

Programowanie dla systemu Android - instrukcja ćwiczeń 2025

Autor: mgr inż. Karol Kaluga

1 Warunki zaliczenia ćwiczeń

Aby zaliczyć z wynikiem pozytywnym ćwiczenia wymagana będzie przynajmniej **50% obecność na zajęciach oraz oddanie projektu wraz ze sprawozdaniem z projektu**. Oddanie projektu będzie na ostatnich zajęciach. Sprawozdanie powinno zawierać imię, nazwisko, nr indeksu następnie opis projektu (co zostało wykonane, jakieś screeny z aplikacji) i wnioski. Projekty będą Państwo wgrywać na platformie Moodle. W tej instrukcji będą szczegółowe wymagania co do projektu, jakie elementy będą wymagane. Na zajęciach będę tłumaczył zagadnienia i będę Państwu pomagał z problemami. Aplikacja musi się kompilować i uruchomić na symulatorze lub fizycznym urządzeniu. Aplikacja ma zawierać funkcjonalności opisane w tej instrukcji ćwiczeniowej z sekcji 6 Projekt. Proszę zapoznać się z sekcją 7 Zasady punktacji i oceniania.

2 Tutoriale i literatura

Kotlin

- YT: Kurs Kotlin 13h
- Oficjalny tutorial Kotlin

Android

- Google: Codelabs Jetpack Compose
- YT: Notes App crash course (Jetpack Compose, MVVM)

3 Dokumentacja

- Dokumentacja Kotlin
- Dokumentacja Android
- biblioteki Jetpack

4 Instalacja Java SDK

Zainstaluj Java SDK.

5 Instalacja Android Studio

Zainstaluj Android Studio. Podczas instalacji upewnij się, że opcje Android SDK, Android SDK Platform-tools, Android SDK Build-tools i Android SDK Command-line Tools są wybrane.

6 Projekt

6.1 Architektura aplikacji

Aplikację należy tworzyć w architekturze MVVM lub MVI (wybór należy do Państwa).

- MVVM
- MVVM vs MVI

6.2 Ekran logowania i rejestracji

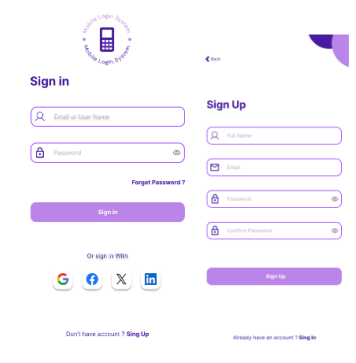
Należy zaimplementować widoki zgodne z designami udostępnionymi przez prowadzącego. Designy aplikacji są dostępne pod następującym linkiem.

Link należy otworzyć w przeglądarce. Na stronie Figma konieczne jest założenie konta, co można zrobić, logując się przez konto Google. Na Rysunku 1 przedstawione są widoki ekranów, które z Figmy należy zaimplementować. Są to ekran **Sign In** oraz **Sign Up**.

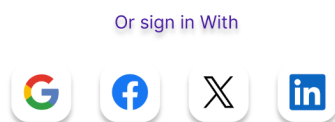
Uwaga

Nie ma potrzeby dodawania elementów z Rysunku 2 na ekranie Sign In. Pozostałe elementy ekranu nie powinny zmieniać swojego położenia na skutek braku poniższego widoku.

Po naciśnięciu klikalnego tekstu **Sign Up**, aplikacja powinna przenosić użytkownika na ekran **Sign In**. Analogicznie, naciśnięcie klikalnego tekstu **Sign In** również powinno prowadzić do ekranu logowania. Naciśnięcie przycisku **Back** ma umożliwiać powrót na ekran **Sign In**, który jest ekranem startowym aplikacji.



Rysunek 1: Ekran Sign In i Sign Up

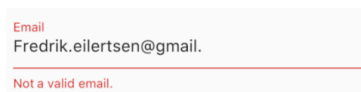


Rysunek 2: Elementy do pominięcia na ekranie Sign In

Po naciśnięciu przycisku **Sign In** użytkownik powinien zostać przeniesiony na kolejny ekran, o ile dane (np. e-mail i hasło) wpisane przez użytkownika będą zgodne z danymi zaszytymi w kodzie. Ekran ten nie jest uwzględniony w powyższym linku do Figma i na razie może być pusty - **Home Screen**. Będzie to ekran główny aplikacji w przyszłości w naszym projekcie.

Po naciśnięciu przycisku **Sign In** powinna następować walidacja wpisanych danych w polach do wpisanie e-maila, hasła itp. Należy w swoim kodzie uwzględnić przykładowe dane: e-mail oraz hasło, które będą służyły do logowania lub rejestracji.

W przypadku wpisania nieprawidłowych danych aplikacja ma wyświetlać informację, że dane są nieprawidłowe. Jeśli na przykład dane wpisane w polu e-mail będą błędne, pod tym polem należy wyświetlić tekst z napisem: Invalid email. Najlepiej, aby tekst był napisany małą czcionką i miał kolor czerwony. Na Rysunku 3 przedstawiona jest idea tego rozwiązania.



Rysunek 3: Idea wyświetlania walidacji

6.3 Kolejne ekrany, shared preferences, SQLite

Do naszej aplikacji musimy dołożyć następujące funkcjonalności:

1. Gdy użytkownik się loguje, aplikacja zapisuje tę informację w shared preferences. Jeśli użytkownik uruchomi aplikację, a był wcześniej zalogowany, aplikacja uruchamia się na ekranie Home. Jeśli nie był zalogowany, aplikacja uruchamia się na ekranie Logowania. Użytkownik ma możliwość wylogowania się z aplikacji po kliknięciu na odpowiedni przycisk na ekranie Home.
2. Trzeba zaimplementować widok ekranu Home - ekranu widocznego dla użytkownika po zalogowaniu. Wygląd tego ekranu według własnego uznania, ale ma być estetycznie.
3. Na ekranie Home użytkownik ma możliwość realizacji operacji CRUD, zapisywania danych do bazy danych SQLite. Może coś dodać na ekran, usunąć, zmodyfikować. Na tym ekranie musi być zrobiona scrollowalna lista. Jakie dane będą przechowywane, zależy od Państwa. Może być to np. nazwa jakiegoś zadania, opis zadania i data modyfikacji (muszą być przynajmniej trzy pola). Będzie to jeden wiersz danych w tabeli w bazie danych, które mają być wyświetlone na ekranie.
4. Użytkownik po kliknięciu na jeden z takich kafelków na ekranie ma zostać przeniesiony do ekranu modyfikacji tego pola lub ma zostać wyświetlony dialog, gdzie użytkownik będzie mógł zmienić dane lub usunąć rekord z bazy danych. Ten ekran może być wykorzystywany przy tworzeniu nowego rekordu.

6.4 Firebase/Supabase

W projekcie trzeba użyć Firebase lub Supabase, aby zapisywać dane na zewnętrznym serwerze. Mogą to być np. dane użytkownika, który się rejestruje. Później powinien być możliwy odczyt tych danych, np. podczas próby logowania użytkownika.

- Firebase
- Supabase

6.5 Autorska funkcjonalność

W projekcie trzeba zrobić swoją autorską funkcjonalność. Poniżej przykłady takich pomysłów:

- Kliknięcie na nr telefonu, np. wyświetlany na ekranie Home -> Otwarcie natywnej aplikacji telefonu
- Po kliknięciu przycisku otwarcie Google Maps z określoną lokalizacją GPS
- Po kliknięciu dodanie wydarzenia do kalendarza Google
- Odczyt danych z Google Fit, np. liczby kroków z danego dnia
- Uwierzytelnianie biometryczne - logowanie do aplikacji z użyciem czytnika linii papilarnych lub rozpoznawania twarzy
- Powiadomienia push z Firebase Cloud Messaging (FCM)

7 Zasady punktacji i oceniania

Zadanie	Punktacja
Zastosowanie architektury MVVM lub MVI	0-3
Poprawny wygląd ekranu logowania- sam wygląd	0-2,5
Poprawny wygląd ekranu rejestracji- sam wygląd	0-2,5
Nawigacja między ekranem logowania a rejestracji	0-1
Zrobiona walidacja danych logowania i nawigacja do ekranu Home	0-1
Zapis faktu zalogowania do shared preferences i późniejszy odczyt w celu przekierowania na prawidłowy ekran	0-1
Zrobienie ekranu do wpisywania danych do wyświetlenia na ekranie Home	0-1
Zrobienie ekranu Home- sam wygląd	0-2
Operacje CRUD na bazie danych	0-3
Użycie Firebase lub Supabase	0-3
Autorska funkcjonalność	0-3
Jakość i czytelność kodu	0-1
Użycie Git i Github do wersjonowania kodu	0-1
Obrona projektu- odpowiedź na pytania	0-2
Oddanie sprawozdania jako pdf	Warunek konieczny
Przesłanie projektu jako plik .zip	Warunek konieczny
Przynajmniej 50% obecności	Warunek konieczny

Suma punktów do zdobycia: 27

Zakres punktów	Ocena
13,5 - 16 pkt.	3.0
16,5 - 19 pkt.	3.5
19,5 - 21,5 pkt.	4.0
22 - 24,5 pkt.	4.5
powyżej 24,5 pkt.	5.0

8 Wytczne: sprawozdanie, format pliku projektu

Projekt należy wgrać na platformie Moodle w formie .zip. Zipa zrobić w Android Studio: klikamy File → Export → Export to Zip File.... Zipa nazwać w formacie nazwisko_imię_indeks.

Sprawozdanie jako PDF. Powinno zawierać opis projektu (co zostało zrobione, widoki ekranów) oraz wnioski. Na samej górze imię, nazwisko, indeks.