Wstęp:

Problemy do rozwiązania:

* Zmienny system wag – gdy ktoś wybiera płeć męską/żeńską waga przyznawanych punktów za odpowiedź zmienia się
* Cache’owanie zaznaczonych odpowiedzi – problem związany z systemem Android, najczęściej przy przeciążeniu systemu lub obrocie ekranu niszczone jest Activity wraz ze wszystkimi zebranymi w nim informacjami
* Dynamiczne tworzenie interfejsu graficznego zależnego od przekazanej listy pytań
* Śledzenie zaznaczonych pytań
* Stworzenie animacji wypełniania mózgu
* Dynamiczne wyświetlanie ekranu końcowego w zależności od wybranego testu
* Wyświetlanie innych odcieni kolorów w zależności od pory dnia (dzień/noc), tak aby użytkownik nie musiał przemęczać swoich oczu
* Ograniczone zasoby systemowe w telefonie
* Dostęp do funkcji w bibliotece standardowej/frameworku Androida w zależności od używanej wersji androida

Wykorzystane narzędzia i technologie:

* Środowisko Programistyczne - **Android Studio**
* Genrowanie stylu i kolorów z Material3 <https://material-foundation.github.io/material-theme-builder/#/custom>
* Git – system wersji kontroli
* Github – repozytorium zdalne w którym będzie przechowywany projekt
* Jetpack Navigation – biblioteka upraszczająca nawigację z automatycznym cache’owaniem bieżącego położenia w stosie
* Epoxy – biblioteka do tworzenia bardziej zaawansowanego interfejsu graficznego w sposób dynamiczny
* Kotlinx Parcelize – biblioteka do automatycznej parcelizacji klas
* Potencjalnie Room do przechowywania danych na temat wyniku, prawdopodobnie użyję Datastore
* MVVM (Model, View, ViewModel) – sposób komunikacji UI i cache’owania danych
* Kotlin Flows
* Dagger Hilt – biblioteka do wstrzykiwania zależności (DI)
* Gradle – Budowanie aplikacji, wersjonowanie aplikacji, zarządzanie bibliotekami w projekcie, konfiguracja projektu
* Pliki XML – zarządzanie stałymi i wartościami, co pozwala na zmienianie ich w zależności od pory dnia, języka itp.

Prezentacja powstałego projektu:

Wnioski: