukiwanie,
- formularz dodawania nowej wizyty,
- podgląd szczegółów wizyty,
- edycja i usuwanie wizyty,
- archiwizacja, przywracanie wizyty,
- przypomnienie (powiadomienie) ustawiane 15, 30 lub 60 minut przed wizytą,
- eksport wszystkich wizyt do pliku CSV.

Część 2: Wybrana technologia

2.1 Technologia

Android natywny (Kotlin + Jetpack Compose). Biblioteki i komponenty:

- Jetpack Compose (Material3) – UI,
- Navigation Compose – nawigacja między ekranami,
- Room – lokalna baza danych SQLite,
- WorkManager – wykonywanie przypomnień w tle i wyświetlanie notyfikacji,
- Kotlin Coroutines + Flow – obserwacja danych (aktualizacja listy i szczegółów wizyt).

2.2 Uzasadnienie wyboru

Wybrano natywne tworzenie aplikacji na Androida w Kotlinie, ponieważ to obecnie standard w ekosystemie Android i daje najlepszą zgodność z systemem oraz stabilność działania. Do interfejsu użyto Jetpack Compose, ponieważ pozwala szybciej budować ekrany i łatwiej zarządzać stanem widoków niż w podejściu opartym o XML, a przy tym dobrze wspiera nowoczesne wzorce projektowania UI. Dane wizyt są przechowywane lokalnie, więc aplikacja działa offline i nie wymaga serwera ani kont użytkowników, co pasuje do prostego terminarza do użytku w małej klinice. Do przypomnień zastosowano mechanizm działający w tle, bo w praktyce użytkownik musi dostać powiadomienie nawet wtedy, gdy aplikacja nie jest otwarta. Całość jest prosta, lekka i możliwa do uruchomienia na typowym telefonie bez dodatkowej infrastruktury.

Część 3: Architektura aplikacji

3.1 Schemat struktury

MainActivity.kt

- start aplikacji, ustawienie Compose, utworzenie kanału powiadomień

ui/screens/

- AppNav.kt – graf nawigacji, pobranie danych z bazy, routing ekranów
- VisitListScreen.kt – lista wizyt, filtrowanie, wyszukiwanie, eksport CSV
- VisitFormScreen.kt – formularz dodawania/edykcji + walidacja + wybór daty i godziny
- VisitDetailsScreen.kt – szczegóły, usuwanie, archiwizacja, przejście do edycji
- Components.kt – małe komponenty UI (np. “pigułki” statusu i przypomnienia)
- Utils.kt – formatowanie daty, budowanie CSV

data/

- Visit.kt – encja Room
- VisitDao.kt – DAO (CRUD + obserwacja danych przez Flow)
- AppDatabase.kt – konfiguracja Room + DbProvider

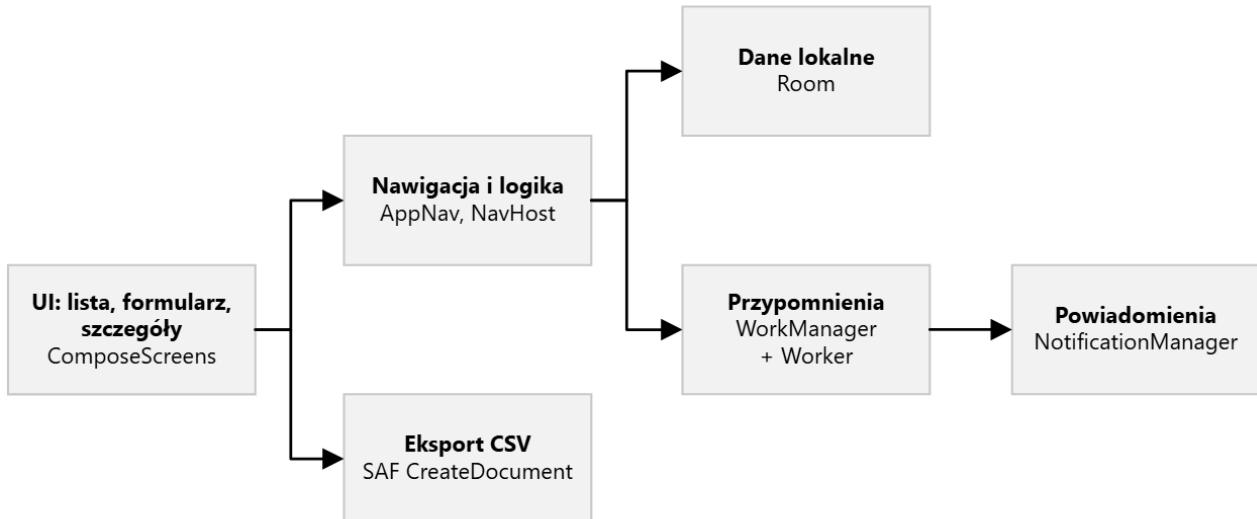
notifications/

- ReminderWorker.kt – worker do przypomnień oraz planowanie lub anulowanie powiadomień

ui/theme/

- motyw kolorystyczny i typografia

3.2 Uproszczony diagram architektury



3.3 Użyte wzorce i podejścia

- Single Activity + Compose Navigation (jedna aktywność i wiele ekranów w NavHost),
- DAO pattern (Room DAO do operacji na bazie),
- Singleton (DbProvider jako pojedyncza instancja bazy),
- Observer / reactive UI (Flow z Room + collectAsState w Compose),
- Worker pattern (WorkManager + CoroutineWorker do przypomnień).

Część 4: Opis zaimplementowanych funkcjonalności

1. Lista wizyt (LazyColumn) – wyświetla wizyty z bazy jako karty. Kliknięcie otwiera szczegóły.
2. Filtrowanie Aktualne/Archiwum – dwa przyciski typu FilterChip przełączają widok między wizytami aktywnymi i zarchiwizowanymi.
3. Wyszukiwanie na liście – pole tekstowe filzuje wizyty po imieniu zwierzaka, właściwiemu, gatunku, lekarzu, opisie i telefonie.
4. Dodawanie wizyty (formularz) – z polami: data i godzina, zwierzak, gatunek, właściciel, telefon, lekarz, opis, przypomnienie.
5. Walidacja danych w formularzu – blokuje zapis, jeśli nie ma daty, data jest w przeszłości albo brakuje wymaganych pól.
6. Edycja wizyty – formularz wypełnia się danymi z bazy i zapisuje zmiany.
7. Szczegóły wizyty – pokazuje pełne dane wizyty oraz informacje o przypomnieniu.

8. Usuwanie wizyty – w szczegółach jest dialog potwierdzający. Po usunięciu kasuje się też przypomnienie.
9. Archiwizacja i przywracanie wizyty – przełącznik w szczegółach zmienia isArchived. Dla archiwum przypomnienia są anulowane.
10. Powiadomienia (przypomnienie) przed wizytą – WorkManager uruchamia ReminderWorker o czasie *wizyta - X minut* i pokazuje notyfikację z danymi wizyty.
11. Obsługa zgody POST_NOTIFICATIONS – przy starcie aplikacja prosi o zgodę, jeśli system jej wymaga.
12. Eksport do CSV – z poziomu listy można wygenerować CSV ze wszystkimi wizytami i zapisać plik w wybranej lokalizacji.

Część 5: Zrzuty ekranu

Terminarz wizyt

Aktualne Archiwum

szukaj Kropka

- 2026-01-31 21:39 • Frytka

(Aktualna) (30 min)

Właściciel: Adam Grzybowski
- 2026-01-31 21:44 • Kropka

(Aktualna) (10 min)

Właściciel: Janusz Nowak
Lekarz: Jakub Bocian
- 2026-02-10 09:20 • Fafik

(Aktualna) (10 min)

Właściciel: Anna Kowalska
- 2026-02-12 12:00 • Luna

(Aktualna) (1 godz.)

Właściciel: Agnieszka Wojtkowska

+

Terminarz wizyt

Aktualne Archiwum

szukaj Kropka

- 2026-02-13 10:10 • Kropka

(Aktualna) (Brak)

Właściciel: Janusz Nowak

+

Dodaj wizytę

Data: 2026-02-11 15:00

Imię zwierzątka* Frytka

Gatunek* Fretka

Imię i nazwisko właściciela* Adam Grzybowski

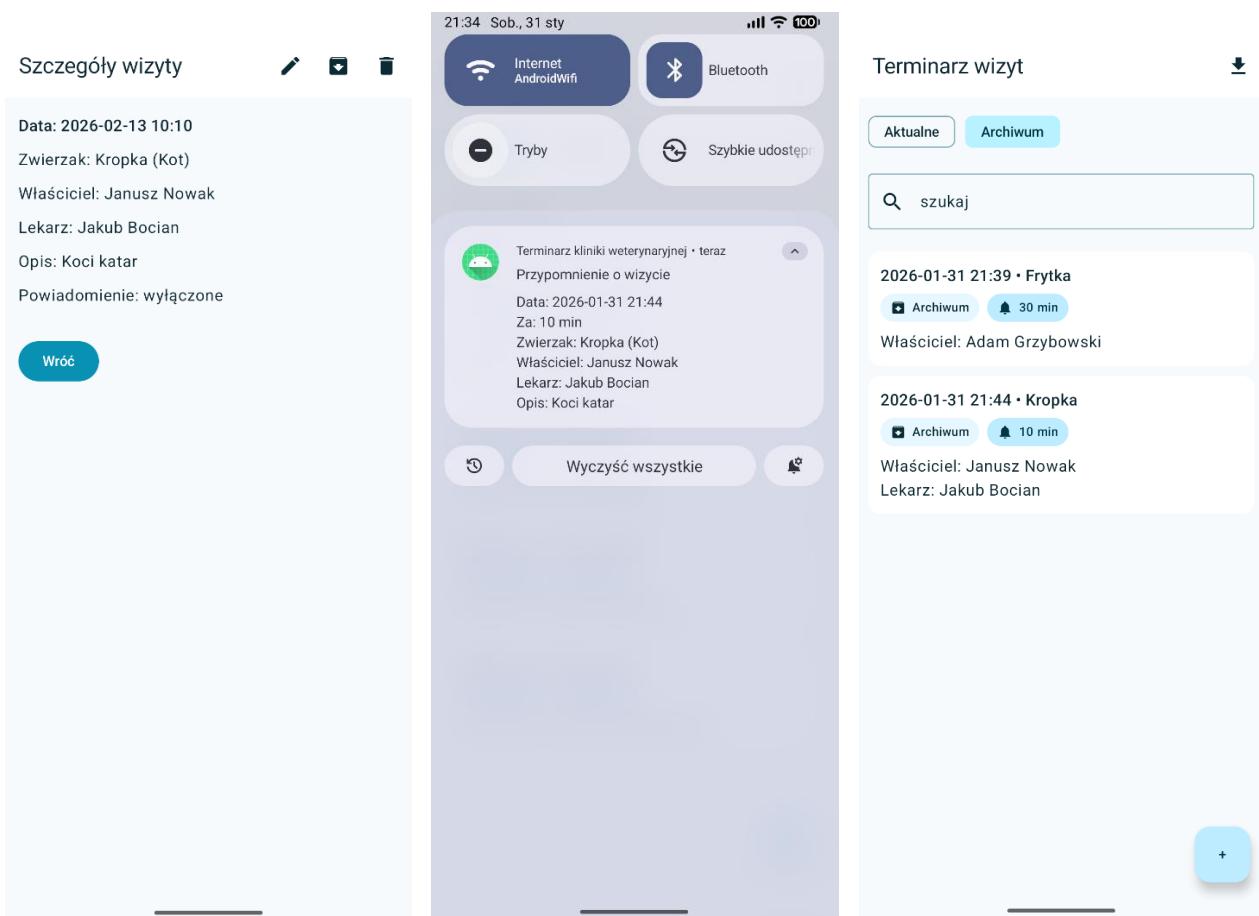
Telefon (opcjonalnie)

Lekarz (opcjonalnie)

Opis wizyty (opcjonalnie)

Powiadomienie

Anuluj Zapisz



Część 6: Instrukcja uruchomienia

6.1 Wymagania

- Android Studio,
- JDK 11,
- emulator lub telefon.

6.2 Uruchomienie w Android Studio

- Rozpakuj projekt .zip.
- Otwórz projekt w Android Studio.
- Poczekaj na Gradle Sync.
- Wybierz urządzenie albo podłącz telefon z włączonym USB debugging.
- Kliknij Run i wybierz konfigurację app.

Część 7: Napotkane problemy

7.1 Opis problemu

W trakcie tworzenia motywu Material3 pojawiał się fioletowy odcień mimo ustawiania własnych kolorów (np. w elementach typu tło w niektórych komponentach Material3).

7.2 Przyczyna

Domyślne ustawienia motywu Material3 oraz mechanizm dynamicznych kolorów potrafią nadpisywać część kolorów albo używać wartości domyślnych, jeśli nie są ustawione wszystkie potrzebne pola schematu kolorów.

7.3 Rozwiązanie

- Zdefiniowano pełny zestaw kolorów w schemacie Material3 (primary, secondary, tertiary, container, surface, background, outline itd.).
- Wyłączono dynamiczne kolory (dynamicColor = false), żeby system nie mieszał kolorystyki aplikacji.

Część 8: Możliwości rozwoju

1. Widok kalendarza (dzień, tydzień, miesiąc) zamiast samej listy.
2. Wykrywanie konfliktów terminów (np. ostrzeżenie, że lekarz ma już wizytę o tej godzinie).
3. Status wizyty: zaplanowana, odwołana, zrealizowana i filtrowanie po statusie.
4. Profil pacjenta (historia wizyt jednego zwierzaka, notatki, leki, zalecenia).
5. Import CSV (ułatwi przenoszenie danych).
6. Obsługa wielu użytkowników i synchronizacja danych między urządzeniami.