|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | |  | **Wydział Informatyki i Zarządzania**  kierunek studiów: informatyka  Praca dyplomowa – inżynierska  **Aplikacja do spersonalizowanej selekcji programów telewizyjnych**  Przemysław Zender  słowa kluczowe:  1 linia  2 linia  3 linia  krótkie streszczenie:  1 linia  2 linia  3 linia  4 linia  5 linia  6 linia   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | opiekun pracy  dyplomowej | .................................................. | | ....................... | | ....................... | | | *Tytuł/stopień naukowy/imię i nazwisko* | | *ocena* | | *podpis* | | | Ostateczna ocena za pracę dyplomową | | | | | | | | Przewodniczący Komisji egzaminu dyplomowego | | ..................................................  *Tytuł/stopień naukowy/imię i nazwisko* | | ....................... | | ....................... | | *ocena* | | *podpis* |   *Do celów archiwalnych pracę dyplomową zakwalifikowano do:\**   1. *kategorii A (akta wieczyste)* 2. *kategorii BE 50 (po 50 latach podlegające ekspertyzie)*   *\* niepotrzebne skreślić*   |  | | --- | | pieczątka wydziałowa | |
|  |  |  | Wrocław  2018 |

Spis treści

[1. Wstęp 3](#_Toc531817527)

[1.1. Uzasadnienie tematu 3](#_Toc531817528)

[1.2. Cel pracy 3](#_Toc531817529)

[1.3. Zakres prac 3](#_Toc531817530)

[2. Przegląd istniejących rozwiązań 4](#_Toc531817531)

[2.1. Standardowe programy TV 4](#_Toc531817532)

[2.2. Silniki rekomendacji 5](#_Toc531817533)

[2.3. Serwisy streamingowe 6](#_Toc531817534)

[2.4. Filmweb 7](#_Toc531817535)

[2.5. Tabela porównawcza 8](#_Toc531817536)

[2.6. Wnioski 9](#_Toc531817537)

[3. Założenia projektowe 10](#_Toc531817538)

[3.1. Opis problemu 10](#_Toc531817539)

[3.2. Wizja rozwiązania 10](#_Toc531817540)

[3.3. Wymagania funkcjonalne 10](#_Toc531817541)

[4. Projekt aplikacji 11](#_Toc531817542)

[4.1. Baza danych 11](#_Toc531817543)

[4.2. Prototyp interfejsu 11](#_Toc531817544)

[4.3. Architektura aplikacji 11](#_Toc531817545)

[5. Implementacja 12](#_Toc531817546)

[5.1. Zastosowane narzędzia 12](#_Toc531817547)

[5.1.1. Azure 12](#_Toc531817548)

[5.1.2. ASP.NET Web API 12](#_Toc531817549)

[5.1.3. Angular 6 12](#_Toc531817550)

[6. Testy 13](#_Toc531817551)

[6.1. Jednostkowe – xUnit 13](#_Toc531817552)

[7. Podsumowanie 14](#_Toc531817553)

[8. Bibliografia 15](#_Toc531817554)

# Wstęp

## Uzasadnienie tematu

Mimo rosnącej konkurencji ze strony internetu i serwisów streamingowych (jak Netflix lub Hulu), telewizja wciąż jest popularną formą rozrywki. Najpopularniejsze programy oglądają miliony ludzi, a gust każdego z nich jest inny. Ogromna różnorodność dostępnych opcji (ponad 400 kanałów na terenie Polski) sprawia, że znalezienie interesującego nas programu wymaga przejrzenia wielu stron pełnych spisów nadawanych audycji. W efekcie często zamiast zrelaksować się przy ulubionym programie, przełączamy po kolei kanały, licząc że w końcu trafimy na coś ciekawego.

## Cel pracy

Celem pracy jest ułatwienie użytkownikom oglądania interesujących ich programów przez opracowanie aplikacji pozwalającej na 3 kluczowe czynności:

* określenie i zapisanie swoich preferencji,
* pobranie informacji o nadawanych w najbliższym czasie programach
* rekomendacja dotycząca pozycji, które mogą uznać za interesujące na podstawie preferencji i aktualnego programu.

## Zakres prac

W przewidzianym zakresie projektu mieści się zbieranie danych dotyczących nadawanych programów oraz preferencji użytkowników, jak również wykorzystanie zebranych danych do przekazania użytkownikom rekomendacji.

Dane dotyczące programów pochodzą z dostępnych w internecie zasobów zebranych z użyciem programu do web scrapingu. Zebrane dane są później poddane analizie w celu wyznaczenia cech charakterystycznych poszczególnych programów i określenia podobieństwa między nimi. Szczegółowy opis wyróżnianych cech znajduje się w dalszej części pracy. Dodatkowo w ramach programu telewizyjnego dostępne są opisy poszczególnych pozycji – na podstawie zawartych w nich słów kluczowych można określić tematyczne podobieństwo między programami.

Pewnym szczególnym przypadkiem są seriale. Różnią się od większości pojedynczych audycji tym, że istotna jest kolejność odcinków, więc nie można rekomendować użytkownikowi oglądania wszystkich odcinków w przypadkowej kolejności. Konieczna jest identyfikacja numeru odcinka i ustalenie który odcinek użytkownik oglądał ostatnio i rekomendowanie tylko jednego, kolejnego, odcinka.

Dane dotyczące swoich preferencji użytkownik wprowadza sam, oznaczając poszczególne seriale, pojedyncze programy, kanały telewizyjne oraz interesujące go cechy charakterystyczne (aktorzy, drużyny, kategorie itd.) jako ulubione. Ponadto, ponieważ nawet najlepsza rekomendacja nie ma sensu, jeżeli użytkownik w danym czasie nie ma dostępu do telewizora, bardzo ważne są godziny w jakich użytkownik może i chce spędzić swój czas oglądając rekomendowane programy. Na podstawie tak zdefiniowanego profilu użytkownika można zaoferować rekomendacje zgodnych z nim programów.

O oferowanych rekomendacjach możemy informować uzytkownika na dwa sposoby. Pierwszym jest standardowy interfejs aplikacji, który jednak wymaga, żeby użytkownik w danej chwili aktywnie z niej korzystał. Drugim sposobem są powiadomienia *push* przekazywane użytkownikowi w czasie gdy aplikacja działa w tle. Najlepiej działa to w przypadku aplikacji mobilnych, ponieważ powiadomienie można przekazać w dowolnym momencie, ale obecnie aplikacje webowe również pozwalają na wyświetlanie powiadomień w przeglądarce użytkownika.

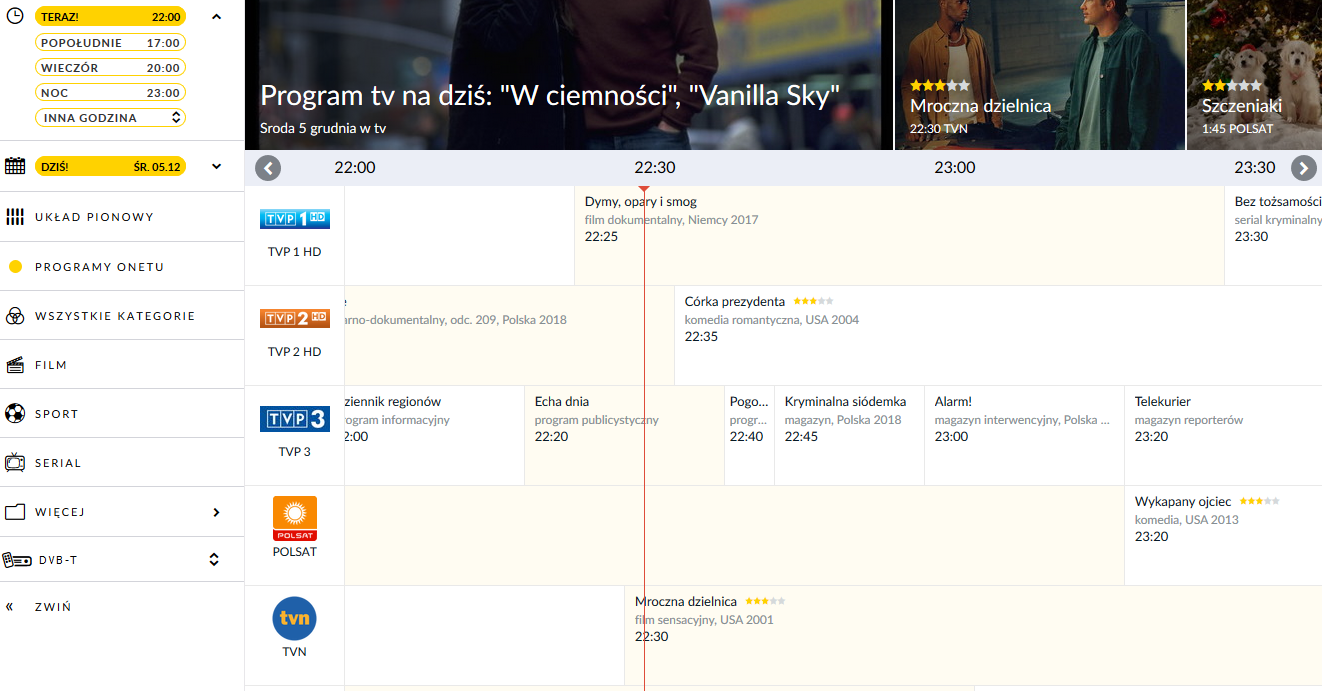
Ze względu na wieloplatformowość i dostępność zbliżonego interfejsu na urządzeniach desktopowych i mobilnych, aplikacja będzie korzystać z rozwiązań webowych. Część serwerowa zostanie wykonana w formie RESTowego API napisanego w języku *C#* z wykorzystaniem frameworka *ASP.NET Core* i umieszczona w chmurze Microsoftu, *Azure*. Część kliencka będzie wykorzystywać framework *Angular* w połączeniu z językiem *TypeScript*. Na funkcjonalność dotyczącą powiadomień pozwala zastosowanie technologii PWA (*Progressive Web Apps*) opisanej w dalszej części pracy.

Praca zawiera rozdziały skupiające się na szczegółach dotyczących kolejnych etapów procesu powstawania aplikacji, od przeglądu istniejących na rynku rozwiązań, przez przyjęte założenia projektowe i projekt, do implementacji i testów.

# Przegląd istniejących rozwiązań

Istniejące rozwiązania można podzielić na 3 główne grupy - standardowe przewodniki programowe, silniki rekomendacji i serwisy streamingowe. W dalszej części nastąpi przegląd każdej z tych grup zawierający przykłady, ich funkcjonalność i ograniczenia.

## Standardowe przewodniki programowe



* [*http://tv.wp.pl*](http://tv.wp.pl)
* [*https://programtv.onet.pl*](https://programtv.onet.pl)
* [*https://programtv.interia.pl*](https://programtv.interia.pl)

Standardowe przewodniki programowe pozwalają na przegląd aktualnie nadawanych programów. Jest to bardzo ważna, podstawowa funkcjonalność, jednakże z wielu powodów niewystarczająca. Brakuje po pierwsze, możliwości zapamiętania ustawień, a po drugie dostosowania wyświetlanych programów do preferencji. W przypadku programu dostępnego przez portal WP.pl, jedynym elementem personalizacyjnym jest zapisywany w pliku *cookie* filtr kanałów wg dostawcy, podobnie w przypadku Onetu i Interii (w 2 ostatnich dochodzi jeszcze możliwość zmiany widoku z pionowego na poziomy).

Wszystkie 3 główne przykłady oferują możliwość wyszukiwania na podstawie tytułu programu. Szukanie na podstawie cech charakterystycznych umożliwia tylko wp.pl, ale nawet tam jest dość ograniczone. Ponieważ nie zapamiętują żadnych danych poza filtrem kanałów, trzeba za każdym razem określać czego szukamy, co jest uciążliwe. Poza tym brakuje jakichkolwiek sugestii dla użytkownika który nie wie dokładnie czego szuka.

## Silniki rekomendacji

Przykłady:



* [*https://tastedive.com/shows*](https://tastedive.com/shows)
* [*https://www.cabletv.com/what-to-watch*](https://www.cabletv.com/what-to-watch)

Silniki rekomendacji są doskonałym rozwiązaniem dla niezdecydowanego użytkownika. Wymagają wpisania pozycje (w tym przypadku seriale, ale istnieją również takie systemy dla, np. filmów, książek czy muzyki), które użytkownik zna i lubi, a w odpowiedzi przedstawiają inne, podobne. Dla zestawu na przykładowym screenie aplikacja zaproponuje podobną produkcję opartą na komiksach – *The Flash*.

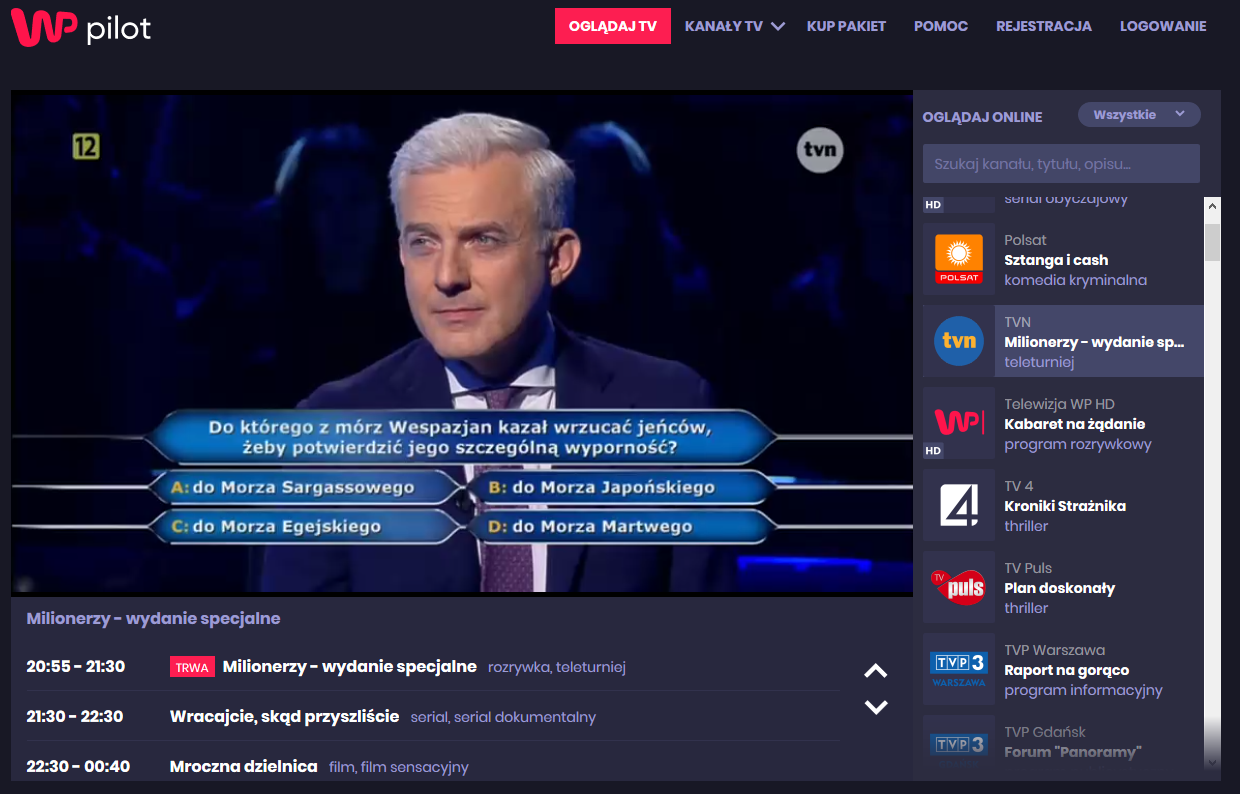
Pokazana aplikacja nie pozwala na żadną dodatkową personalizację, nie można zapamiętać rekomendacji, jedyny sposób na dokładniejsze rekomendacje to 2 przyciski – „not that one” i „start again”. Prostota użytkowania jest niewątpliwym plusem, ale brakuje wielu istotnych funkcjonalności.

Istnieją również silniki rekomendacji korzystające z serwisów społecznościowych do określenia preferencji użytkownika, ich przykładem jest *tastedive.com*. *Tastedive* działa w 2 trybach – pierwszy jest podobny do poprzedniego, ale pozwala tylko na wprowadzenie jednej pozycji (serialu, filmu, książki, artysty muzycznego lub autora). Drugi wymaga zalogowania się i pozwala oznaczać istniejące pozycje jako ulubione (jeżeli logujemy się za pośrednictwem portalu *Facebook*, *Tastedive* zbiera nasze polubienia i wyznacza rekomendacje na tej podstawie).

Problemem, który ogranicza przydatność wszystkich aplikacji z tej grupy jest to, że nie uwzględniają w żaden sposób aktualnego programu telewizyjnego. Znalezienie serialu lub filmu, który okaże się interesujący (o czym użytkownik wie dzięki rekomendacjom) nadal wymaga przeszukiwania tradycyjnych programów telewizyjnych.

## Serwisy streamingowe

Przykłady:



* [*https://pilot.wp.pl/tv*](https://pilot.wp.pl/tv)
* [*https://www.ipla.tv/start*](https://www.ipla.tv/start)
* [*https://www.popcornflix.com*](https://www.popcornflix.com)
* [*https://123vidz.com*](https://123vidz.com)

Serwisy streamingowe są mocno zróżnicowaną grupą, pod względem dostarczanej treści, kosztów użytkowania jak i kwestii prawnych.

Jednym przykładem jest oferowany przez portal *wp.pl* serwis *WP pilot* – pozwala on oglądać dokładnie to, co jest aktualnie nadawane w telewizji. Podstawowa oferta 28 kanałów jest dostępna za darmo, pełna wymaga opłat. Wyszukiwarka pozwala znaleźć kanał lub aktualnie nadawany program na podstawie tytułu lub opisu, dostępny jest też podgląd programów na najbliższe godziny. Nadal brakuje możliwości zapamiętania ulubionych kanałów lub programów, nawet po zalogowaniu, a także rekomendacji.

Przykładem z drugiej strony jest serwis *ipla.tv*. Dostarcza on ofertę filmów, seriali i programów sportowych uboższą od telewizyjnej, ale dostępną na żądanie, w większości za opłatą, choć częśc programów jest darmowa. Rekomendacje sprowadzają się jedynie do wskazania najpopularniejszych, nie są personalizowane. Podobne możliwości, poszerzone o rekomendacje, oferują serwisy takie, jak *Netflix*, *Hulu*, czy *Amazon Video.*

## Filmweb



Polski *Filmweb* jest dość unikalnym przypadkiem, który wydaje się oferować wszystko, czego można by wymagać. Przewodnik programowy, rekomendacje uwzględniające preferencje użytkownika i nadawane programyboth, możliwość wyszukiwania na podstawie różnych cech, filtrowania kanałów wg dostawców i tematycznie.

Brakuje jednak kilku elementów, po pierwsze programów sportowych, popularnonaukowych czy ogólnie niezwiązanych ze światem filmów. Nie powinno to być zaskoczeniem, biorąc pod uwagę to, że Filmweb od początku skupiony był, jak sama nazwa wskazuje, na filmach, częściowo także na serialach. Jest to jednak funkcjonalność, która ma duże znaczenie dla wielu użytkowników.

Poza tym brakuje powiadomień. Dostępne są powiadomienia dotyczące aktywności w serwisie – opublikowanych recenzji, dyskusji na forach itd., ale nic związanego z programem telewizyjnym. Poza tym, nawet te powiadomienia, które są dostępne, wykorzystują e-mail lub komunikator GG zamiast korzystania z nowszych technologii powiadomień push – przez przeglądarkę lub aplikację mobilną.

## Tabela porównawcza

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| funkcjonalność | programtv .onet.pl | tv.wp.pl | pilot.wp.pl | filmweb.pl | tastedive.com | cabletv.com | ipla.pl | popcornflix. com | moja propozycja |
| przegląd nadawanych programów | TAK | TAK | TAK | TAK | NIE | NIE | NIE | NIE | TAK |
| personalizacja kanałów | TAK | TAK | TAK | TAK | NIE | NIE | NIE | NIE | TAK |
| personalizacja treści | NIE | NIE | NIE | TAK | NIE | TAK | TAK | TAK | TAK |
| personalizacja czasu | TAK | TAK | NIE | *1 niżej* | NIE | NIE | TAK | TAK | TAK |
| wyszukiwanie programów | TAK | TAK | TAK | *2 niżej* | TAK | TAK | TAK | TAK | TAK |
| widok chronologiczny | TAK | TAK | NIE | TAK | NIE | NIE | NIE | NIE | TAK |
| rekomendacje filmów | *3 niżej* | *3 niżej* | NIE | TAK | TAK | NIE | *3 niżej* | TAK | TAK |
| rekomendacje seriali | *3 niżej* | *3 niżej* | NIE | TAK | TAK | TAK | *3 niżej* | TAK | TAK |
| rekomendacje programów sportowych | *3 niżej* | *3 niżej* | NIE | NIE | NIE | NIE | *3 niżej* | NIE | TAK |
| śledzenie odcinków seriali | NIE | NIE | NIE | TAK | NIE | NIE | NIE | NIE | TAK |
| VOD | NIE | NIE | NIE | NIE | NIE | NIE | TAK | TAK | NIE |
| streaming treści na żywo | NIE | NIE | TAK | NIE | NIE | NIE | TAK | NIE | NIE |
| powiadomienia | NIE | NIE | NIE | NIE | NIE | NIE | NIE | NIE | TAK |
| darmowy | TAK | TAK | *4 niżej* | TAK | TAK | TAK | *4 niżej* | TAK | TAK |

**Uwagi:**

1. ograniczenia w personalizacji czasu nadawania polegają na możliwości wyboru tylko określonych pór dnia (cały dzień, po południu lub wieczorem)
2. ograniczenia w wyszukiwaniu programów - można znaleźć tylko tytuły
3. ograniczenia w rekomendacjach - wyświetlane są "najpopularniejsze" programy danego typu, bez uwzględnienia preferencji użytkownika
4. *pilot.wp.pl* ma darmowy zestaw kanałów, *ipla.tv* pozwala za darmo obejrzeć tylko pojedyncze programy (głównie informacyjne). *popcornflix.com* jest darmowy, ale dostępne treści są ograniczone do tych mniej popularnych

## Wnioski

Funkcjonalność dotyczącą rekomendacji i programu w pewien sposób łączy serwis Filmweb (*<https://www.filmweb.pl>*), jednak skupia się on tylko na filmach i serialach. W związku z tym nie uwzględnia programów sportowych ani muzycznych, a dane dotyczące np. programów popularnonaukowych są mocno ograniczone.

Brakuje na rynku rozwiązań oferujących rekomendacje z uwzględnieniem aktualnego programu. Legalnie działające aplikacje dostarczające treści (w formie VOD lub strumienia na żywo) najczęściej są płatne (co nie powinno dziwić ze względu na koszty samej treści). Z tego powodu implementacja tej funkcjonalności nie mieści się w zakresie pracy.

Wszystkie rozwiązania pozwalające na wyszukiwanie mają jeden problem, polegający na tym, że wymagają poprawnie wprowadzonych tytułów, nie są w stanie skorygować błędów użytkownika. Literówka w tytule, oznacza że wyszukiwarka nie zwróci żadnych wyników, mimo że istnieją tytuły podobnie brzmiące, które lepszy algorytm byłby w stanie wychwycić. Jest to szczególnie istotne w przypadku aplikacji skupionych na rekomendacjach.

# Założenia projektowe

## Opis problemu

Obecnie nie ma na rynku żadnego rozwiązania, które pozwalałoby stwierdzić który z aktualnie nadawanych programów zainteresowałby użytkownika. Takie rozwiązanie jest potrzebne, ze względu na dużą liczbę możliwości, która sprawia że ręczne szukanie jest niewygodne, a wielu użytkowników szuka „czegoś do oglądania na wieczór”, bardziej niż konkretnego programu.

## Wizja rozwiązania

Aplikacja ma łączyć w sobie rozwiązania z silników rekomendacji i standardowych programów TV. Dzięki temu użytkownicy będą mogli dowiedzieć się jaki interesujący ich program jest nadawany, w momencie kiedy jest nadawany. To pozwoli im skupić się na oglądanym programie, zamiast na poszukiwaniach i spędzić więcej czasu na oglądaniu.

Po stronie klienta główny widok zawiera chronologiczny wykazu rekomendowanych programów nadawanych w najbliższym tygodniu. Wykaz uwzględnia programy powtarzane – zawiera tylko jedno wystąpienie, wybrane na podstawie godzin w sposób unikający ewentualnych konfliktów z innymi programami, a także uwzględniający godziny w jakich użytkownik ma dostęp do telewizora. Ponadto główny widok zawiera formularz do konfiguracji filtrów, pozwalający użytkownikowi ustalić godziny, kanały i zakres tematyczny. Konflikty, których nie udało się uniknąć są zaznaczone kolorem. Schematy interfejsu zostaną przedstawione w części projektowej.

Do dodatkowych widoków należy widok przewodnika programowego, zawierający wszystkie nadawane programy bez uwzględniania rekomendacji, widok wyszukiwania oraz widok pokazujący szczegóły danego programu. Z poziomu każdego z tych widoków użytkownik może zaznaczyć, że dany program mu się podobał lub nie, może też zapamiętać że chce obejrzeć dany program. Ustawione filtry czasowe i kanałowe oraz oznaczenia programów są zapisywane na serwerze na bieżąco.

Po stronie serwera, rekomendacje generowane są na podstawie cech charakterystycznych programów, do których zaliczamy:

* Osoby pracujące przy produkcji programu (aktorzy, reżyserowie, prezenterzy, scenarzyści, w przypadku transmisji sportowych – grające drużyny, w przypadku programów muzycznych – wykonawcy, itd.)
* Kategorie (dla filmów i seriali są to, np. thriller, fantasy, komedia, dokument; dla sportu dyscyplina i ranga zawodów – liga, puchar)
* Zawarte w słownym opisie programu słowa kluczowe

Obsada i kategoria są znane dzięki danym zebranym z programów telewizyjnych. Słowa kluczowe są dobierane automatycznie. Pozwoli to na wychwycenie tematycznych podobieństw, które nie są widoczne w zdefiniowanych cechach (jak kategorie czy obsada).

Tytuły seriali można ustalić na podstawie wspólnych części początkowych w tytułach odcinków. Łatwo to zauważyć patrząc na tytuły przykładowego serialu krymiminalnego:

* *Castle -ost.*
* *Castle: Kumple z funduszu hedgingowego. s01e03*
* *Castle: W piekle furia nie gości. s01e04*
* *Castle: Der Scharfschütze*
* *Castle. s01e09*
* *Castle . s08e09*

Wszystkie odcinki mają część wspólną na początku – *Castle* – poza tym część z nich zawiera w tytule numer kolejny odcinka (w notacji zawierającej również sezon). Na tej podstawie można stwierdzić do jakiego serialu należy i wykorzystać to do rekomendacji.

**Metoda generowania rekomendacji**

Rekomendacje generowane są w oparciu o ważoną sumę ocen użytkownika wyrażoną wzorem:

Gdzie:

* – program, którego współczynnika rekomendacji poszukujemy
* – programy wcześniej ocenione przez użytkownika
* – ocena wystawiona j-temu programowi przez użytkownika, wynosi 1, jeżeli użytkownik zaznaczył, że program mu się podobał, -1 jeżeli zaznaczył, że program mu się nie podobał. Do tych celów, zapisanie programu jako ten, który użytkownik chce obejrzeć, jest traktowane jako ocena pozytywna.
* – podobieństwo dwóch programów określone wg wzoru:

Gdzie – wagi dla i-tej grupy cech, – podobieństwo i-tej grupy cech danego programu. Wyróżniamy 6 grup cech:

* Gatunki
* Słowa kluczowe
* Aktorzy
* Reżyser
* Kraj produkcji
* Rok produkcji

Dla gatunków, słów kluczowych i aktorów, podobieństwo wyrażamy współczynnikiem Dice’a:

Dla kraju produkcji i reżysera podobieństwo jest binarne, wynosi 1 gdy wartości są te same, 0 w przeciwnym wypadku.

Dla roku produkcji podobieństwo wyraża się wzorem:

Jest to funkcja homograficzna, przyjmująca wartość 0, gdy argument jest równy 0, wartość gdy i asymptotycznie dążąca do wartości 1. Dla parametru przyjmujemy wartość 20 lat.

Wagi dostosowywane są indywidualnie do każdego z użytkowników. Przy rejestracji nowego użytkownika przyjmuje się wartości średnie dla wszystkich użytkowników, w przypadku ich braku – losowane z przedziału . Za każdym razem, kiedy użytkownik oceni program, dokonywana jest korekcja wag o wartość:

Gdzie

* – parametr z przedziału ,
* – ocena wystawiona programowi przez użytkownika,
* – wartość funkcji wsparcia rekomendacji dla programu

## Wymagania funkcjonalne

Gość:

* ST-001: Jako gość, chcę przeglądać program TV, żeby sprawdzić co mogę obejrzeć
* ST-002: Jako gość, chcę wybrać tylko te kanały które mnie interesują, żeby uprościć szukanie
* ST-003: Jako gość, chcę zawęzić kanały do oferty mojego dostawcy, żeby uprościć szukanie
* ST-004: Jako gość, chcę sprawdzić o której godzinie nadawany jest interesujący mnie program
* ST-005: Jako gość, chcę zarejestrować się w systemie, żeby zapisać moje preferencje

[Zarejestrowany] użytkownik

* ST-006: Jako użytkownik chcę sprawdzić kiedy nadawane są interesujące mnie programy
* ST-007: Jako użytkownik chcę otrzymywać powiadomienia o interesujących mnie programach, żeby ich nie przegapić
* ST-008: Jako użytkownik chcę zapisać filtr programów, żeby nie ustawiać go za każdym razem
* ST-009: Jako użytkownik chcę otrzymać rekomendacje programów, które mogą mnie zainteresować, żeby ułatwić wyszukiwanie
* ST-010: Jako użytkownik chcę potwierdzić lub odrzucić rekomendację, żeby otrzymywać dokładniejsze wyniki
* ST-011: Jako użytkownik chcę ustawić w jakich godzinach otrzymuję powiadomienia, żeby nie otrzymywać ich wtedy, kiedy nie mam dostępu do telewizora
* ST-012: Jako użytkownik chcę śledzić kolejność odcinków wybranych seriali, żeby nie otrzymywać rekomendacji dotyczących odcinków, które już widziałem

# Projekt aplikacji

## Baza danych



## Prototyp interfejsu

qq

## Architektura aplikacji

qqq

# Implementacja

## Zastosowane narzędzia

### Azure

### ASP.NET Web API

### Angular 6

# Testy

## Jednostkowe – xUnit

# Podsumowanie

# Bibliografia

**There are no sources in the current document.**