

# Aplikacja EntertainMatch – Raport końcowy

Adrian Bednarz, Bartłomiej Dach

11 czerwca 2017

## 1 Opis aplikacji

Aplikacja EntertainMatch przeznaczona jest dla wieloosobowych grup znajomych. Za jej pomocą użytkownicy z grupy wspólnie wybierają wydarzenie, na które chcą się wybrać, przechodząc przez wieloetapowy proces głosowania, złożony z:

- wyboru kategorii wydarzenia (np. filmy, koncerty),
- wyboru konkretnego wydarzenia z danej kategorii (wydarzenia są wybierane ręcznie przez deweloperów w obrębie poszczególnych miast). Użytkownicy mogą przejrzeć wybrane informacje o wydarzeniach, zależne od kategorii (np. w przypadku filmów są to: zwiastun filmu, krótki opis, obsada, reżyser, czas trwania i ocena filmu wg agregatora recenzji Rotten Tomatoes);
- wyboru miejsca oraz daty dla danego wydarzenia (możliwe jest wyświetlenie za pomocą aplikacji Google Maps dokładnej lokalizacji),
- deklaracji, czy dany użytkownik może przyjść na dane wydarzenie w danym terminie.

Poszczególni użytkownicy nie znają wyborów podjętych przez innych w obrębie głosowania, aby nie sugerować się ich wyborami. Każda zmiana etapu powoduje wyświetlenie powiadomień u wszystkich użytkowników.

Po zakończeniu głosowania, wydarzenia są dodawane do odpowiedniej zakładki w głównym widoku. Dodatkowo, jeśli użytkownik zezwoli aplikacji na dostęp do terminarza, dodawane jest do niego przypomnienie na dzień, w którym dane wydarzenie ma się odbyć. Wydarzenia usuwane są z listy po zakończeniu.

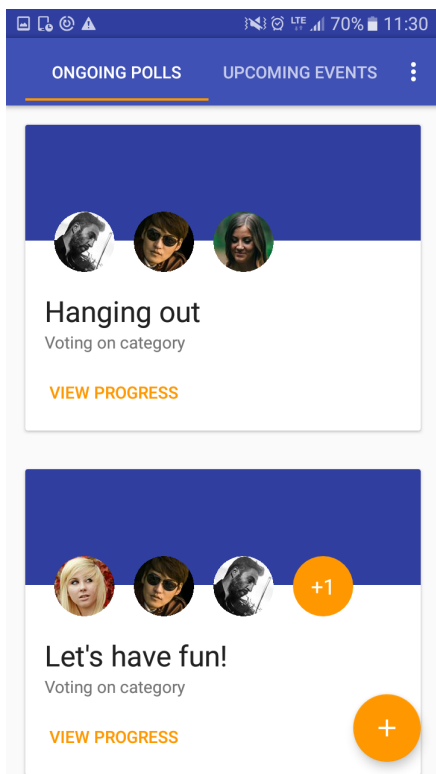
## 2 Logika biznesowa

Logowanie do aplikacji realizowane jest za pośrednictwem API Facebooka, co pozwala na bezproblemowe obsługiwanie kont użytkowników. Dodatkowo, autentykacja poprzez Facebook dobrze integruje się z bazą Firebase, co pozwala na równoczesną kontrolę nad czynnościami wykonywanymi przez poszczególnych użytkowników.

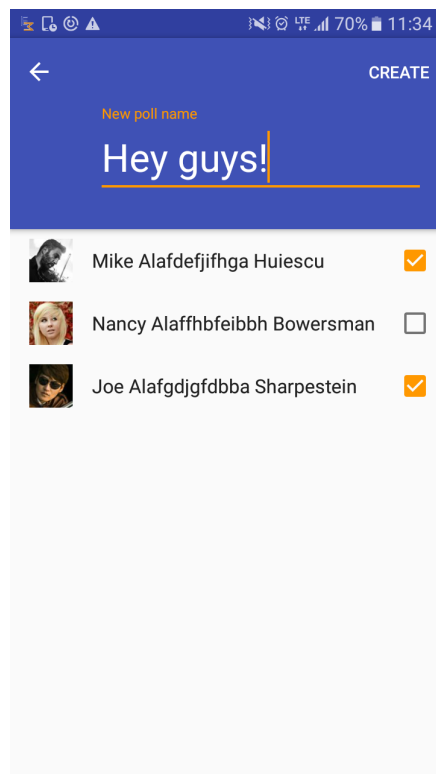
Centralnymi obiektami w aplikacji są poszczególne głosowania – ich zawartość jest uzupełniana stopniowo w poszczególnych etapach (najpierw kategoria, potem konkretne wydarzenie i data). Ze względu na wykorzystanie bazy Firebase, większość aktualizacji danych jest wykonywanych po stronie klienta, co usuwa konieczność stworzenia i utrzymywania własnego serwera aplikacyjnego. Duży nacisk położono na wsparcie współbieżności, która w takiej architekturze stanowi szczególny problem.

Dane w Firebase mają postać hierarchicznego słownika, którego kluczami są łańcuchy znaków. Format ten przypomina nieco w swojej budowie format JSON. Z tego względu schemat danych w bazie został zaprojektowany tak, aby było jak najmniej kluczy słownika, do których pisze wielu użytkowników. W ten sposób problemy z współbieżnością stają się rzadsze. W miejscach, gdzie nie można było wykluczyć wspólnych zapisów, stosowany został wbudowany w Firebase system transakcji.

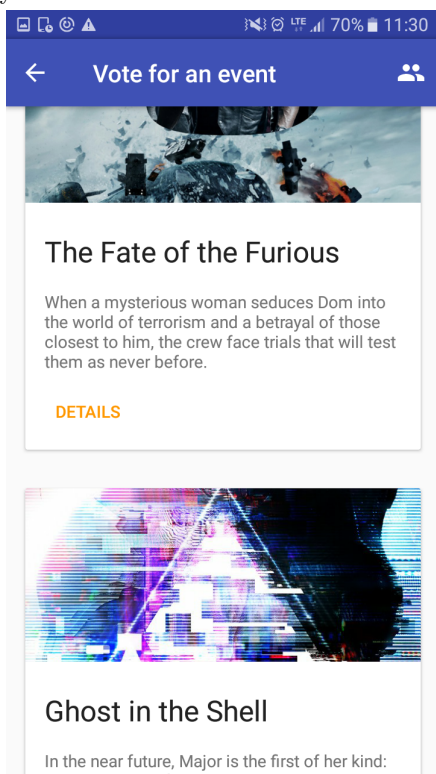
Głosowanie w poszczególnych fazach odbywa się poprzez wpisanie odpowiedniego klucza do bazy, a następnie sprawdzenia warunku przejścia do następnej fazy. W wypadku pierwszego remisu głosowanie jest powtarzane z mniejszą liczbą opcji, w przypadku kolejnych – z wygrywających opcji wybierana jest losowo jedna.



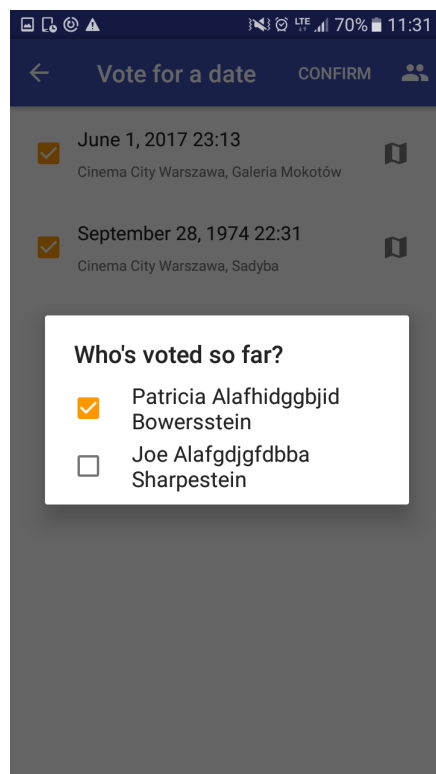
(a) Główny widok aplikacji, zawierający toczące się głosowania oraz nadchodzące wydarzenia



(b) Ekran dodawania nowego głosowania



(c) Głosowanie na etapie wyboru wydarzenia



(d) Głosowanie na etapie wyboru daty. Widoczne są opcje: podglądu danej lokalizacji dla wydarzenia oraz sprawdzenia, który z użytkowników zagłosował już w danym etapie

Rysunek 1: Przykładowe widoki interfejsu użytkownika aplikacji

W przypadku stwierdzenia zakończenia danej fazy, aktualizowana jest zawartość pola zawierającego nazwę kolejnej fazy, oraz poprzez zmianę wartości kluczy w specjalnym obiekcie wywoływane są powiadomienia dla pozostałych użytkowników. Powiadomienia te są realizowane za pośrednictwem serwisu Androida, nasłuchującego zmiany zachodzące w bazie.

Część zadań aplikacji, takie jak:

- wyświetlanie zwiastunów dla określonych wydarzeń,
- wyświetlanie mapy z położeniem danej lokalizacji,
- dodanie przypomnienia dla danego wydarzenia

są delegowane do innych aplikacji (odpowiednio: YouTube lub przeglądarka internetowa, Google Maps, Kalendarz Google) za pośrednictwem mechanizmu intencji (ang. **Intent**).

### 3 Wykorzystane narzędzia i wzorce projektowe

W aplikacji wykorzystywana jest baza danych dostarczana przez usługę Firebase [6], dostarczająca asynchroniczne aktualizacje danych do aplikacji, gdy aplikacja jest połączona z Internetem za pośrednictwem usług Google Play. Aktualizacje te obsługiwane są w aplikacji za pomocą pakietu Reactive Extensions oraz mechanizmu tzw. obiektów obserwowalnych (ang. **Observable**). Obiekty te reprezentują strumień asynchronicznych zdarzeń, na które można reagować, dodając do obserwowalnego obiektu subskrypcję.

Logowanie do aplikacji odbywa się za pośrednictwem Facebook SDK [5]. W aplikacji do głosowań można dodawać znajomych z Facebooka, którzy zainstalowali aplikację.

Część komponentów aplikacji implementuje wzorzec wstrzykiwania zależności i inwersji kontroli. W celu wstrzykiwania użyty został kontener Dagger [3]. Pozwala on na lepsze odseparowanie funkcjonalności, poprawia możliwości testowania oraz wspiera kompozycję klas.

Przy projektowaniu i implementacji interfejsu użytkownika użyto biblioteki Butter Knife [2], która pozwala wiązać pola w klasach aktywności z widokami definiowanymi w kodzie XML. Dodatkowo w celu wyświetlania obrazów z Internetu, wykorzystana została biblioteka Picasso [9], zapewniająca dynamiczne ładowanie obrazów oraz ich *caching* na dysku.

W celu ułatwienia pisania kodu, użyta została biblioteka Lombok [10], pozwalająca na automatyczne generowanie getterów, setterów i konstruktorów.

W przykładowych testach wykorzystany został framework do testowania JUnit [7], biblioteka Mockito [8] do tworzenia tzw. mocków oraz AssertJ [1] wspomagająca pisanie asercji w testach jednostkowych. Dodatkowo dołączone zostało kilka testów interfejsu, używających biblioteki Espresso [4] w celu automatyzowanego wykonywania kliknięć.

### 4 Raport z testów z użytkownikami

Poniżej znajduje się lista testów z użytkownikami wykonanych w trakcie tworzenia aplikacji. Większość z uwag została uwzględniona i poprawiona w ostatecznej wersji.

#### 1. Zalogowanie się do aplikacji

- **Opis testu:** Użytkownik po zainstalowaniu aplikacji loguje się za pośrednictwem konta w serwisie Facebook.
- **Przewidywany rezultat:** Po poprawnym zalogowaniu się następuje przejście do głównego ekranu aplikacji.
- **Wynik testu:** Pozytywny
- **Uwagi:** Brak.

#### 2. Dodanie nowego głosowania

- **Opis testu:** Użytkownik z poziomu głównego ekranu przechodzi do widoku dodawania nowego głosowania, wypełnia go i zatwierdza wprowadzone dane.
  - **Przewidywany rezultat:** W głównym ekranie aplikacji pojawia się dodane głosowanie.
  - **Wynik testu:** Pozytywny
  - **Uwagi:** Myląca ikona na przycisku dodającym głosowania, dodane głosowania pojawiały się na końcu listy, brak domyślnej nazwy głosowania.
3. Głosowanie na kategorię
- **Opis testu:** Użytkownik w głównym ekranie wybiera głosowanie znajdujące się na etapie kategorii i oddaje głos na jedną z nich.
  - **Przewidywany rezultat:** Głos zostaje zarejestrowany; jeśli głosujący był ostatni, następuje przejście do kolejnego etapu.
  - **Wynik testu:** Pozytywny
  - **Uwagi:** Kliknięcie na obrazek reprezentujący kategorię nie powoduje reakcji, brak możliwości podejrzenia głosów innych użytkowników lub sprawdzenia kto zagłosował w danym etapie.
4. Głosowanie na wydarzenie
- **Opis testu:** Użytkownik w głównym ekranie wybiera głosowanie znajdujące się na etapie wydarzeń i oddaje głos na jedno z nich.
  - **Przewidywany rezultat:** Rejestracja głosu, przejście do kolejnego etapu.
  - **Wynik testu:** Pozytywny
  - **Uwagi:** Snackbar informujący o sposobie głosowania niepotrzebnie pojawiał się za każdym razem.
5. Głosowanie na datę
- **Opis testu:** Użytkownik w głównym ekranie wybiera głosowanie znajdujące się na etapie daty i oddaje głos na jedno z nich.
  - **Przewidywany rezultat:** Rejestracja głosu, przejście do kolejnego etapu.
  - **Wynik testu:** Pozytywny
  - **Uwagi:** Brak.
6. Potwierdzenie lub odmowa potwierdzenia obecności
- **Opis testu:** Użytkownik wybiera w głównym ekranie zakończone głosowanie i potwierdza lub odmawia potwierdzenia obecności w wybranym terminie.
  - **Przewidywany rezultat:** Wybór użytkownika zostaje zapisany, głosowanie znika z listy. W przypadku zgody w zakładce z wydarzeniami dodane jest nowe wydarzenie; dodatkowo, jeśli użytkownik wyrazi zgodę, dodawane jest również w terminarzu przypomnienie związane z terminem wydarzenia.
  - **Wynik testu:** Pozytywny
  - **Uwagi:** Brak.

## Literatura

- [1] AssertJ, Joel Costigliola, <https://joel-costigliola.github.io/assertj/>, Apache 2.0
- [2] Butter Knife, Jake Wharton, <https://jakewharton.github.io/butterknife/>, Apache 2.0
- [3] Dagger, Square, <https://square.github.io/dagger/>, Apache 2.0

- [4] Espresso, Android Testing Support Library, <https://google.github.io/android-testing-support-library/docs/espresso/>, CC BY-SA 2.5
- [5] Facebook SDK for Android v4.x, Facebook for Developers, <https://developers.facebook.com/docs/android/downloads>
- [6] Firebase, Google, <https://firebase.google.com/>
- [7] JUnit, <http://junit.org/junit4/>, Eclipse Public License 1.0
- [8] Mockito, <http://site.mockito.org/>, MIT
- [9] Picasso, Square, <https://square.github.io/picasso/>, Apache 2.0
- [10] Project Lombok, <https://projectlombok.org/>, MIT