

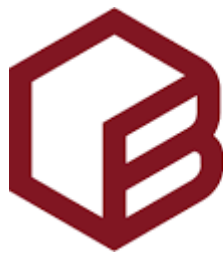
**WYDZIAŁ TELEKOMUNIKACJI, INFORMATYKI
I ELEKTROTECHNIKI**

[Programowanie urządzeń mobilnych \[05-IST-PUM-ND1\]](#)- laboratorium

SPRAWOZDANIE Z LABORATORIUM

Lab 2 Temat sprawozdania:

Aplikacja Kotlin cz 1 - Lista



**POLITECHNIKA
BYDGOSKA**
im. Jana i Jędrzeja Śniadeckich

Wykonał:

Piotrkowski Sławomir (slapio000@pbs.edu.pl)

[01-IST-PUM-ND1] rok 1 sem. 1 gr 1 stopień II

Data: 2024-11-09

1. Cel pracy

Stworzyć aplikację wyświetlającą zawartość listy z elementami String na ekranie Smartfonu z wykorzystaniem architektury Model-widok-Kontroler. Szkielet aplikacji zrealizować z użyciem komponentu Scaffold. Zaprogramować zdarzenie wyboru elementu listy, element listy wyświetlić do okna logów Log.d.

2. Implementacja

W implementacji zastosowano nowoczesne podejście z użyciem Jetpack Compose oraz struktury funkcji kompozycyjnych. Poniżej opisano kluczowe elementy aplikacji:

MainActivity:

Główna aktywność aplikacji wykorzystuje komponent Scaffold, który zarządza układem ekranu. Wewnątrz Scaffold wywoływane są funkcje kompozycyjne Greeting oraz MainContent.

Composable Functions:

- **MainContent:** Odpowiada za wyświetlanie listy elementów (*LazyColumn*) z przykładami tytułów filmów.
- **MovieRow:** Reprezentuje pojedynczy element listy w postaci karty (*Card*) zawierającej ikonę i nazwę filmu. Kliknięcie na element wywołuje zdarzenie rejestrowane w logach (*Log.d*).

Interfejs użytkownika:

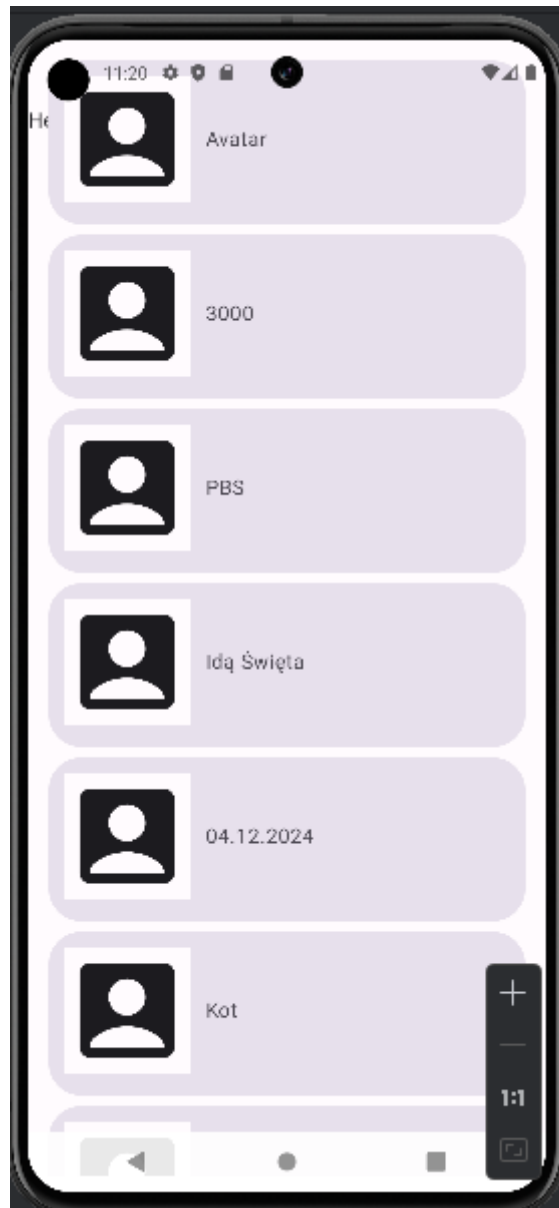
- **LazyColumn:** Efektywne renderowanie dużych zbiorów danych w postaci listy.
- **Card:** Stylizacja elementów listy z zaokrąglonymi rogami (*RoundedCornerShape*) i interaktywnością.
- **Icon i Text:** Ikona i tekst w każdym elemencie listy.

3. Wyniki

Aplikacja działa zgodnie z założeniami i wyświetla listę elementów (np. tytuły filmów) na ekranie. Kliknięcie dowolnego elementu powoduje rejestrację w logach komunikatu z nazwą wybranego filmu.

Poniżej znajdują się dodane zrzuty ekranu prezentujące:

- Ekran główny aplikacji z listą elementów.



- Przykładowy log w Logcat dla zdarzenia kliknięcia na element.

```
Pixel 8 API 30 (emulator-5554) Android 11, API 30 package:com.example.slawomirpiotrkowskilab2
2025-01-06 22:49:26.307 12268-12291 OpenGLRenderer com...mple.slawomirpiotrkowskilab2 I Davey! duration=1401ms; Flags=1, IntendedVsync=45036683702
2025-01-06 22:49:26.313 12268-12268 Choreographer com...mple.slawomirpiotrkowskilab2 I Skipped 71 frames! The application may be doing too much
2025-01-06 22:49:29.962 12268-12268 TAG com...mple.slawomirpiotrkowskilab2 0 MainContent:PBS
2025-01-06 22:49:30.111 12268-12296 ProfileInstaller com...mple.slawomirpiotrkowskilab2 0 Installing profile for com.example.slawomirpiotrkowskilab2
2025-01-06 22:49:31.094 12268-12268 TAG com...mple.slawomirpiotrkowskilab2 0 MainContent:3000
2025-01-06 22:49:31.980 12268-12268 TAG com...mple.slawomirpiotrkowskilab2 0 MainContent:PBS
2025-01-06 22:49:37.395 12268-12268 TAG com...mple.slawomirpiotrkowskilab2 0 MainContent:Idą Święta
2025-01-06 22:49:38.039 12268-12268 TAG com...mple.slawomirpiotrkowskilab2 0 MainContent:PBS
2025-01-06 22:49:38.955 12268-12268 TAG com...mple.slawomirpiotrkowskilab2 0 MainContent:3000
2025-01-06 22:49:43.257 12268-12268 TAG com...mple.slawomirpiotrkowskilab2 0 MainContent:Idą Święta
womirPiotrkowskiLab2 > app > src > main > java > com > example > slawomirpiotrkowskilab2 > MainActivity.kt 115:2 (2140 chars, 83 line breaks) LF UTF-8 4 spaces
```

4. Podsumowanie

Wykorzystanie Jetpack Compose w tym laboratorium pozwoliło na zapoznanie się z nowoczesnymi technologiami do tworzenia interfejsów użytkownika w Androidzie. Modularność funkcji kompozycyjnych umożliwiła czytelne oddzielenie logiki od interfejsu, co zwiększa skalowalność projektu.

Dzięki aplikacji nauczyłem się korzystać z komponentów takich jak *LazyColumn*, *Card* oraz systemu zdarzeń w *Jetpack Compose*, co stanowi solidną podstawę do rozwijania bardziej zaawansowanych aplikacji.

Kod Źródłowy został dołączony do sprawozdania.