Aplikacja mobilna jako natywna aplikacja dla systemu Android

Z wykorzystaniem Android Studio
ARIUS Zadanie 5.

Julia Chilczuk 325261

Politechnika Warszawska
Wydział Elektroniki i Technik Informacyjnych

10 maja 2025

Spis treści

1. Wstęp

W ramach realizacji zadania zaimplementowano aplikację w języku Java dla systemu Android, która wyświetla istę zadań do wykonania. W celu osiągnięcia założonych funkcji wykorzystano ListView oraz intencje.

2. Struktura aplikacji

Aplikacja została zaimplementowana w języku Java z wykorzystaniem Android SDK. Składa się z dwóch głównych komponentów:

— Aktywności (Java) – odpowiedzialne za logikę aplikacji i obsługę interakcji użytkownika.
— Układów widoków (XML) – definiujących graficzny interfejs użytkownika.

2.1. Aktywności Java

2.1.1. MainActivity.java

Główna aktywność wyświetla listę zadań w komponencie ListView. Lista zasilana jest przez niestandardowy adapter TaskAdapter, który wyświetla tytuł zadania, datę oraz ikonę stanu (wykonane/niewykonane/przeterminowane). Dane zadań są pobierane z plików zasobów strings.xml jako tablice string-array.

Po kliknięciu na element listy użytkownik przenoszony jest do drugiej aktywności (TaskDetailsActivity) przy użyciu mechanizmu Intent.

2.1.2. TaskDetailsActivity.java

Druga aktywność wyświetla szczegóły wybranego zadania: tytuł, status, opis, termin wykonania oraz dwa przyciski umożliwiające zmianę statusu zadania. Jeśli zadanie jest przeterminowane (na podstawie porównania bieżącego czasu z terminem wykonania), przyciski te zostają zablokowane, a status zostaje automatycznie ustawiony na OVERDUE. Po edycji dane są zwracane do MainActivity za pomocą setResult().

2.2. Klasy pomocnicze

2.2.1. Task.java

Model danych Task implementuje interfejs Serializable, co umożliwia przekazywanie obiektów pomiędzy aktywnościami. Klasa zawiera pola: tytuł, opis, termin oraz status, a także metodę isOverdue(), która określa, czy zadanie jest przeterminowane.

2.2.2. TaskAdapter.java

Adapter rozszerza klasę ArrayAdapter¡Task¿ i odpowiada za wyświetlanie każdego elementu zadania w liście. Na podstawie statusu zadania dobierana jest odpowiednia ikona (ic_done, ic_not_done, ic_overdue) z katalogu drawable.

2.3. Układy widoków XML

2.3.1. activity_main.xml

Szablon głównego widoku aplikacji. Zawiera tytuł strony oraz komponent ListView, który zajmuje większość powierzchni ekranu. Lista jest wiązana z adapterem w MainActivity.

2.3.2. single_row.xml

Szablon jednego elementu listy w głównym widoku aplikacji, zawiera ikonę stanu zadania, tytuł zadania oraz termin wykonania.

2.3.3. activity_task_details.xml

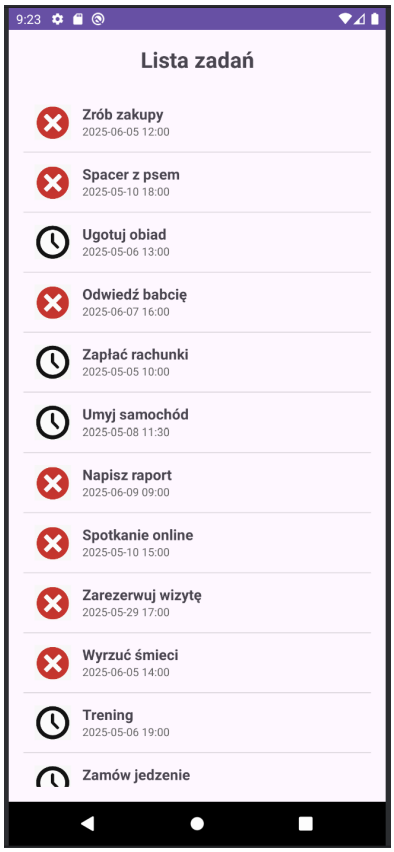
Szablon widoku szczegółów zadania, zawiera:

— TextView's do wyświetlenia tytułu, statusu, opisu i terminu wykonania zadania,
— Dwa przyciski do ustawienia statusu na „Wykonane" lub „Niewykonane".

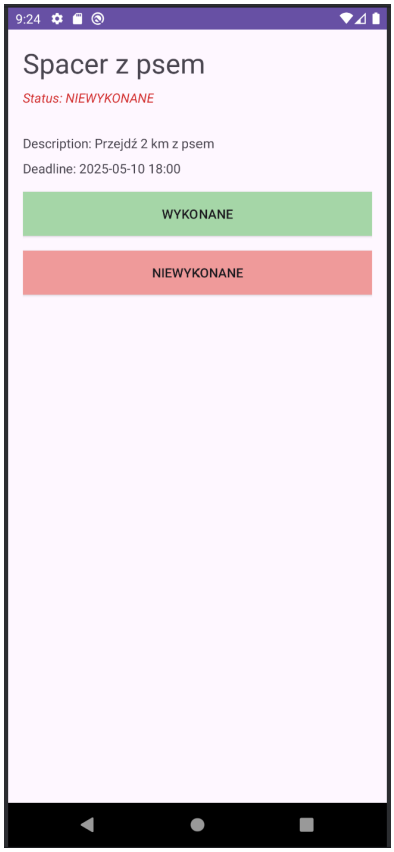
3. Efekty działania aplikacji

3.1. Wyświetlanie szczegółów zadania

Poniżej lista zadań do wykonania (po lewej stronie) oraz aktywność prezentująca szczegóły wybranego elementu listy – „Spacer z psem" (po prawej stronie). Status zadania można zmienić w aktywności prezentującej szczegóły zadania wciskając przycisk „WYKONANE".



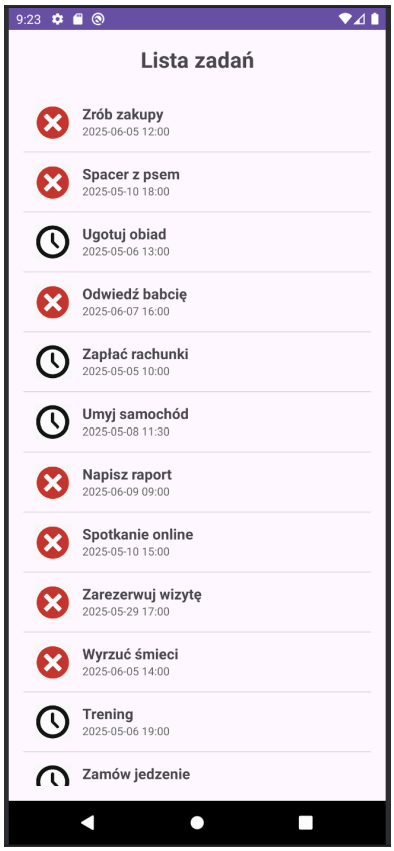
Rys. 1: Widok główny - lista zadań do wykonania.



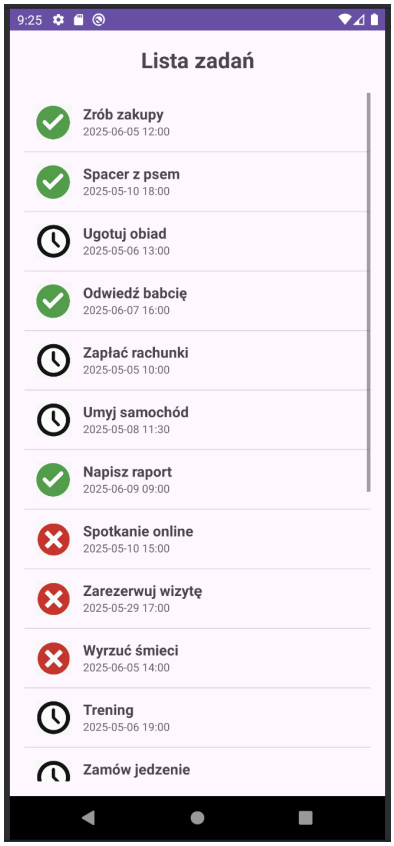
Rys. 2: Szczegóły zadania "Spacer z psem" wyświetlone po kliknięciu w zadanie.

3.2. Zmiana statusu zadań na ”wykonane”

Poniżej po lewej stronie lista zadań do wykonania, w której żadne z zadań nie ma statusu ”wykonane”. Po prawej stronie lista zadań do wykonania z wykonanymi zadaniami: „Zrób zakupy”, „Spacer z psem”, „Odwiedź babcię” i ”Napisz raport” w wyniku klikania przycisku „wykonane” w aktywności pokazującej szczegóły tych zadań.



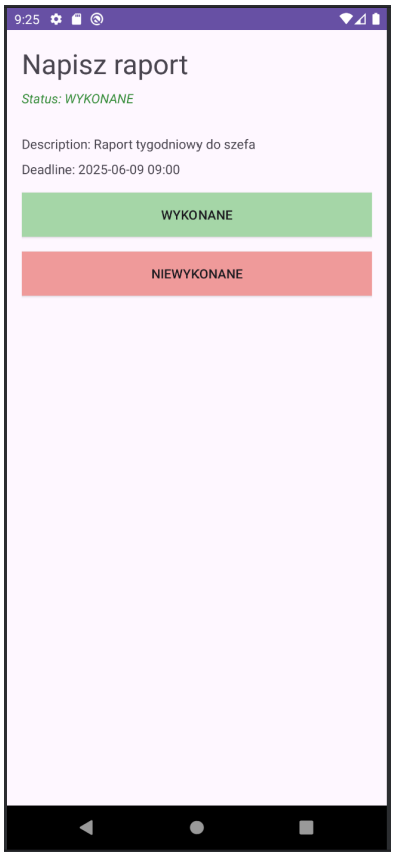
Rys. 3: Widok główny - lista zadań do wykonania, brak wykonanych zadań.



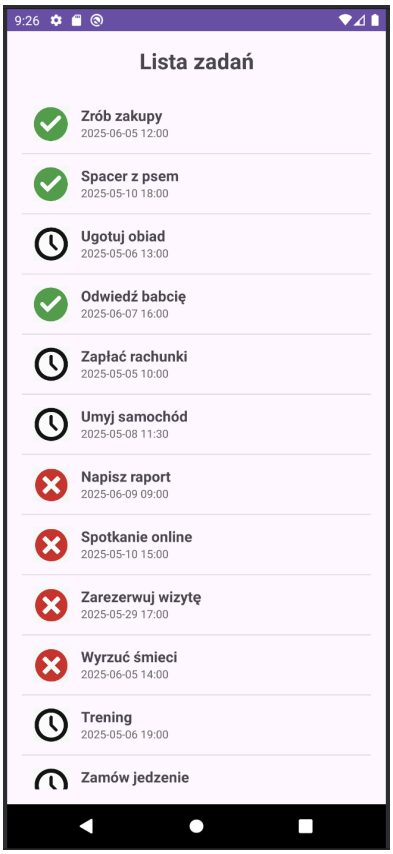
Rys. 4: Widok główny - lista zadań do wykonania, 4 wykonane zadania.

3.3. Zmiana statusu zadań na ”niewykonane”

Zadaniu o statusie „wykonane” można zmienić status na „niewykonane” poprzez wybranie go z listy, co spowoduje uruchomienie aktywności prezentującej szczegóły i w tym widoku naciśnięcie przycisku „NIEWYKONANE”. Kliknięcie w jeden z dwóch przycisków spowoduje zmianę stanu zadania i powrót do głównej aktywności aplikacji. Poniżej ilustracja takiego działania – po lewej stronie widok prezentujący szczegóły zadania ”Napisz raport”, który umożliwia zmianę statusu zadania, po prawej stronie lista zadań ze zmienionym statusem zadania „Napisz raport” na „niewykonane” (w widoku z listą status zadania prezentowany jest odpowiednią ikoną).



Rys. 5: Widok szczegółów zadania ”Napisz raport”.

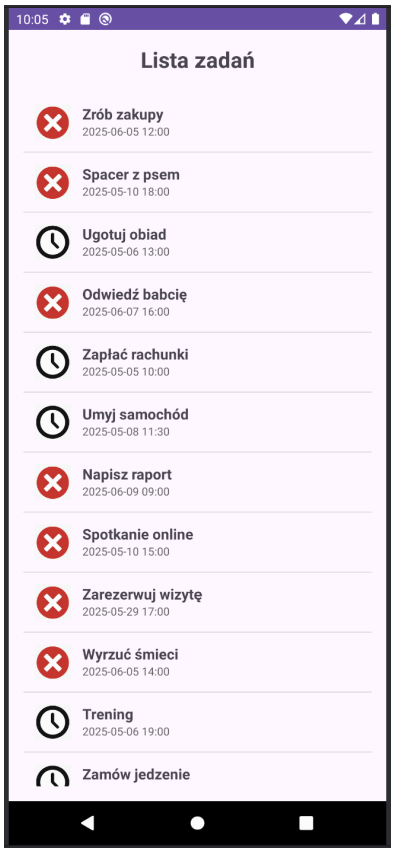


Rys. 6: Lista zadań do wykonania, status zadania ”Napisz raport” zmieniony na niewykonane”

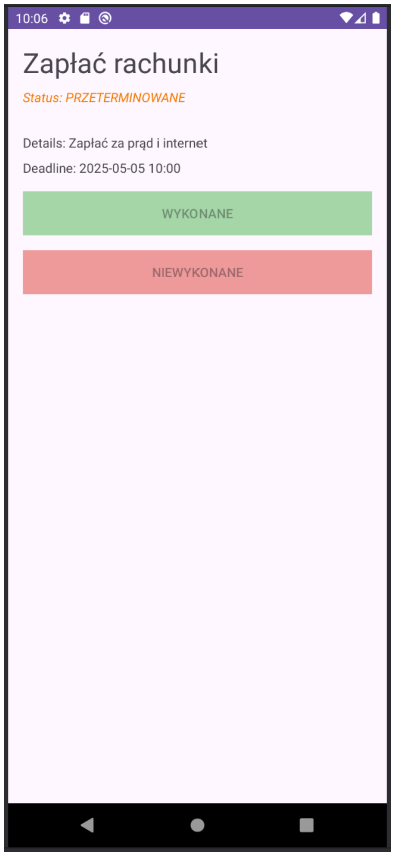
3.4. Zmiana statusu zadania na ”przeterminowane”

W widoku głównym status zadania (ikona) ustawiany jest dynamicznie przez adapter na podstawie wyniku zwróconego przez metodę isOverdue(), która sprawdza, czy aktualna data i godzina przekraczają wartość deadline’u.

W aktywności wyświetlającej szczegóły zadania status zmienia się na „przeterminowane” w momencie otwarcia szczegółów zadania, jeśli metoda isOverdue() zwróci wartość true. Jeśli zadanie jest przeterminowane, przyciski zmiany statusu są nieaktywne, a do głównego widoku aplikacji można wrócić klikając przycisk ”Back” w dolnym pasku zadań smartfona.



Rys. 7: Widok listy zadań - zadania z terminem wykonania wcześniejszym niż aktualna data i czas mają status OVERDUE, co symbolizuje ikona zegara.



Rys. 8: Szczegóły przeterminowanego zadania "Zapłać rachunki".