

Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej



Aplikacja przeznaczona dla osób
oglądających seriale telewizyjne

Jacek Dziwulski, Tymon Felski

29 maja 2017

Spis treści

1	Raport	2
1.1	Opis logiki aplikacji	2
1.2	Opis logiki biznesowej	5
1.2.1	Wykorzystane API	5
1.2.2	Serwer .NET WebAPI	5
1.3	Wykorzystane rozwiązania	5
1.3.1	Biblioteki wraz z określeniem licencji	5
1.3.2	Wzorce projektowe	5
2	Lista użytych skrótów	6
3	Bibliografia	6

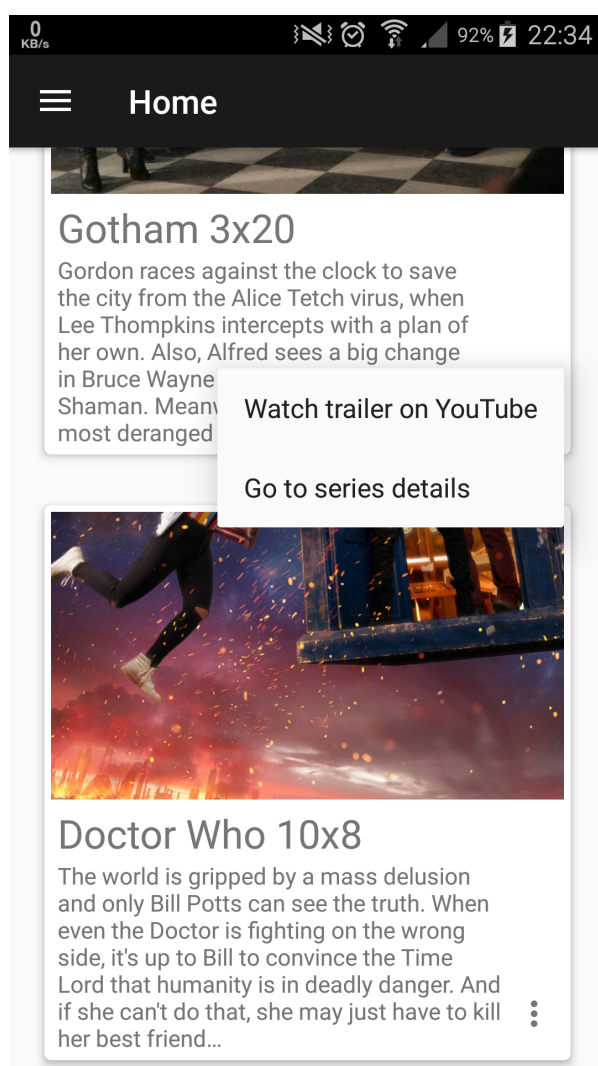
1 Raport

1.1 Opis logiki aplikacji

Stworzona w ramach projektu aplikacja **TV Tracker** jest narzędziem skierowanym do osób regularnie oglądających serie telewizyjne. Jej celem jest zarówno umożliwienie śledzenia ulubionych programów, jak i odkrywanie nowych, ponieważ na podstawie polubionych seriali, aplikacja proponuje odbiorcy nowe, podobne do nich.

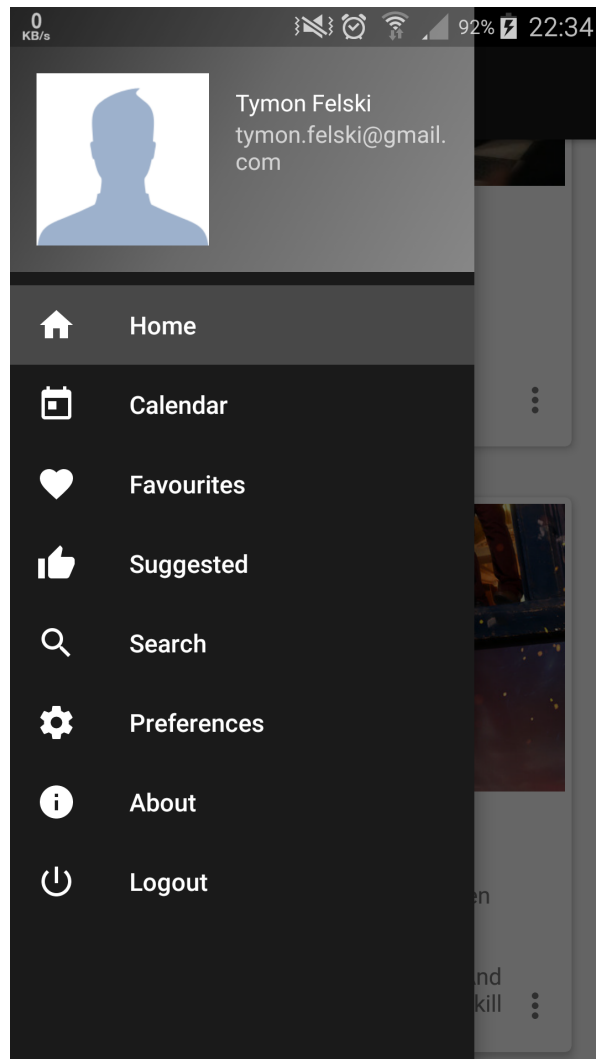
Użytkownik może się zalogować przy pomocy konta założonego na portalu **Facebook** lub swoim kontem **Google**. Przy pierwszym logowaniu zostanie utworzone konto, które będzie odpowiadać za gromadzenie informacji o serialach użytkownika.

Ekran główny aplikacji zawiera karty z nadchodzącymi odcinkami polubionych seriali. Pod zdjęciem użytkownik znajdzie krótki opis fabuły, niezdradzający zbyt wielu szczegółów, a także opcje pozwalające na przejście do opisu całego serialu lub wyszukanie zwiastunu odcinka w serwisie **YouTube**.



Rysunek 1: Ekran główny

Nawigacja w aplikacji odbywa się za pomocą panelu bocznego, w którym poza wszystkimi zakładkami znajdziemy także informacje o zalogowanym użytkowniku - jego zdjęcie, imię i nazwisko oraz adres email. Wybranie jednej z zakładek spowoduje zmianę widoku poprzez podmianę fragmentu.

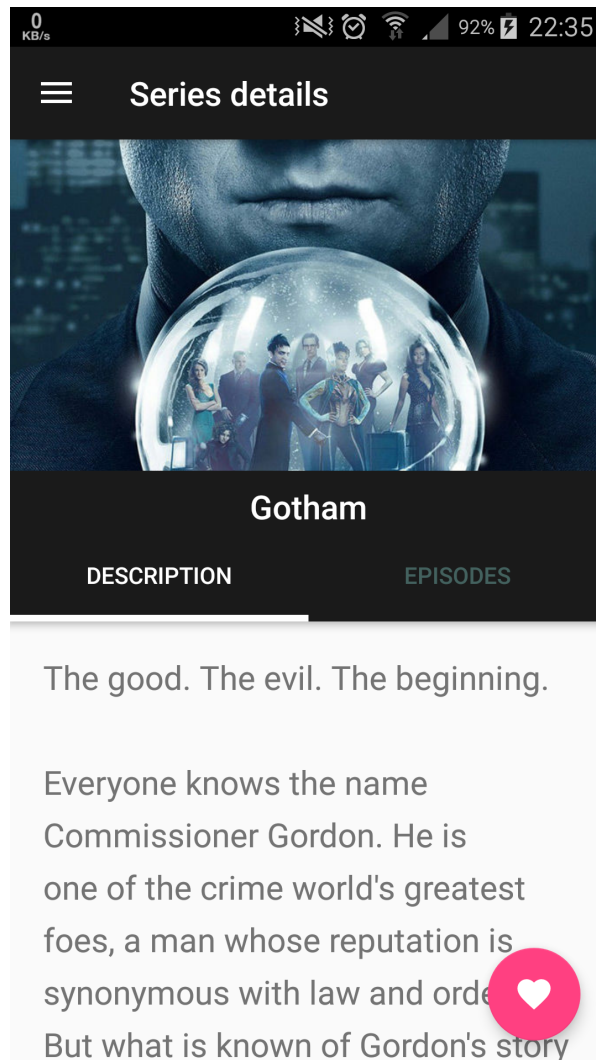


Rysunek 2: Panel boczny

Aby znaleźć nowe serie, należy wejść w zakładkę **Suggested**, w której pojawi się dwadzieścia seriali najlepiej pasujących do obecnie polubionych lub najwyżej oceniane, jeżeli użytkownik nie dodał żadnych do ulubionych. Można także skorzystać z zakładki **Search**, gdzie po wpisaniu w pole tekstowe przynajmniej trzech znaków pojawią się serie spełniające kryteria wyszukiwania.

Ustawienia aplikacji przewidują możliwość ograniczenia transferu danych w jej obrębie poprzez wyłączenie pobierania obrazków lub zastrzeżenie, że mogą być one pobierane tylko przy pomocy Wi-Fi.

Zakładka **Favourites** zawiera polubione przez użytkownika seriale. Można je stąd usunąć lub wejść w dokładny opis jednego z nich, w którym znajdziemy także wszystkie dotychczasowe odcinki. Użytkownik ma możliwość zaznaczania obejrzanych odcinków. Możliwe jest także wyświetlenie dokładnego opisu danego odcinka poprzez kliknięcie w kartę na ekranie głównym lub przytrzymanie nazwy odcinka na wcześniej wspomnianej liście w opisie serialu.



Rysunek 3: Opis serialu

W aplikacji znajduje się także kalendarz, w którym można znaleźć wszystkie odcinki z polubionych seriali. Aplikacja wyśle użytkownikowi powiadomienie przypominające o nadchodzącym odcinku nie później niż pół godziny przed jego rozpoczęciem.

1.2 Opis logiki biznesowej

Logika biznesowa

1.2.1 Wykorzystane API

Wszystkie dane wykorzystywane przez aplikację pochodzą z API udostępnianego przez serwis TVmaze. Jest to strona zawierająca informacje oraz najnowsze wiadomości dotyczące programów telewizyjnych. Udostępniane API jest darmowe, dostępne na licencji CC BY-SA, co oznacza, że adres do tej strony musi jawnie pojawić się w aplikacji (zakładka **About**). Liczba zapytań jest limitowana (według zapewnień, możliwe jest wykonanie co najmniej 20 zapytań na 10 sekund, górna granica nie jest ustalona, ale istnieje i zależy od aktualnego obciążenia serwera). Z tego powodu, utworzony przez nas serwer kopiuje wybrane dane do własnej bazy, aby nie było konieczne ciągłe wysyłanie zapytań do API. Raz w tygodniu, przechowywane dane są aktualizowane, aby były zgodne z udostępnianymi przez API.

1.2.2 Serwer .NET WebAPI

Serwer został napisany w technologii .NET, wykorzystując framework ASP.NET WebAPI. Przy uruchomieniu serwera, tworzona jest nowa baza danych, jeśli jeszcze nie istnieje. Następnie, zapisywane są do niej wybrane dane z wykorzystywanego API. Efekty wszystkich operacji wykonywanych przez użytkowników z poziomu aplikacji również są zapisywane w tej bazie. Każdy użytkownik ma przypisany identyfikator, który pozwala mu na modyfikowanie danych powiązanych z jego kontem. Cała komunikacja pomiędzy serwerem a klientem, w której częścią wiadomości jest ten identyfikator, odbywa się przy wykorzystaniu protokołu **https**, co uniemożliwia uzyskanie identyfikatora użytkownika poprzez podsłuchiwanie wiadomości.

1.3 Wykorzystane rozwiązania

Wstęp do rozwiązań

1.3.1 Biblioteki wraz z określeniem licencji

Przy tworzeniu projektu zostały użyte następujące biblioteki:

Nr	Komponent i wersja	Opis	Licencja	
1	Android Week View, 1.2.6	Kalendarz pozwalający na wyświetlanie wydarzeń	Apache	[1]
2	Floating Search View, 2.0.3	Wyszukiwarka wyświetlająca podpowiedzi	Apache	[2]
3	Butterknife, 8.5.1	Biblioteka pozwalająca na bindowanie widoków ze zmiennymi	Apache	[3]
4	Picasso, 2.5.2	Biblioteka pozwalająca na pobieranie, modyfikowanie i przypisywanie do widoków obrazów	Apache	[4]
5	Retrofit, 2.7	Biblioteka pozwalająca na wysyłanie zapytań do API	Apache	[5]
6	Gson, 2.1.0	Biblioteka pozwalająca na konwertowanie reprezentacji JSON na obiekty	Apache	[6]
7	OkHttp, 3.8.0	Klient http	Apache	[7]

1.3.2 Wzorce projektowe

Wzorce

2 Lista użytych skrótów

Apache Apache License Version 2.0, <http://www.apache.org/licenses/LICENSE-2.0>

3 Bibliografia

- [1] Android Week View, *Raquib-ul-Alam*, <https://github.com/alamkanak/Android-Week-View>
- [2] Floating Search View, *Ari C.*, <https://github.com/arimorty/floatingsearchview>
- [3] Butterknife, *Jake Wharton*, <http://jakewharton.github.io/butterknife/>
- [4] Picasso, *Square, Inc.*, <http://square.github.io/picasso/>
- [5] Retrofit, *Square, Inc.*, <http://square.github.io/retrofit/>
- [6] Gson, *Google Inc.*, <https://github.com/google/gson>
- [7] OkHttp, *Square, Inc.*, <http://square.github.io/okhttp/>