



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

kierunek studiów: wpisz właściwy

specjalność: wpisz właściwą

Praca dyplomowa - inżynierska

Aplikacja mobilna dla kinomanów

Mateusz Gadzaliński

słowa kluczowe:

Android

Firebase

MVC

krótkie streszczenie:

Celem pracy jest zaprojektowanie i implementacja aplikacji mobilnej na system Android o tematyce filmowej. Zadaniem aplikacji jest ułatwienie użytkownikom wyboru filmu do obejrzenia oraz gromadzenie biblioteki ocenionych filmów. W pracy zawarte zostały istniejące konkurencyjne rozwiązania, wymagania funkcjonalne i нефункционалне, przypadki użycia oraz najważniejsze aspekty implementacyjne.

opiekun pracy dyplomowej <i>Tytuł/stopień naukowy/imię i nazwisko</i> <i>ocena</i> <i>podpis</i>
Ostateczna ocena za pracę dyplomową			
Przewodniczący Komisji egzaminu dyplomowego <i>Tytuł/stopień naukowy/imię i nazwisko</i> <i>ocena</i> <i>podpis</i>

Do celów archiwalnych pracę dyplomową zakwalifikowano do: *

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

* niepotrzebne skreślić

pieczęć wydziałowa

Wrocław 2017

Spis treści

1	Wstęp.....	4
1.1	Wprowadzenie	4
1.2	System Android	5
1.3	Przegląd istniejących rozwiązań	11
1.4	Cel pracy	13
2	Projekt	14
2.1	Wymagania funkcjonalne i нефункционалне	14
2.2	Przypadki użycia	15
2.3	Wzorzec architektoniczny MVC.....	26
3	Interfejs użytkownika	27
3.1	Widok rejestracji i logowania	27
3.2	Widok ekranu głównego	28
3.3	Widok najlepiej ocenianych filmów i polecanych dla użytkownika	29
3.4	Widok wyszukiwania pozycji filmowych.....	30
3.5	Widok profilu użytkownika i ocenionych filmów	31
3.6	Widok filmu	32
4	Implementacja	33
4.1	Środowisko i narzędzia programistyczne	33
4.2	Składowe projektu.....	35
4.3	Problemy implementacyjne	36
5	Podsumowanie	39
	Spis ilustracji	40
	Spis tabel	41
	Bibliografia.....	42

Streszczenie

Celem pracy jest zaprojektowanie i implementacja aplikacji mobilnej na system Android o tematyce filmowej. Zadaniem aplikacji jest ułatwienie użytkownikom wyboru filmu do obejrzenia oraz gromadzenie biblioteki ocenionych filmów. W pracy zawarte zostały istniejące konkurencyjne rozwiązania, wymagania funkcjonalne i нефункционалне, przypadki użycia oraz najważniejsze aspekty implementacyjne. Założenia projektowe i implementacyjne zostały zrealizowane. Efektem zakończonej pracy jest działająca aplikacja mobilna dla systemu Android, służąca użytkownikowi jako pomoc do wyboru filmu na podstawie ocen innych użytkowników, znajomych oraz osobistego gustu użytkownika.

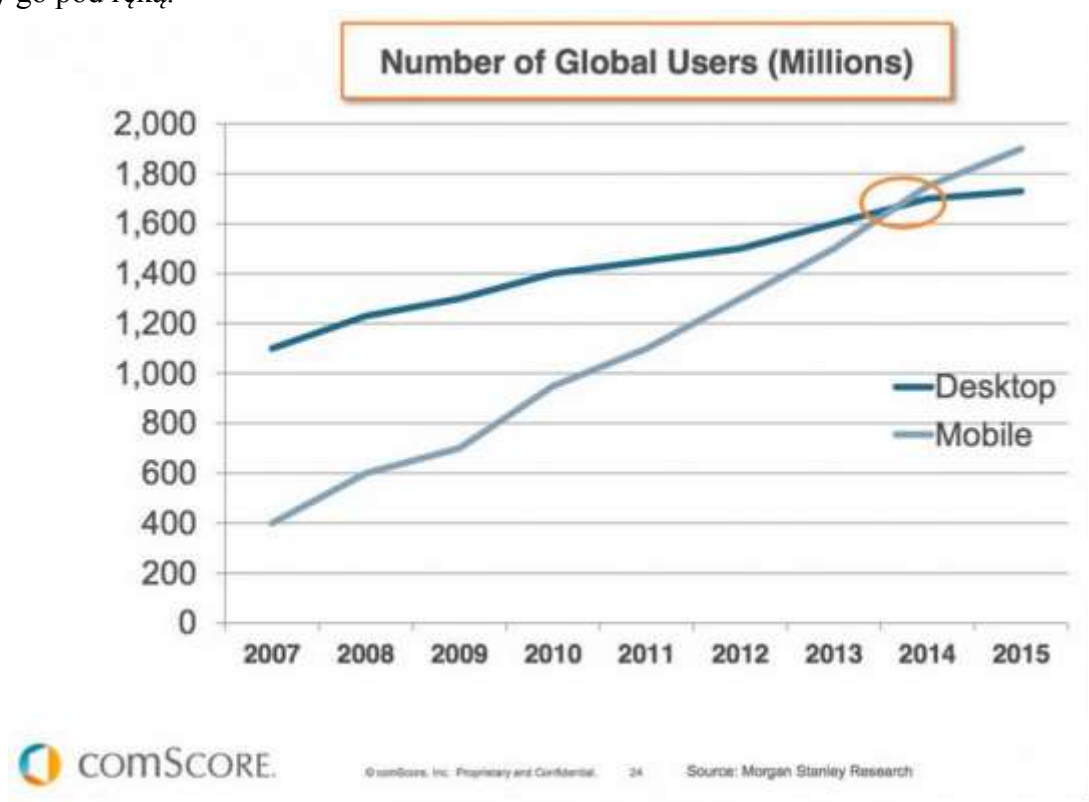
Abstract

The purpose of this thesis is to create a project and implement mobile application about movie theme for system Android. The purpose of the application is to make it easy for viewers to choose which movie to watch and collecting a library of rated movies. The work includes existing competitive solutions, functional and non-functional requirements, user cases and the most important implementation aspects. Project and implementation assumptions were realized. The end result is a working Android mobile application that serves as a help for users to choose a movie from based on ratings from other users, friends and personal taste.

1 Wstęp

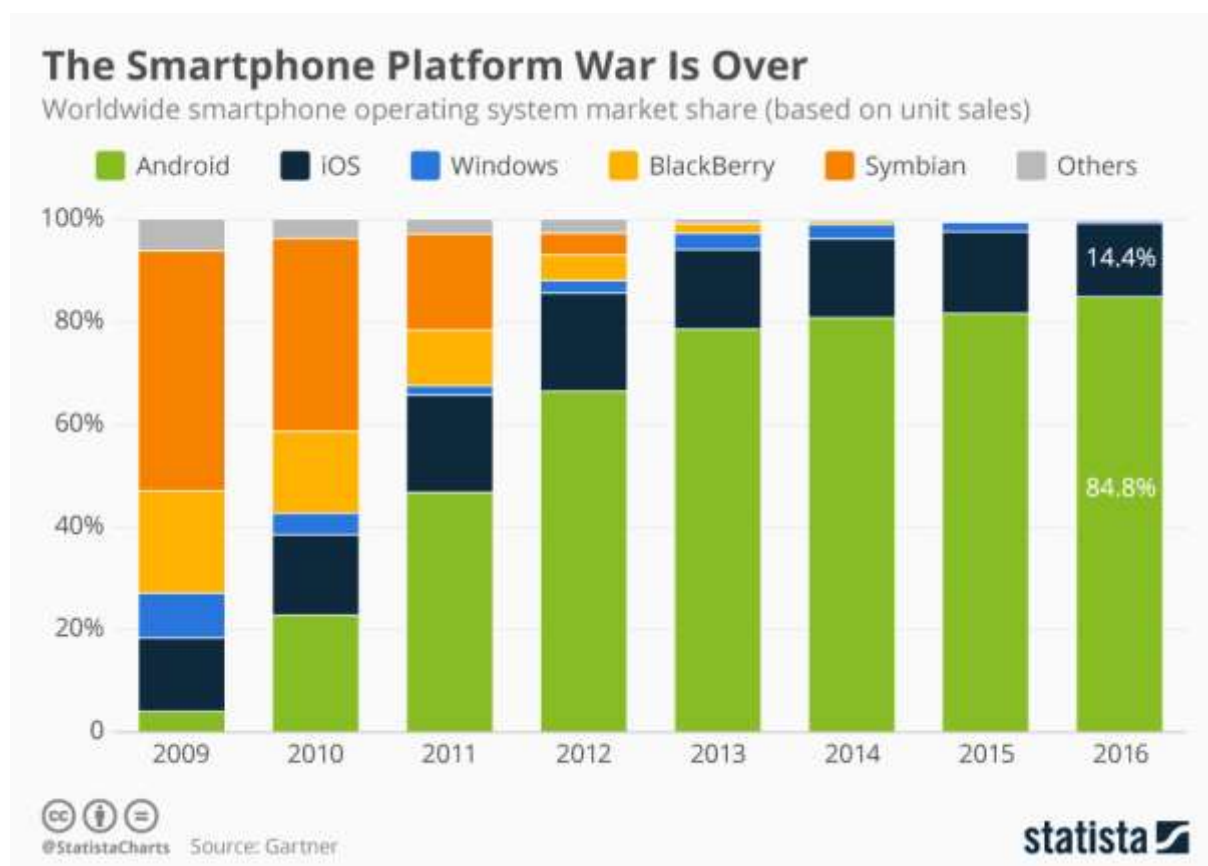
1.1 Wprowadzenie

W dzisiejszych czasach, w dobie cyfryzacji i informatyzacji bardzo popularne stały się smartfony. Powszechny dostęp do Internetu oraz postęp technologiczny doprowadził do tego, że rynek telefonów przewyższył już rynek komputerów osobistych. Prawie każdy z naszego otoczenia posiada swój własny telefon, który nie tylko służy nam do wykonywania połączeń telefonicznych, ale pomaga nam na każdym kroku w naszym codziennym życiu. Przykładem takiego rozwoju może być potrzeba szybkiego wyszukiwania informacji, nie używamy do tego tomu encyklopedii, ani nawet komputera, tylko smartfona, ponieważ prawie zawsze mamy go pod ręką.



Rys. 1.1 Liczba użytkowników smart fonów a komputerów stacjonarnych latach 2007-2015(w milionach)[1]

Na powyższym wykresie widać, że od 2014 roku liczba użytkowników urządzeń mobilnych przewyższyła liczbę użytkowników komputerów stacjonarnych i nadal utrzymuje tendencję wzrostową. Liczba posiadaczy smart fonów wciąż rośnie, producenci co roku wypuszczają coraz to nowsze modele telefonów, a co za tym idzie powstaje również coraz więcej aplikacji mobilnych. Pytanie brzmi, jaki system mobilny jest najbardziej popularny wśród użytkowników? Czy jest to Android, iOS, a może Windows?



Rys. 1.2 Procentowy udział systemów na urządzeniach mobilnych latach 2009-2016[2]

Jak wskazują dane w 2016 roku aż 84,8% urządzeń mobilnych było opartych na systemie Android, drugim wynikiem było 14,4% uzyskane przez system iOS, z czego wynika, że Android jest prawie monopolistą w tej dziedzinie. Autor pracy, tworząc aplikację chciał aby była ona dostępna dla jak największej liczby osób, dlatego po prześledzeniu statystyk, zdecydowano, że aplikacja będzie oparta na systemie Android.

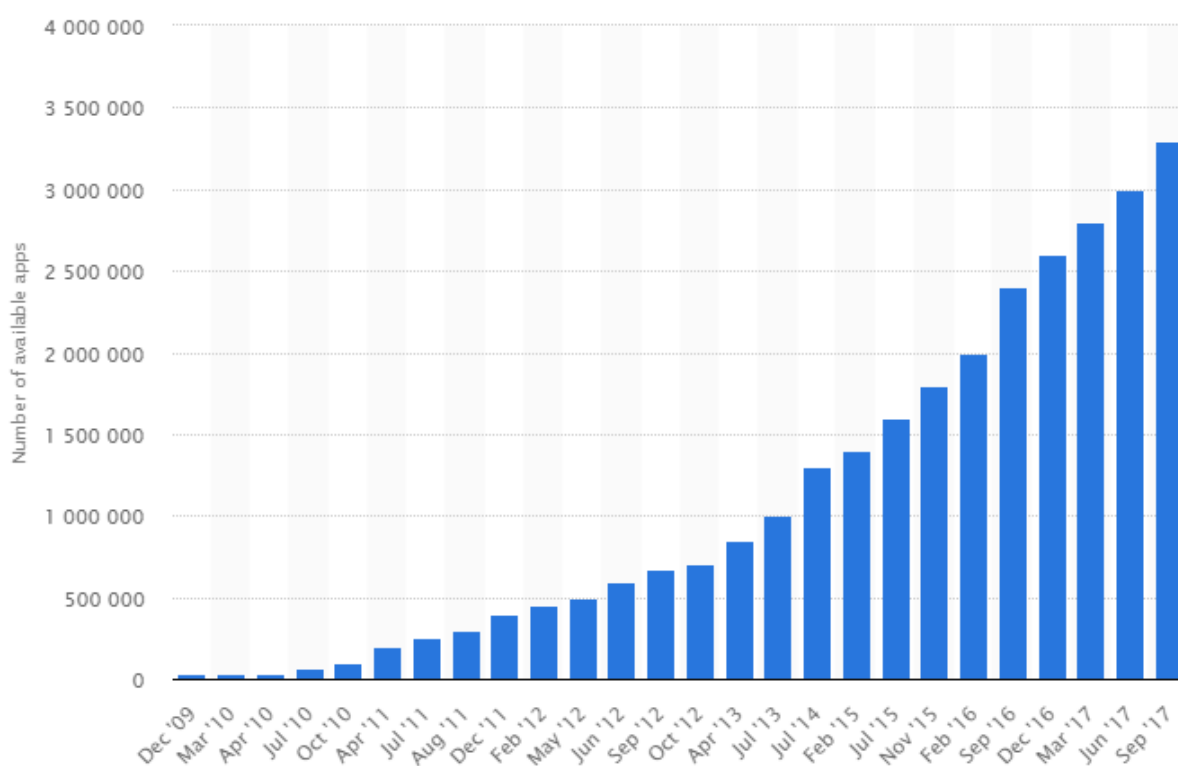
W dzisiejszych czasach bardzo ważny jest dla nas czas. Wszędzie się spieszymy, liczymy każdą minutę, mamy coraz mniej czasu na odpoczynek. Dlatego też jeśli już znajdziemy wolny czas, nie chcemy go marnować. Jednym ze sposobów spędzania wolnego czasu jest oglądanie filmów. Jakie mogą wiązać się z tym problemy? Często chcemy obejrzeć film, a nie możemy zdecydować się na konkretny tytuł, gatunek. Spędzamy dużo czasu na jego wyborze, przeglądamy Internet w poszukiwaniu, gdzie ciągle powtarzają się oglądane już przez nas tytuły. Czasem nawet przez brak wyboru rezygnujemy z przyjemności oglądania. Poniższa aplikacja ma na celu głównie pomóc użytkownikowi zaoszczędzić czas na szukaniu odpowiedniego filmu dla siebie.

1.2 System Android

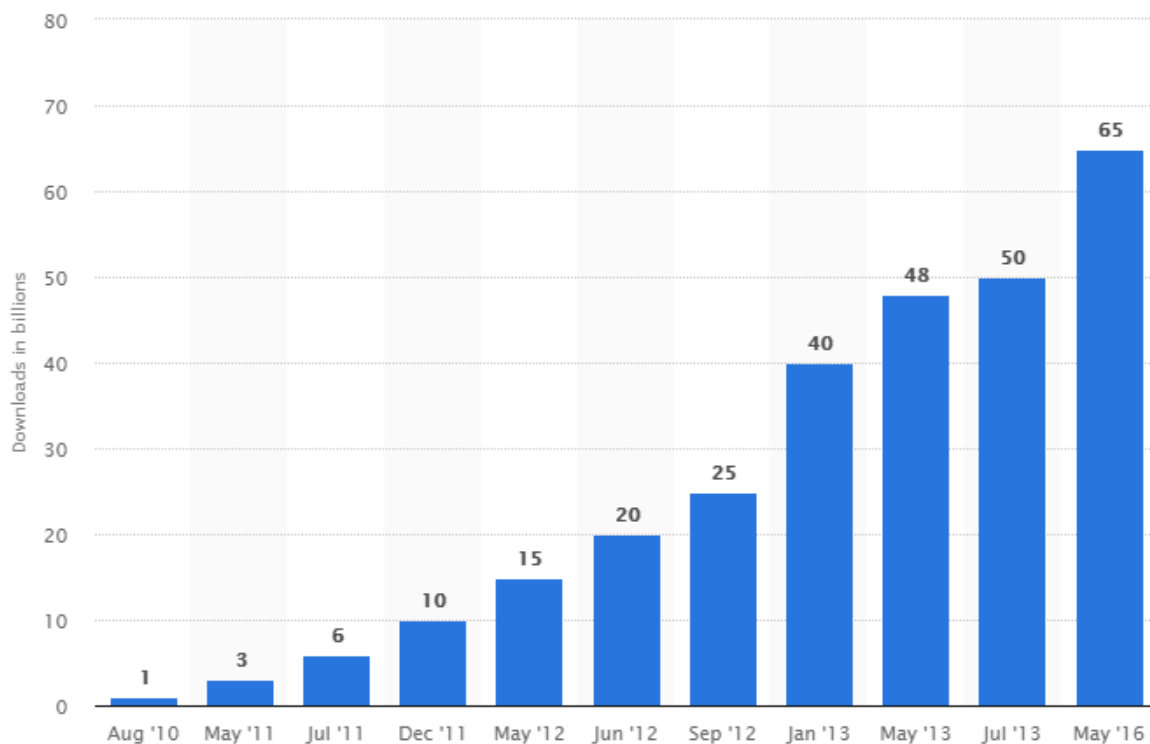
Według statystyk Android panuje na rynku systemów mobilnych. Jaka jest historia powstania systemu i w jaki sposób udało mu się odnieść taki sukces?[3] Początek miał miejsce w 2003 roku, kiedy to – Andy Rubin, Rich Miner, Chris White oraz Nick Sears postanowili założyć firmę o nazwie Android Inc. Początkowo zajmowała się tworzeniem oprogramowania dla aparatów cyfrowych. Jednak po jakimś czasie właściciele zdali sobie

sprawę, że nie był to dobry wybór ze względu na coraz większy brak zainteresowania tego typu produktami.

Kolejnym krokiem, na który zdecydowała się firma miało być stworzenie oprogramowania mobilnego, które mogłoby konkurować z dotychczasowymi popularnymi rozwiązaniami - Symbianem i Windows Mobile. W 2005 roku nastąpiło bardzo ważne wydarzenie dla systemu Android, a mianowicie Android Inc. zostało kupione za co najmniej 50 milionów dolarów przez Google. W 2007 roku Google założyło sojusz z 34 firmami (obecnie 78) z branży telekomunikacyjnej o nazwie Open Handset Alliance (OHA), mający na celu rozwój otwartych standardów dla urządzeń mobilnych. Tak działający sojusz we wrześniu 2008 roku wypuścił pierwszą komercyjną wersję systemu Android 1.0 na telefonie HTC Dream. Każde kolejne wersje systemu przyciągały do siebie coraz to większą liczbę użytkowników. Według informacji przekazanych od firmy Google, miesięcznie aktywnych urządzeń z systemem Android jest już ponad 2 miliardy oraz 1.5 miliarda ściągnięć aplikacji ze sklepu Google Play, natomiast sama liczba aplikacji znajdujących się w sklepie wynosi ponad 3 miliardy. Są to ogromne liczby i ciągle rosną, na co wskazuje rysunek 1.3 i 1.4

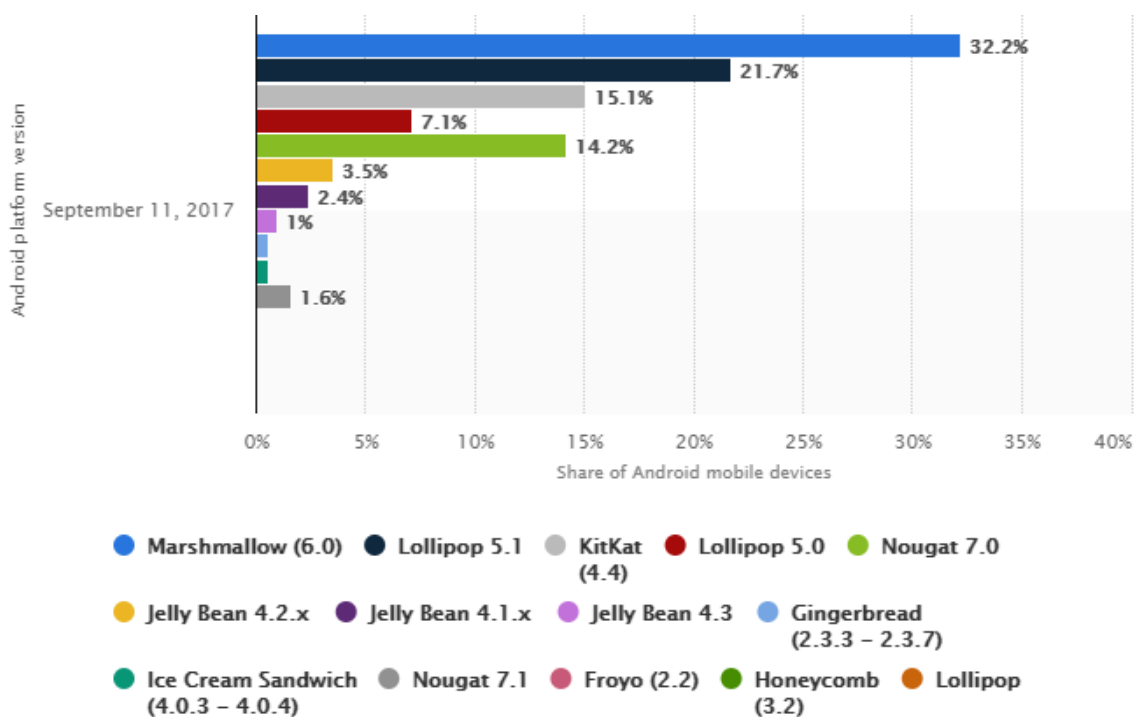


Rys. 1.3 Liczba dostępnych aplikacji w Sklepie Google Play od grudnia 2009 do września 2017[4]



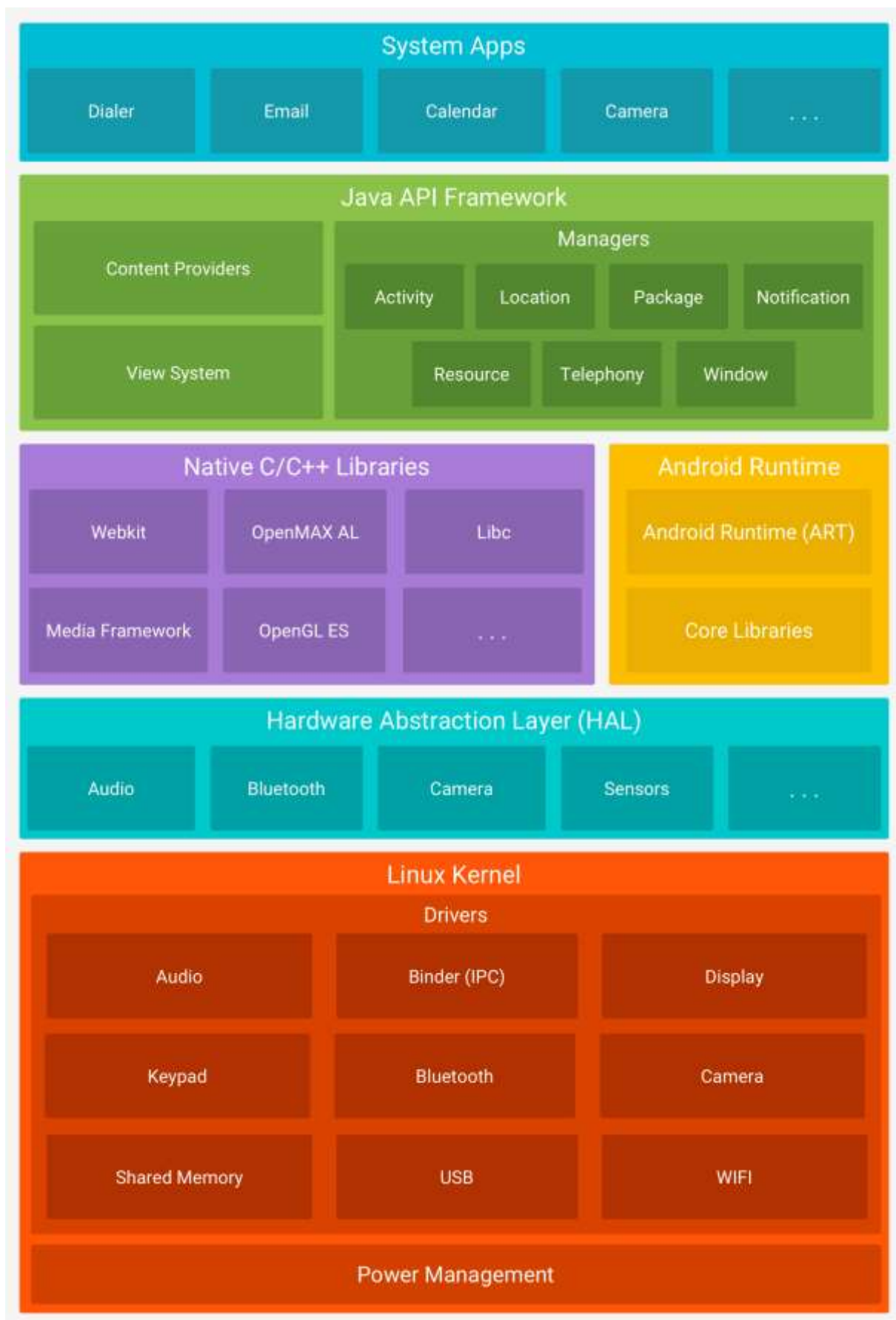
Rys. 1.4 Łączna liczba aplikacji pobranych z Google Play od maja 2016 (w miliardach)[5]

W przeciągu 9 lat od pierwszego wydania systemu zostało wydanych kilka wersji Androida, które wprowadziły coraz to nowsze funkcjonalności, nie tylko dla użytkowników, ale również dla developerów. Przed rozpoczęciem pisania aplikacji, autor musiał zdecydować, od której wersji systemu miała być kompatybilna.



Rys. 1.5 Dystrybucja wersji Androida właścicieli smartfonów – wrzesień 2017r.[6]

Jak widać na Rys.1.5 cztery najbardziej popularne wersje systemu to kolejno Marshmallow (6.0) – 32.2%, Lollipop (5.1) – 21.7%, Kitkat (4.4) – 15.1% oraz Nougat(7.0) – 14.2%. Ludzie kupują coraz to nowsze modele telefonów, na którym znajdują się nowe wersje systemu, czego skutkiem jest to, że liczba urządzeń aktywnych ze starszymi wersjami maleje, dlatego podjęto decyzję, że aplikacja będzie wspierać urządzenia z systemem od wersji 5.0 w górę.



Rys. 1.6 Architektura systemu Android[7]

Na podstawie Rys. 1.6 można zauważyć, że architektura systemu Android podzielona jest na 6 głównych warstw:

- Jądro Linuxa (Kernel) – warstwa fundamentalna. Jest odpowiedzialna za obsługę sprzętu oraz posiada do niego niezbędne sterowniki, takie jak np. kamera, klawiatura, ekran, wifi, itd. Producenci sprzętowi piszą odpowiednie sterowniki pod konkretną wersję jądra.
- Hardware Abstraction Layer (HAL) – warstwa abstrakcji sprzętowej. Tworzy ona standardowe interfejsy do komunikacji ze sprzętem, wykorzystywane przez producentów. Dzięki niej umożliwiona zostaje implementacja funkcjonalności bez wpływania i modyfikowania systemu wyższego poziomu.
- Natywne biblioteki C/C++ - podstawowe komponenty, takie jak warstwa aplikacji sprzętowej, czy Android Runtime budowane są z natywnego kodu, który wymaga natywnych bibliotek napisanych w języku C i C++.
- Android Runtime – to środowisko uruchomieniowe używane przez aplikacje i niektóre usługi systemowe. Od wersji Androida 5.0 każda aplikacja uruchamiana jest jako nowy proces, który posiada własną instancję Android Runtime.
- Java API Framework – cały zestaw funkcji systemu Android jest dostępny za pośrednictwem interfejsów API napisanych w języku Java. Dzięki tym interfejsom możliwe jest tworzenie aplikacji.
- Aplikacje systemowe – wszystkie podstawowe aplikacje systemowe, np. do obsługi kontaktów, smsów, kalendarzy, przeglądania Internetu itp.

1.3 Przegląd istniejących rozwiązań



Rys. 1.7 Widok startowy serwisu Filmweb



Rys. 1.8 Lista filmów do obejrzenia

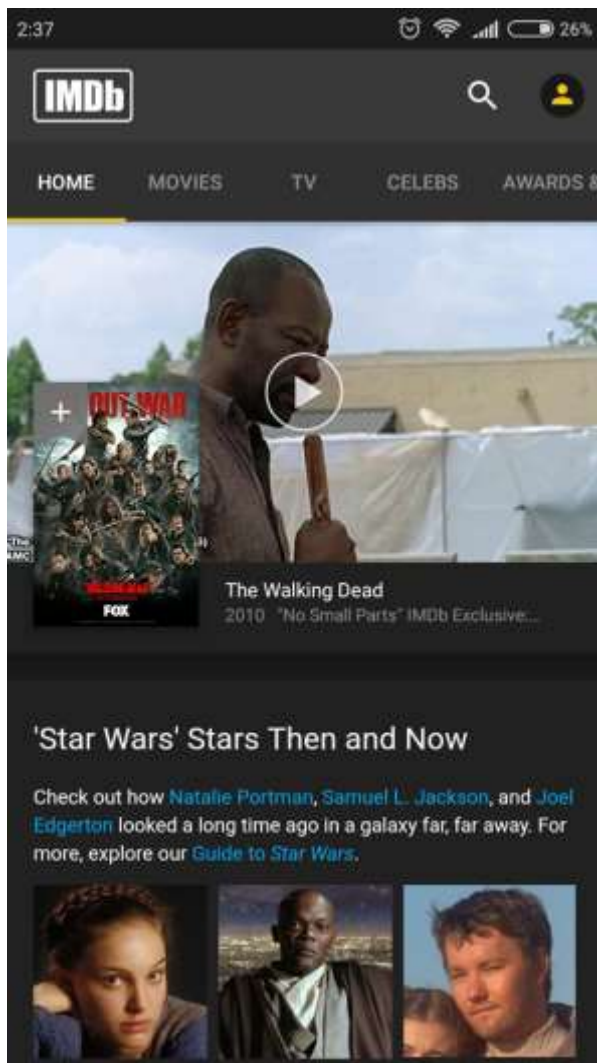
Filmweb jest największym serwisem o tematyce filmowej i kinowej w Polsce. Istnieje w sieci od ponad 19 lat. W swojej bazie posiada ponad 600 000 filmów. Jest to druga co do wielkości baza filmowa na świecie. Jest tworzony aktywnie przez użytkowników, którzy sami mogą pomagać w uzupełnianiu danych o filmach.

Zalety:

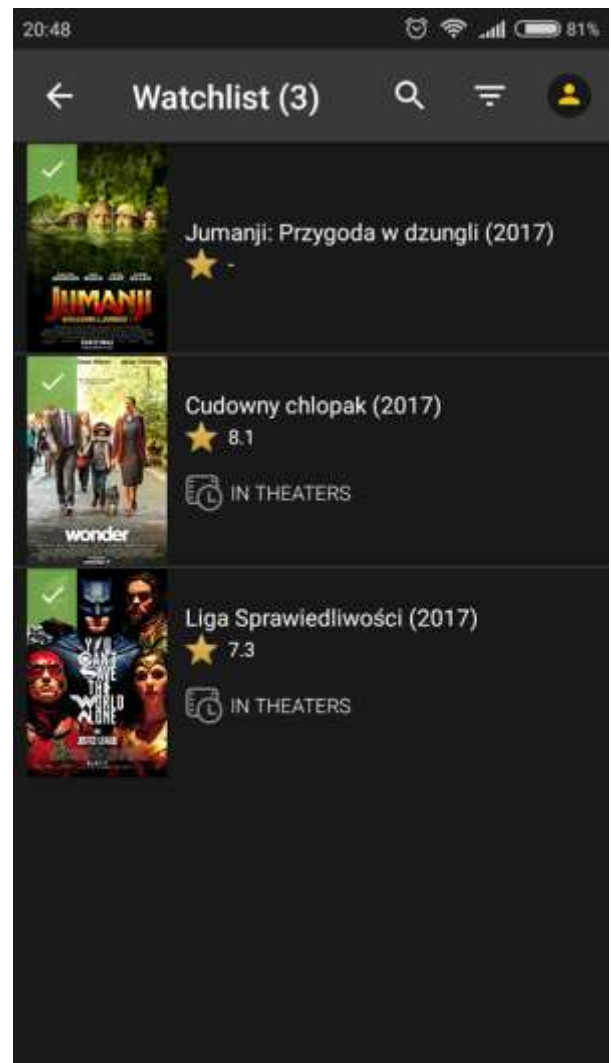
- Największa internetowa baza filmów w Polsce.
- Rozpoznawalności w sieci.
- Posiada serwis webowy oraz aplikację mobilną.

Wady:

- Filmweb stał się bardziej serwisem informacyjnym.
- Brak możliwości polecania filmów znajomym.
- Mniej znane filmy czekają latami na uzupełnienie informacji.
- Pomóc użytkownikowi w znalezieniu odpowiedniego filmu stało się drugorzędną sprawą.



Rys. 1.9 Widok startowy serwisu IMDb



Rys. 1.10 Lista filmów do obejrzenia

IMDb to największa na świecie internetowa baza filmów i ludzi z nimi związanymi. Został założony w 1990 roku. Umożliwia także swoim użytkownikom dodawanie nowych treści jak i aktualizowanie już istniejących.

Zalety:

- Największa na świecie internetowa baza filmów.
- Rozpoznawalność w sieci.
- Posiada serwis webowy oraz aplikację mobilną.

Wady:

- Brak możliwości polecania filmów znajomym.
- Niektóre informacje o filmach nie są zgodne z rzeczywistością.
- Mniej znane filmy czekają latami na uzupełnienie informacji.
- Pomóc użytkownikowi w znalezieniu odpowiedniego filmu stało się drugorzędną sprawą.

1.4 Cel pracy

W dzisiejszych czasach, gdy rynek związany z kinematografią rozwija się z dnia na dzień, potrzebne są miejsca, w których moglibyśmy mieć wszystkie informacje z nim związane zebrane razem. Ludzie nie mają czasu na przeszukiwanie wielu miejsc, dlatego potrzebują jednego rozwiązania spełniającego ich oczekiwania.

Serwis aktualizujący swoją bazę o nowo powstające filmy, a także o osoby z nimi związane, ułatwia użytkownikom bycie na bieżąco ze światem kina. Dzięki niemu w szybki sposób możemy zdobyć informację o produkcjach, w których biorą udział nasi ulubieni aktorzy bądź są tworzone przez najlepszych reżyserów. Dodatkowo chcielibyśmy mieć możliwość udostępnienia swoich preferencji najbliższym. Oprócz tego, ważna jest opcja zapoznania się z tym, co polecają nam znajomi, ponieważ znają oni nasze preferencje.

Często z braku czasu, chcemy mieć szybką informację o tym, co moglibyśmy obejrzeć, a przy tym, aby wybór był związany z naszym gustem. Do tego celu przydatny byłby algorytm polecający nam nowe pozycje na podstawie opinii na temat filmów, które mieliśmy okazję kiedyś zobaczyć. Ze względu na ulotność pamięci, ważnym aspektem jest posiadanie możliwości zapisania tytułów, które przypadły nam do gustu. Ponadto zaletą byłoby móc sprawdzanie powyższych informacji w dowolnym miejscu i sytuacji, w której się znajdujemy.

Z powyższych rozważań powstał pomysł, który polega na zaprojektowaniu i implementacji aplikacji mobilnej dla systemu Android od wersji 5.0 i wyższych, pozwalającej użytkownikowi na ocenę filmów, dodawanie pozycji do listy tytułów do obejrzenia, przeglądanie polecanych filmów jak i również bibliotek innych użytkowników. Ponad to aplikacja będzie posiadała funkcjonalność polecania filmów znajomym.

Zakres pracy obejmuje zbudowanie aplikacji mobilnej przy użyciu środowiska programistycznego Android Studio, bazy danych czasu rzeczywistego Firebase oraz dostępnych bibliotek open-source w celu przyspieszenia tworzenia aplikacji.

Praca składa się z pięciu rozdziałów. Wstęp wprowadza w dokładną tematykę pracy, opisuje aktualne rozwiązania oraz motyw wyboru technologii. Rozdział drugi zawiera wymagania funkcjonalne i нефункционалне aplikacji, przypadki użycia oraz odpowiednie do nich scenariusze[8]. Rozdział ten zawiera również opis wzorca projektowo MVC, według którego tworzona była aplikacja. W trzecim rozdziale przedstawiono interfejs graficzny gotowej aplikacji. Czwarty rozdział zawiera opis technologii oraz procesu implementacji. Ostatni rozdział to podsumowanie pracy oraz opis dalszej możliwości rozwoju aplikacji.

2 Projekt

2.1 Wymagania funkcjonalne i нефункционалне

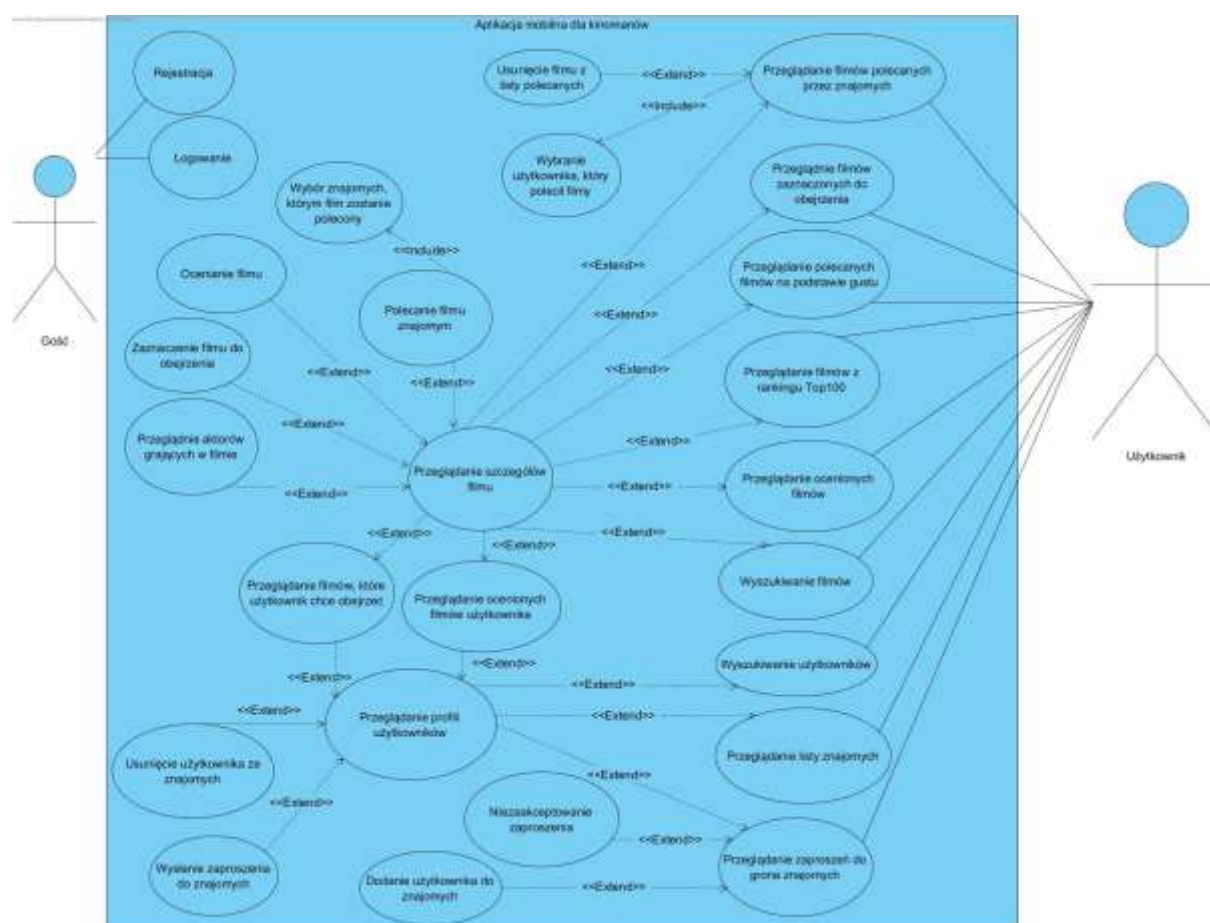
Wymagania funkcjonalne:

- Przeglądanie polecanych filmów na podstawie gustu.
- Przeglądanie filmów zaznaczonych do obejrzenia.
- Przeglądanie filmów polecanych przez znajomych.
- Przeglądanie filmów z rankingu Top100.
- Przeglądanie ocenionych filmów.
- Przeglądanie aktorów grających w danym filmie.
- Możliwość wyszukiwania pośród wszystkich filmów.
- Przeglądanie listy znajomych.
- Przeglądanie zaproszeń do grona znajomych.
- Przeglądanie profili innych użytkowników.
- Możliwość wysłania zaproszenia do znajomych przez użytkownika.
- Możliwość dodawania użytkownika do znajomych.
- Możliwość usunięcia użytkownika ze znajomych.
- Możliwość wyszukiwania profilu użytkownika.
- Możliwość otrzymania szczegółów dotyczące filmu.
- Możliwość oceny filmu.
- Możliwość zaznaczenia filmu do obejrzenia.
- Możliwość polecenia filmu znajomemu.
- Możliwość przeglądania ocen znajomych dla danego filmu.
- Wyświetlanie maksymalnie 10 losowych filmów w widoku startowym.
- Wyświetlanie maksymalnie 10 filmów pasujących do gustu w widoku startowym.
- Wyświetlanie maksymalnie 10 filmów, które chce się obejrzeć w widoku startowym.
- Wyświetlanie maksymalnie 10 filmów, które zostały obejrzone w widoku startowym.
- Możliwość utworzenia nowego konta użytkownika.
- Możliwość zalogowania użytkownika.

Wymagania нефункционалне:

- Działanie pod systemem Android 5.0 i wyższym.
- Dostęp do Internetu.
- Synchronizacja zmian w bazie danych ograniczona do 500ms.

2.2 Przypadki użycia



Rys. 2.1 Diagram przypadków użycia

Tabela 2.1 PU-1 – Przeglądanie polecanych filmów na podstawie gustu.

Lp.	PU-1
Nazwa:	Przeglądanie polecanych filmów na podstawie gustu.
Warunki początkowe:	Użytkownik znajduje się w widoku głównym aplikacji.
Opis:	Użytkownik chce uzyskać listę polecanych filmów na podstawie jego gustu.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik rozwija boczne menu. 2. Użytkownik wybiera z menu opcję „Polecane dla Ciebie”. 3. Aplikacja wyświetla odpowiednią listę filmów z bazy danych.
Ścieżka alternatywna:	3.a Wyświetlona zostaje pusta lista jeśli użytkownik nie ocenił żadnego filmu.
Warunki końcowe:	Wyświetlona zostaje lista polecanych filmów na podstawie gustu.

Tabela 2.2 PU-2 - Przeglądanie filmów zaznaczonych do obejrzenia

Lp.	PU-2
Nazwa:	Przeglądanie filmów zaznaczonych do obejrzenia.
Warunki początkowe:	Użytkownik znajduje się w widoku głównym aplikacji.
Opis:	Użytkownik chce uzyskać listę filmów, które zaznaczył do obejrzenia.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik rozwija boczne menu. 2. Użytkownik wybiera z menu opcję „Chcę obejrzeć”. 3. Aplikacja wyświetla odpowiednią listę filmów z bazy danych.
Ścieżka alternatywna:	3.a Wyświetlona zostaje pusta lista jeśli użytkownik nie zaznaczył żadnego filmu do obejrzenia.
Warunki końcowe:	Wyświetlona zostaje lista filmów, które użytkownik zaznaczył do obejrzenia.

Tabela 2.3 PU-3 - Przeglądanie filmów polecanych przez znajomych

Lp.	PU-3
Nazwa:	Przeglądanie filmów polecanych przez znajomych.
Warunki początkowe:	Użytkownik znajduje się w widoku głównym aplikacji.
Opis:	Użytkownik chce uzyskać listę polecanych filmów przez konkretnego znajomego.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik rozwija boczne menu. 2. Użytkownik wybiera z menu opcję „Polecane przez znajomych”. 3. Aplikacja wyświetla listę znajomych, którzy polecieli użytkownikowi filmy. 4. Użytkownik wybiera znajomego, od którego chce zobaczyć polecane filmy. 5. Aplikacja wyświetla odpowiednią listę filmów z bazy danych.
Ścieżka alternatywna:	3.a Wyświetlona zostaje pusta lista jeśli użytkownik nie posiada znajomych.
Warunki końcowe:	Wyświetlona zostaje lista filmów, które znajomy polecił użytkownikowi.

Tabela 2.4 PU-4 - Przeglądanie filmów z rankingu Top100

Lp.	PU-4
Nazwa:	Przeglądanie filmów z rankingu Top100.
Warunki początkowe:	Użytkownik znajduje się w widoku głównym.
Opis:	Użytkownik chce uzyskać listę najlepiej ocenianych filmów w aplikacji.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik rozwija boczne menu. 2. Użytkownik wybiera z menu opcję „Top100”. 3. Aplikacja wyświetla listę odpowiednią listę 100 filmów z bazy danych.
Ścieżka alternatywna:	
Warunki końcowe:	Wyświetlona zostaje lista 100 najlepiej ocenianych filmów w aplikacji.

Tabela 2.5 PU-5 - Przeglądanie ocenionych filmów

Lp.	PU-5
Nazwa:	Przeglądanie ocenionych filmów.
Warunki początkowe:	Użytkownik znajduje się w widoku głównym.
Opis:	Użytkownik chce uzyskać listę filmów, które już ocenił.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik rozwija boczne menu. 2. Użytkownik wybiera z menu opcję „Moje filmy”. 3. Aplikacja wyświetla odpowiednią listę filmów z bazy danych.
Ścieżka alternatywna:	3.a Wyświetlona zostaje pusta lista jeśli użytkownik nie ocenił żadnego filmu.
Warunki końcowe:	Wyświetlona zostaje lista ocenionych filmów przez użytkownika.

Tabela 2.6 PU-6 - Wyszukiwanie filmów

Lp.	PU-6
Nazwa:	Wyszukiwanie filmów.
Warunki początkowe:	Użytkownik znajduje się w widoku głównym.
Opis:	Użytkownik chce wyszukać film.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik rozwija boczne menu. 2. Użytkownik wybiera z menu opcję „Szukaj”. 3. Aplikacja wyświetla okno z edytowalnym polem tekstowym. 4. Użytkownik wpisuje wyszukiwaną frazę. 5. Aplikacja wyświetla w czasie rzeczywistym listę filmów zawierającą podaną frazę.
Ścieżka alternatywna:	5.a Jeśli aplikacja nie znajdzie filmów z daną frazą, wyświetlona zostaje pusta lista.
Warunki końcowe:	Wyświetlona zostaje lista filmów z wpisaną frazą w nazwie.

Tabela 2.7 PU-7 - Wyszukiwanie użytkowników

Lp.	PU-7
Nazwa:	Wyszukiwanie użytkowników.
Warunki początkowe:	Użytkownik znajduje się w widoku głównym.
Opis:	Użytkownik chce wyszukać użytkownika.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik rozwija boczne menu. 2. Użytkownik wybiera z menu opcję „Znajdź znajomych”. 3. Aplikacja wyświetla okno z edytowalnym polem tekstowym. 4. Użytkownik wpisuje wyszukiwaną frazę. 5. Aplikacja wyświetla w czasie rzeczywistym listę użytkowników zawierającą w nazwie podaną frazę.
Ścieżka alternatywna:	5.a Jeśli aplikacja nie znajdzie użytkowników z podaną frazą, wyświetlona zostaje pusta lista.
Warunki końcowe:	Wyświetlona zostaje lista użytkowników z wpisaną frazą w nazwie.

Tabela 2.8 PU-8 - Przeglądanie listy znajomych

Lp.	PU-8
Nazwa:	Przeglądanie listy znajomych.
Warunki początkowe:	Użytkownik znajduje się w widoku głównym.
Opis:	Użytkownik chce wyświetlić listę znajomych.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik rozwija boczne menu. 2. Użytkownik wybiera z menu opcję „Znajomi”. 3. Aplikacja wyświetla okno z listą znajomych użytkownika.
Ścieżka alternatywna:	3.a Jeśli użytkownik nie posiada znajomych, wyświetlona zostaje pusta lista.
Warunki końcowe:	Wyświetlona zostaje lista użytkowników, których użytkownik posiada w znajomych.

Tabela 2.9 PU-9 - Przeglądanie zaproszeń do grona znajomych

Lp.	PU-9
Nazwa:	Przeglądanie zaproszeń do grona znajomych.
Warunki początkowe:	Użytkownik znajduje się w widoku głównym.
Opis:	Użytkownik chce wyświetlić listę zaproszeń, które otrzymał.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik rozwija boczne menu. 2. Użytkownik wybiera z menu opcję „Zaproszenia”. 3. Aplikacja wyświetla listę zaproszeń otrzymanych przez użytkownika.
Ścieżka alternatywna:	3.a Jeśli użytkownik nie otrzymał żadnych zaproszeń, zostaje wyświetlona pusta lista.
Warunki końcowe:	Wyświetlona zostaje lista zaproszeń, wysłanych do użytkownika.

Tabela 2.10 PU-10 - Przeglądanie profili użytkowników

Lp.	PU-10
Nazwa:	Przeglądanie profili użytkowników.
Warunki początkowe:	Użytkownik znajduje się w widoku wyszukiwania użytkowników lub w widoku przeglądania listy znajomych lub przeglądania zaproszeń do grona znajomych.
Opis:	Użytkownik chce wyświetlić profil danego użytkownika.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik wybiera z listy użytkownika, którego profil chce wyświetlić. 2. Aplikacja wyświetla profil użytkownika.
Ścieżka alternatywna:	1.a Lista jest pusta.
Warunki końcowe:	Wyświetlony zostaje profil użytkownika.

Tabela 2.11 PU-11 - Dodanie użytkownika do znajomych

Lp.	PU-11
Nazwa:	Dodanie użytkownika do znajomych.
Warunki początkowe:	Użytkownik znajduje się w widoku przeglądania zaproszeń do grona znajomych.
Opis:	Użytkownik chce dodać użytkownika do grona znajomych.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik z widoku przeglądania zaproszeń wybiera przycisk potwierdzający dodanie użytkownika do grona znajomych. 2. Aplikacja dodaje użytkownika do grona znajomych.
Ścieżka alternatywna:	1.a Lista zaproszeń jest pusta.
Warunki końcowe:	Użytkownik zostaje dodany do listy znajomych.

Tabela 2.12 PU-12 - Niezaakceptowanie zaproszenia

Lp.	PU-12
Nazwa:	Niezaakceptowanie zaproszenia.
Warunki początkowe:	Użytkownik znajduje się w widoku przeglądania zaproszeń. Użytkownik posiada zaproszenia.
Opis:	Użytkownik chce odrzucić zaproszenie od innego użytkownika.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik wybiera przycisk odrzucenia zaproszenia. 2. Aplikacja usuwa z listy zaproszenie od użytkownika.
Ścieżka alternatywna:	1.a Użytkownik wybiera przycisk akceptacji zaproszenia.
Warunki końcowe:	Zaproszenie zostaje odrzucone.

Tabela 2.13 PU-13 - Usunięcie użytkownika ze znajomych

Lp.	PU-13
Nazwa:	Usunięcie użytkownika ze znajomych.
Warunki początkowe:	Użytkownik znajduje się w widoku profilu użytkownika. Użytkownik posiada danego użytkownika w znajomych.
Opis:	Użytkownik chce usunąć użytkownika ze znajomych.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik wybiera przycisk usunięcia ze znajomych. 2. Aplikacja wyświetla okienko z potwierdzeniem usunięcia ze znajomych. 3. Użytkownik potwierdza usunięcie ze znajomych. 4. Aplikacja usuwa użytkownika z listy znajomych.
Ścieżka alternatywna:	<p>1.a Jeśli użytkownik nie posiada danego użytkownika w znajomych, przycisk jest zablokowany.</p> <p>3.a Użytkownik nie potwierdza usunięcia z listy znajomych.</p>
Warunki końcowe:	Użytkownik zostaje usunięty z listy znajomych.

Tabela 2.14 PU-14 - Przeglądanie ocenionych filmów użytkownika

Lp.	PU-14
Nazwa:	Przeglądanie ocenionych filmów użytkownika.
Warunki początkowe:	Użytkownik znajduje się w widoku profilu użytkownika.
Opis:	Użytkownik chce wyświetlić listę ocenionych filmów danego użytkownika.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję „Ocenione” 2. Aplikacja wyświetla listę ocenionych filmów użytkownika
Ścieżka alternatywna:	2.a Jeśli dany użytkownik nie posiada żadnych ocenionych filmów, wyświetlona zostaje pusta lista.
Warunki końcowe:	Wyświetlona zostaje lista ocenionych filmów danego użytkownika.

Tabela 2.15 PU-15 - Przeglądanie filmów, które użytkownik chce obejrzeć

Lp.	PU-15
Nazwa:	Przeglądanie filmów, które użytkownik chce obejrzeć.
Warunki początkowe:	Użytkownik znajduje się w widoku profilu użytkownika.
Opis:	Użytkownik chce wyświetlić listę zaznaczonych do obejrzenia filmów danego użytkownika.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik wybiera opcję „Chce zobaczyć” 2. Aplikacja wyświetla listę zaznaczonych do obejrzenia filmów danego użytkownika.
Ścieżka alternatywna:	2.a Jeśli dany użytkownik nie posiada żadnych filmów zaznaczonych do obejrzenia, wyświetlona zostaje pusta lista.
Warunki końcowe:	Wyświetlona zostaje lista zaznaczonych do obejrzenia filmów danego użytkownika.

Tabela 2.16 PU-16 - Wysłanie zaproszenia do znajomych

Lp.	PU-16
Nazwa:	Wysyłanie zaproszenia do znajomych.
Warunki początkowe:	Użytkownik znajduje się w widoku profilu użytkownika. Użytkownik nie posiada danego użytkownika w znajomych.
Opis:	Użytkownik chce wysłać zaproszenie do grona znajomych innemu użytkownikowi.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik wybiera przycisk dodania do znajomych użytkownika. 2. Aplikacja wyświetla okienko z potwierdzeniem wysłania zaproszenia. 3. Użytkownik potwierdza wysłanie zaproszenia. 4. Aplikacja wysyła zaproszenie do użytkownika.
Ścieżka alternatywna:	1.a Jeśli użytkownik posiada już danego użytkownika w znajomych, przycisk jest zablokowany.
Warunki końcowe:	Zaproszenie do znajomych zostaje wysłane.

Tabela 2.17 PU-17 - Przeglądanie szczegółów filmu

Lp.	PU-17
Nazwa:	Przeglądanie szczegółów filmu.
Warunki początkowe:	Użytkownik wykonał jeden z przypadków użycia: PU-1, PU-2, PU-3, PU-4, PU-5, PU-6, PU-14, PU- 15. Lista w podanych przypadkach użycia nie jest pusta.
Opis:	Użytkownik chce wyświetlić szczegóły na temat filmu.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik wybiera film z listy filmów. 2. Aplikacja wyświetla widok z szczegółami na temat filmu.
Ścieżka alternatywna:	1.a Lista jest pusta.
Warunki końcowe:	Wyświetlona zostają szczegóły na temat filmu.

Tabela 2.18 PU-18 - Ocenianie filmu

Lp.	PU-18
Nazwa:	Ocenianie filmu.
Warunki początkowe:	Użytkownik znajduje się w widoku szczegółów filmu.
Opis:	Użytkownik chce ocenić dany film.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik klika na odpowiednią gwiazdkę na pasku oceny. 2. Aplikacja zapisuje daną ocenę. 3. Aplikacja zmienia liczbę ocen dla filmu 4. Aplikacja wylicza nową średnią ocen dla filmu.
Ścieżka alternatywna:	
Warunki końcowe:	Ocena dla filmu zostaje zapisana.

Tabela 2.19 PU-19 - Zaznaczenie filmu do obejrzenia

Lp.	PU-19
Nazwa:	Zaznaczenie filmu do obejrzenia.
Warunki początkowe:	Użytkownik znajduje się w widoku szczegółów filmu. Dany film nie jest już zaznaczony jako do obejrzenia.
Opis:	Użytkownik chce oznaczyć film jako do obejrzenia.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik klika na ikonkę „oka”. 2. Aplikacja zapisuje film jako zaznaczony do obejrzenia. 3. Aplikacja zmienia kolor ikonki na czerwony.
Ścieżka alternatywna:	2a. Jeśli film był już zaznaczony jako do obejrzenia, zostaje usunięty z tej listy.
Warunki końcowe:	Film zostaje oznaczony do obejrzenia.

Tabela 2.20 PU-20 - Polecanie filmu znajomym

Lp.	PU-20
Nazwa:	Polecanie filmu znajomym.
Warunki początkowe:	Użytkownik znajduje się w widoku szczegółów filmu. Użytkownik posiada znajomych.
Opis:	Użytkownik chce polecić film konkretnym znajomym.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik klika na przycisk „Poleć znajomym”. 2. Aplikacja wyświetla okienko z listą znajomych. 3. Użytkownik wybiera, którym znajomym polecić dany film.(PU-21) 4. Aplikacja kolejno poleca filmy znajomym.
Ścieżka alternatywna:	<ol style="list-style-type: none"> 2.a Jeśli użytkownik nie posiada znajomych, wyświetli się okno z pustą listą. 2.b Użytkownik zamyka okno. 3.a Użytkownik zamyka okno.
Warunki końcowe:	Filmy zostają polecane znajomym.

Tabela 2.21 PU-21 - Wybór znajomych, którym film zostanie polecony.

Lp.	PU-21
Nazwa:	Wybór znajomych, którym film zostanie polecony.
Warunki początkowe:	Użytkownik znajduje się w okienku wyboru znajomych.
Opis:	Użytkownik chce polecić film konkretnym znajomym.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Aplikacja wyświetla okienko z listą znajomych. 2. Użytkownik wybiera, którym znajomym polecić dany film. 3. Aplikacja kolejno poleca filmy znajomym.
Ścieżka alternatywna:	<ol style="list-style-type: none"> 2.a Jeśli użytkownik nie posiada znajomych, wyświetli się okno z pustą listą. 2.b Użytkownik zamyka okno.
Warunki końcowe:	Filmy zostają polecane znajomym.

Tabela 2.22 PU-22 - Przeglądanie aktorów grających filmie

Lp.	PU-22
Nazwa:	Przeglądanie aktorów grających filmie.
Warunki początkowe:	Użytkownik znajduje się w widoku szczegółów filmu.
Opis:	Użytkownik chce zobaczyć obsadę filmu.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik klika przycisk „Pełna obsada”. 2. Aplikacja wyświetla widok z listą pełnej obsady filmu.
Ścieżka alternatywna:	
Warunki końcowe:	Wyświetlona zostaje lista pełnej obsady filmu.

Tabela 2.23 PU-23 – Usunięcie filmu z listy polecanych

Lp.	PU-23
Nazwa:	Usunięcie filmu z listy polecanych.
Warunki początkowe:	Użytkownik znajduje się w widoku polecanych filmów przez użytkownika. Lista nie jest pusta.
Opis:	Użytkownik chce usunąć dany film z listy polecanych od znajomego.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik klika ikonkę krzyżyka obok propozycji filmu, którą chce usunąć. 2. Aplikacja usuwa daną pozycję filmową z listy polecanych przez znajomego.
Ścieżka alternatywna:	1.a Lista jest pusta.
Warunki końcowe:	Film zostaje usunięty z listy polecanych przez znajomego.

Tabela 2.24 PU-24 – Wybranie użytkownika, który polecił film

Lp.	PU-24
Nazwa:	Wybranie użytkownika, który polecił film.
Warunki początkowe:	Użytkownik wybrał z menu opcję „Polecane przez znajomych”
Opis:	Użytkownik musi wybrać znajomego, od którego chce zobaczyć polecane filmy.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik wybiera z listy znajomego, od którego chce zobaczyć polecane filmy. 2. Aplikacja kontynuuje PU-3.
Ścieżka alternatywna:	1.a Użytkownik nie wybiera żadnego znajomego. 1.b Lista jest pusta.
Warunki końcowe:	PU-3 jest kontynuowany.

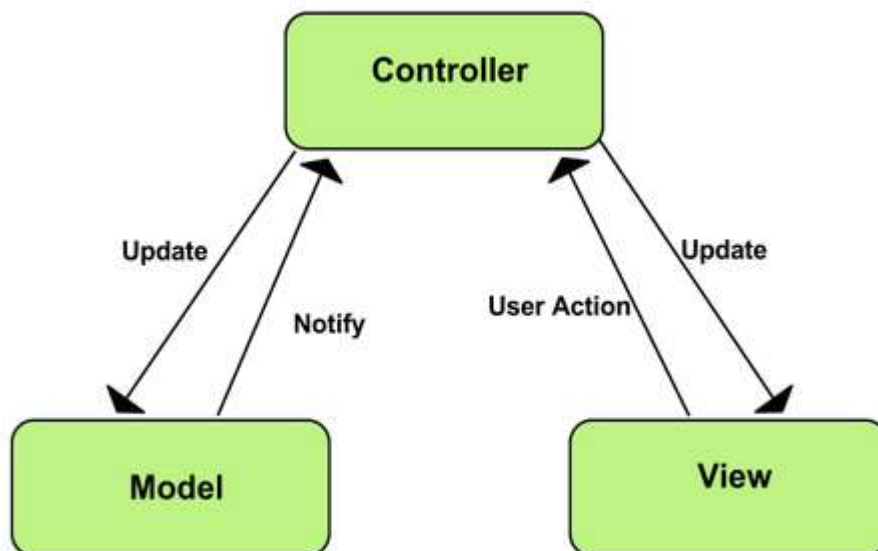
Tabela 2.25 PU-25 - Rejestracja

Lp.	PU-25
Nazwa:	Rejestracja.
Warunki początkowe:	Użytkownik nie jest zalogowany.
Opis:	Użytkownik chce się zarejestrować w aplikacji.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik klika przycisk „REJESTRUJ”. 2. Aplikacja wyświetla widok z danymi do uzupełnienia. 3. Użytkownik wypełnia adres e-mail, hasło dwukrotnie oraz płeć. 4. Użytkownik klika przycisk „REJESTRUJ”. 5. Aplikacja poprawnie rejestruje i loguje użytkownika.
Ścieżka alternatywna:	5.a Użytkownik proszony jest o wpisanie poprawnie danych.
Warunki końcowe:	Użytkownik zostaje zarejestrowany i zalogowany.

Tabela 2.26 PU-26 - Logowanie

Lp.	PU-26
Nazwa:	Logowanie.
Warunki początkowe:	Użytkownik nie jest zalogowany. Użytkownik jest zarejestrowany.
Opis:	Użytkownik chce się zalogować do aplikacji.
Ścieżka główna:	<ol style="list-style-type: none"> 1. Użytkownik wpisuje adres e-mail jako login oraz swoje hasło. 2. Użytkownik klika przycisk „ZALOGUJ”. 3. Aplikacja loguje użytkownika i wyświetla ekran główny.
Ścieżka alternatywna:	3.a Aplikacja wyświetla informację o nieudanej próbie logowania.
Warunki końcowe:	Użytkownik zostaje zalogowany w aplikacji.

2.3 Wzorzec architektoniczny MVC.



Rys. 2.2 Wzorzec Model-View-Controller

Wzorcem architektonicznym użytym przy tworzeniu pracy był MVC, czyli Model-View-Controller. Dzięki niemu można w łatwy sposób osobno zarządzać modelem danych, widokiem oraz kontrolerem.[9]

W modelu danych zawarta została logika związana z bazą danych. Posiada ona klasy, odpowiedzialne za przyjmowanie obiektów z bazy danych Firebase. Stworzone obiekty są skonstruowane w taki sposób, aby mieć bardzo łatwy dostęp do potrzebnych danych.

Widok natomiast to głównie pliki xml oraz klasy JAVA, które komunikują się między aplikacją, a plikami xml.

Kontroler zawiera całą logikę aplikacji, odpowiadającą na zachowanie użytkownika. W aplikacji często była konieczność wyświetlania danych poprzez listy. W tym celu wykorzystano tutaj klasy, będące adapterami, które na podstawie dostarczanych danych dynamicznie zmieniały swój wygląd i zawartość.

Model MVC jest jednym z modeli wykorzystywanych przy tworzeniu aplikacji. Jego celem jest oddzielenie logiki biznesowej, widoków graficznych, oraz warstwy obsługującej reakcje użytkownika. W znaczny sposób ułatwia implementację aplikacji i ułatwia późniejsze zmiany w projekcie.

3 Interfejs użytkownika

3.1 Widok rejestracji i logowania



Rys. 3.1 Widok ekranu logowania



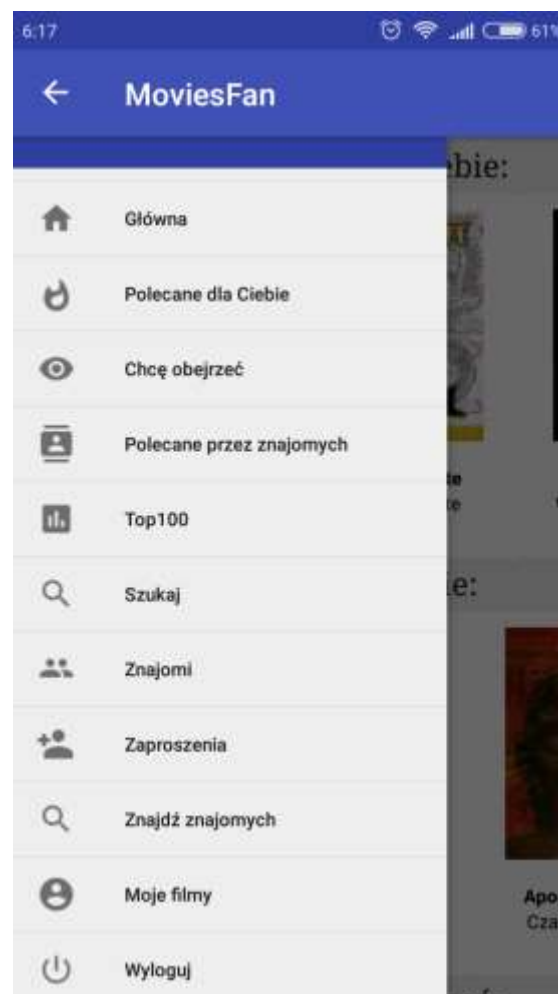
Rys. 3.2 Widok ekranu rejestracji

Widok ekranu logowania jest pierwszym widokiem z jakim spotyka się użytkownik po zainstalowaniu aplikacji. Jeśli posiada konto, może się zalogować i korzystać z dalszych funkcjonalności, w przeciwnym wypadku musi najpierw założyć konto w widoku rejestracji lub zalogować się za pośrednictwem konta z serwisu społecznościowego Facebook. Przy rejestracji użytkownik jest zobowiązany podać adres e-mail, który będzie służył jako login identyfikujący oraz hasła. Dodatkowo musi zaznaczyć jakiej jest płci.

3.2 Widok ekranu głównego



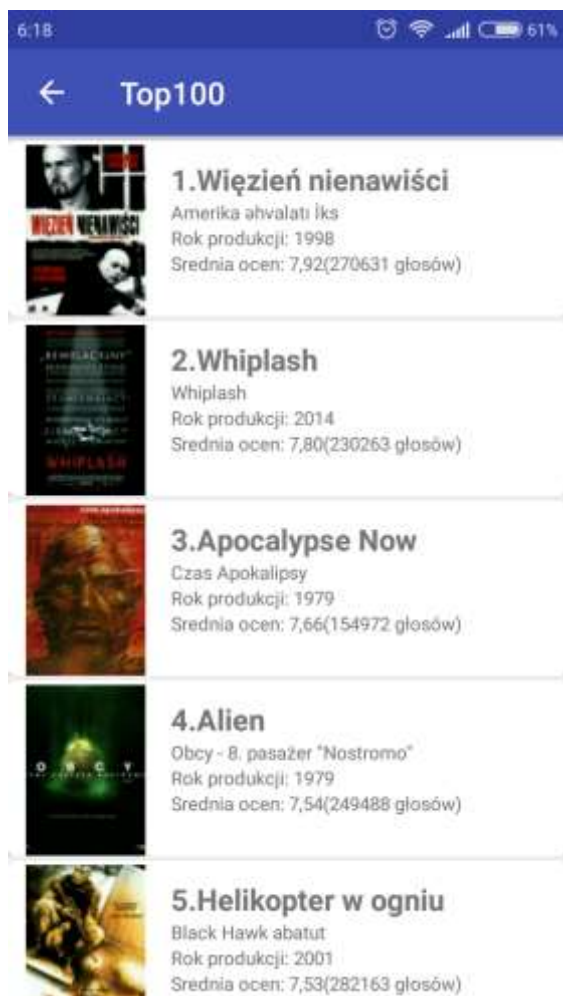
Rys. 3.3 Widok ekranu startowego aplikacji



Rys. 3.4 Menu nawigacyjne

Uruchamiając aplikację, zalogowany użytkownik w widoku głównym ma do dyspozycji cztery listy, która każda zawiera maksymalnie dziesięć pozycji filmowych. Pierwsza z nich to filmy całkowicie losowo wybrane z bazy danych. Następna lista przedstawia filmy polecane na podstawie gustu użytkownika, kolejna te, które zazaczył do obejrzenia, a ostatnia pozycje już przez niego ocenione. Te cztery różne od siebie listy mają na celu szybką pomoc w wyborze filmu do obejrzenia. Ponad to użytkownik może rozwinąć menu nawigacyjne i przejść do innych interesujących go funkcjonalności aplikacji.

3.3 Widok najlepiej ocenianych filmów i polecanych dla użytkownika



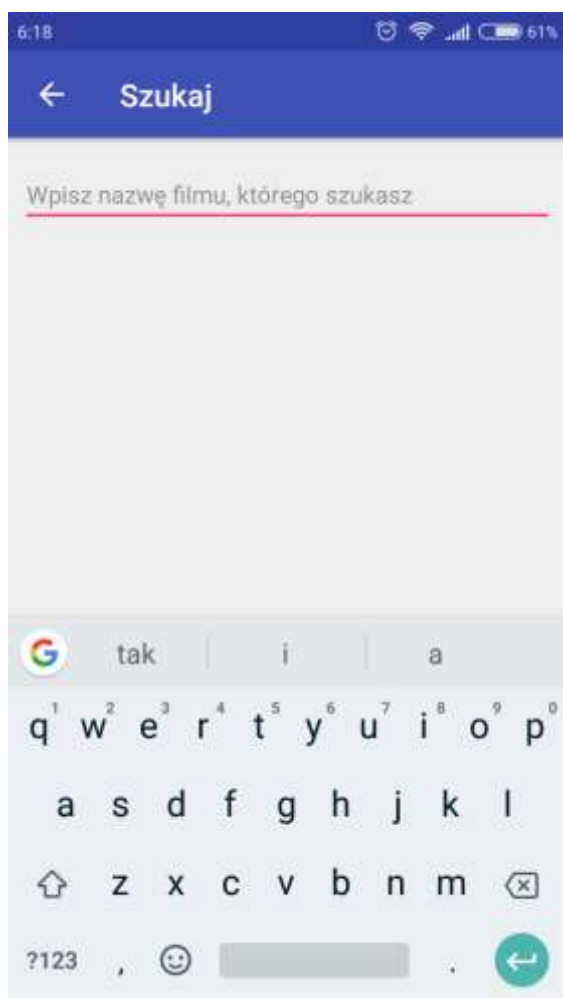
Rys. 3.5 Widok listy rankingowej filmów



Rys. 3.6 Widok listy filmów polecanych

Widok przedstawia sto pozycji filmowych, które otrzymały najwyższe oceny w całej aplikacji. Użytkownik w ten sposób może sprawdzić co podoba się społeczności i znaleźć film odpowiedni dla siebie. Kolejnym widokiem jest lista polecanych filmów na podstawie gustu użytkownika, które są wyświetlane na podstawie gatunków filmowych, które użytkownik najczęściej ogląda.

3.4 Widok wyszukiwania pozycji filmowych



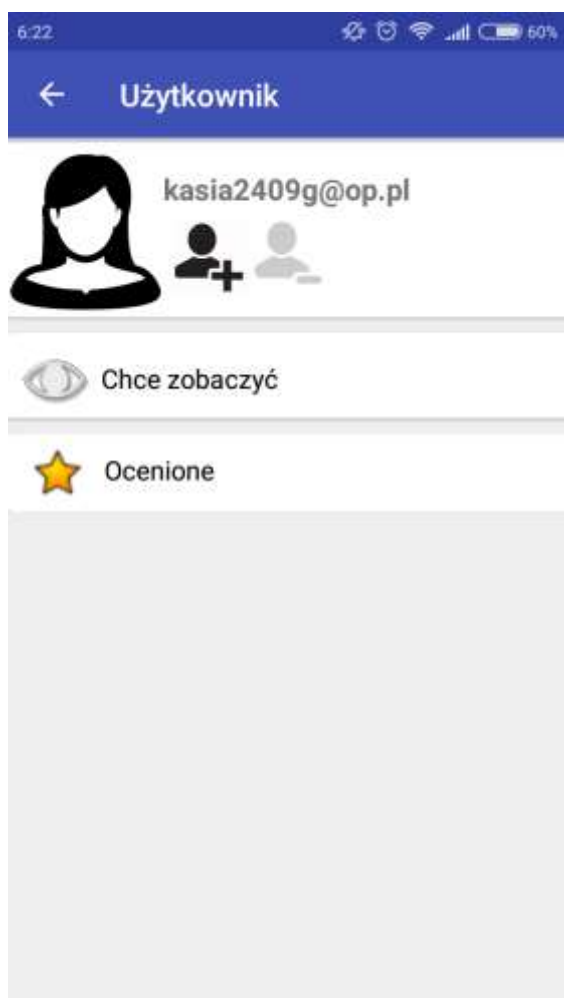
Rys. 3.7 Widok przed wyszukiwaniem



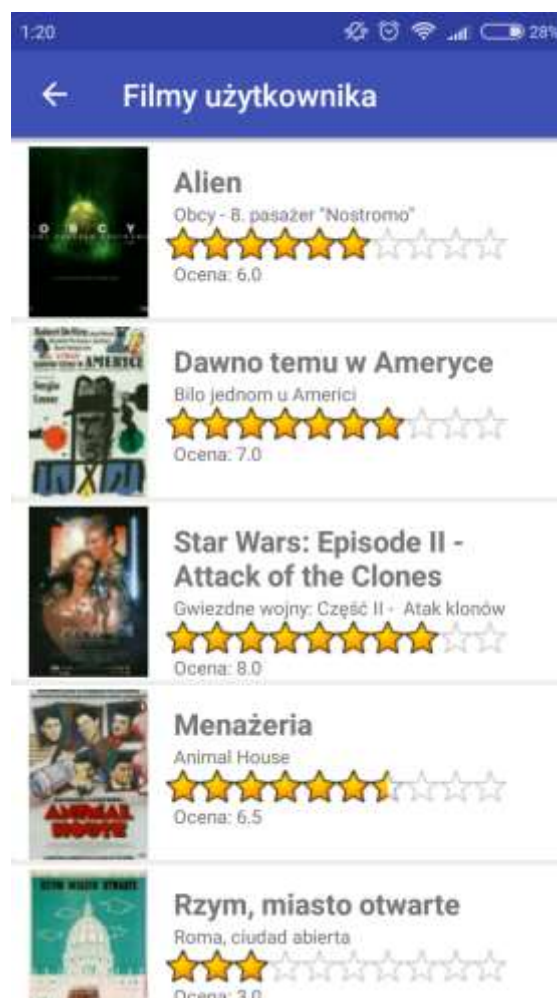
Rys. 3.8 Widok po wyszukiwaniu

Widok wyszukiwania pozycji filmowych służy użytkownikowi do znalezienia odpowiedniej pozycji na podstawie tytułu. Użytkownik nie musi pamiętać całej nazwy filmu, aby go znaleźć, ponieważ wyszukiwarka na bieżąco wyświetla wyniki odpowiadające aktualnie wpisanej frazie.

3.5 Widok profilu użytkownika i ocenionych filmów



Rys. 3.9 Widok profilu użytkownika



Rys. 3.10 Widok ocenionych filmów

Widok profilu użytkownika, pozwala korzystającemu z aplikacji na dodawanie i usuwanie innych użytkowników do grona znajomych. Ponadto pozwala przeglądać ocenione przez nich filmy, oraz te zaznaczone jako pozycje, które chcą zobaczyć.

3.6 Widok filmu



Rys. 3.11 Widok szczegółów filmu 1



Rys. 3.12 Widok szczegółów filmu 2

Widok filmu pozwala użytkownikowi dowiedzieć się szczegółowych informacji o filmie. Zawiera on krótki opis fabuły, gatunek, informację **kraj produkcji** oraz obsadę aktorską. Ponadto w tym widoku użytkownik może oznaczyć dany tytuł jako, ten który chciałby obejrzeć w przyszłości. Kolejną opcją jest możliwość sprawdzenia jakie oceny zostały przyznane przez znajomych użytkownika. Dodatkowo jest możliwość polecenia tytułu znajomym.

4 Implementacja

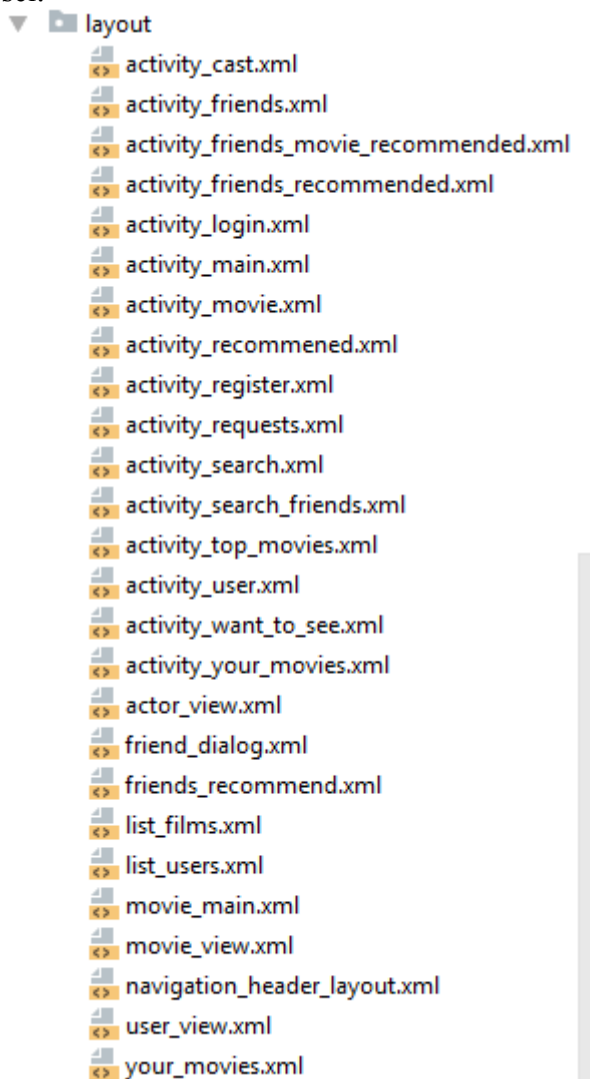
4.1 Środowisko i narzędzia programistyczne

Środowisko programistyczne

Do pracy nad projektem wykorzystane zostało środowisko programistyczne Google Android Studio początkowo w wersji 2.3.3, zaktualizowane w trakcie pracy do wersji 3.0. [10][11]Diagram przypadków użycia został utworzony przy pomocy programu Visual Paradigm Standard 13.2 z użyciem licencji studenckiej. W trakcie tworzenia aplikacji wykorzystywano również system kontroli wersji Git[12], wykorzystując do tego prywatne repozytorium w serwisie GitHub.0

Języki programowania

Projekt został zaimplementowany w języku programowania Java. Interfejs graficzny aplikacji został napisany przy użyciu języka znaczników XML[13], który służył do opisanie widoku wszystkich aktywności.



Rys. 4.1 Lista widoków zaimplementowana w aplikacji

Kod był pisany w taki sposób, aby była zgodność z założeniami „czystego kodu”[14], które zostały opisane w książce autorstwa Roberta C. Martina „Clean Code. Przy tworzeniu aplikacji przydatna była dokumentacja Google’a, opisującej proces tworzenia aplikacji dla systemu Android[7].

Baza danych

W celu utworzenia bazy danych użyto rozwiązania udostępnianego przez firmę Google, a mianowicie bazy danych czasu rzeczywistego – Firebase. Jest to baza danych NoSql, przechowująca dane w formacie klucz i wartość.[17]

1103

```
..... avgRate: "7.7443"
..... countries: "Wielka Brytania"
..... duration: "104"
..... genre: "Dramat, Psychologiczn
..... id: "1103"
..... photoUrl: "http://1.fwcdn.pl/po/11/03/1103/7430538.4.
..... plot: "Po wyjeździe współlokatorów Carol zaczyna popac
..... polishTitle: "Avsporet
..... rateCount: "22503"
..... title: "Wstręt"
..... weightAvg: "5.361658397686361
..... year: "1965"
```

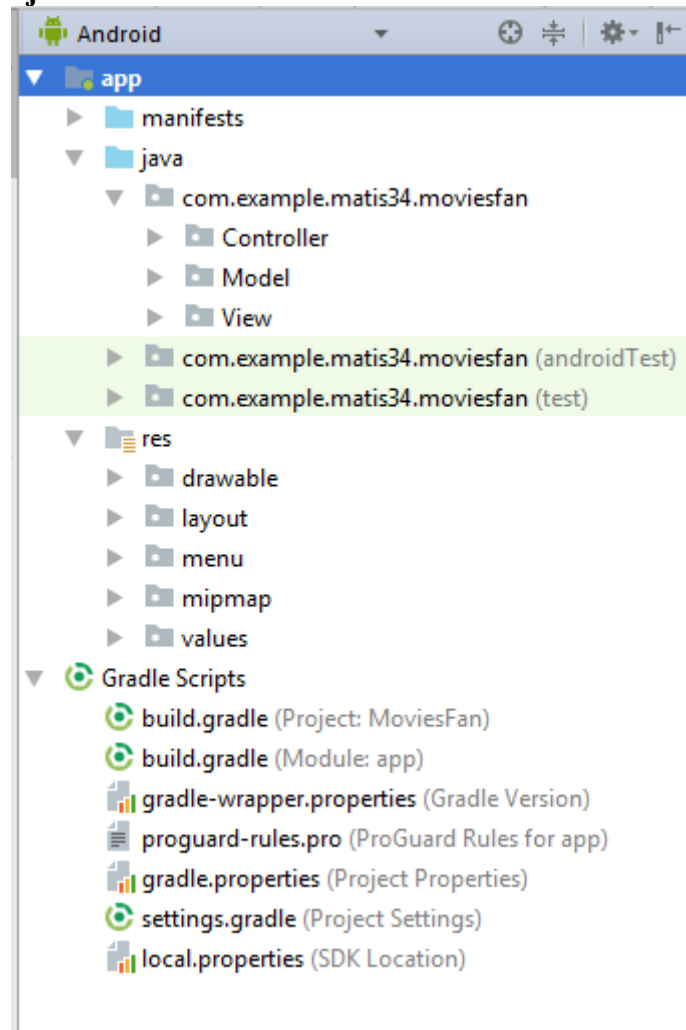
Rys. 4.2 Przykładowy sposób przechowywania danych filmu w bazie Firebase

Dodatkowe biblioteki

Glide – jest to szybka i wydajna biblioteka do wczytywania obrazów dla Androida skupiona na płynnym przewijaniu. [15]

Facebook Android SDK – pakiet bibliotek pozwalających połączenie aplikacji Android z kontem społecznościowym na portalu Facebook.[16]

4.2 Składowe projektu



Rys. 4.3 Elementy składowe projektu

Na powyższym rysunku przedstawiony został wycinek elementów, z których składał się projekt. Najważniejsze elementy to:

- Folder Controller – zawiera w sobie klasy odpowiadające na zachowanie użytkownika.
- Folder Model – zawiera w sobie klasy odzwierciedlające obiekty w bazie danych.
- Folder View – zawiera w sobie klasy aktywności.
- Folder layout – zawiera pliki XML definiujące widok aktywności.
- Folder drawable – zawiera w sobie grafiki użyte w wyglądzie aplikacji.

4.3 Problemy implementacyjne

Podczas pracy nad implementacją aplikacji doszło do problemów implementacyjnych, które zostaną opisane poniżej wraz z opisem ich rozwiązania.

Tabela 4.1 PI-01 - Czas wyszukiwania aktorów do filmu.

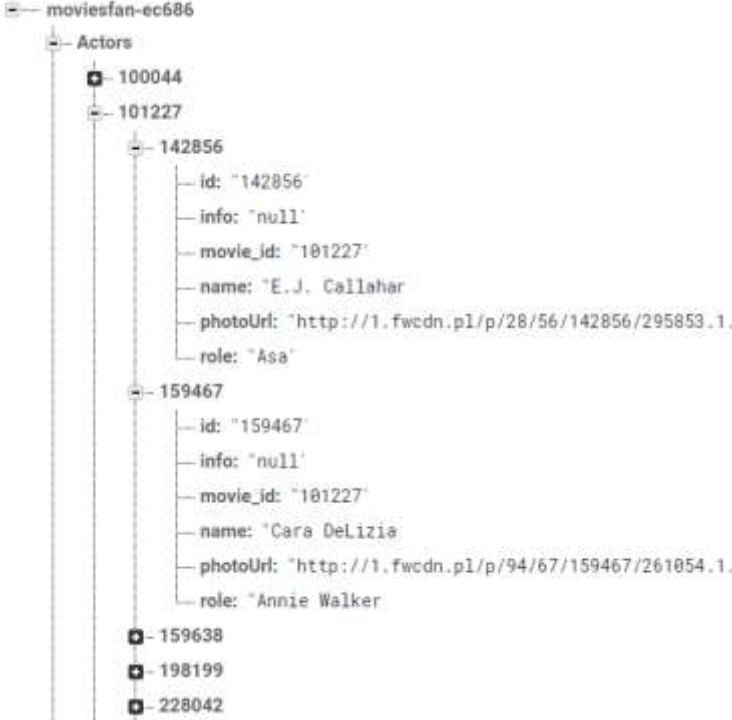
Lp.	PI-01
Nazwa	Czas wyszukiwania aktorów do filmu.
Opis problemu	Problem dotyczył czasu oczekiwania na otrzymanie informacji o aktorach dla danego filmu. Głównym problemem była tutaj początkowa struktura bazy danych, a wraz z tym czas potrzebny na osiągnięcie odpowiedniego efektu.
Rozwiązanie	<p>Rozwiązanie polegało na zmianie struktury bazy na taką, w której aktorzy posiadali oddzielne odgałęzienie, a w nim dopiero filmy wraz aktorami. Jest to nadmiarowość danych, która jest stosowana w tego typu bazach.</p>  <pre>graph TD moviesfan[moviesfan-ec686] --> Actors Actors --> 100044 Actors --> 101227 Actors --> 142856 Actors --> 159467 Actors --> 159638 Actors --> 198199 Actors --> 228042 142856 --> id142856[id: "142856"] 142856 --> info142856[info: "null"] 142856 --> movie_id142856[movie_id: "101227"] 142856 --> name142856[name: "E.J. Callahar"] 142856 --> photoUrl142856[photoUrl: "http://1.fwcdn.pl/p/28/56/142856/295853.1."] 142856 --> role142856[role: "Asa"] 159467 --> id159467[id: "159467"] 159467 --> info159467[info: "null"] 159467 --> movie_id159467[movie_id: "101227"] 159467 --> name159467[name: "Cara DeLizia"] 159467 --> photoUrl159467[photoUrl: "http://1.fwcdn.pl/p/94/67/159467/261054.1."] 159467 --> role159467[role: "Annie Walker"]</pre> <p>Rys. 4.4 Struktura przetrzymywania informacji o aktorach w bazie danych.</p> <p>Zastosowanie tego rozwiązania w stu procentach rozwiązało problem z czasem oczekiwania.</p>

Tabela 4.2 PI-02 - Wyświetlanie aktorów i opisu filmu w jednej aktywności.

Lp.	PI-02
Nazwa	Wyświetlanie aktorów i opisu filmu w jednej aktywności.
Opis problemu	Problem dotyczył wyświetlania przy pomocy adaptera dwóch różnych obiektów.
Rozwiązanie	<p>Rozwiązanie polegało na utworzeniu dwóch różnych klas rozszerzanych przez „RecyclerView.ViewHolder”. Dane przesyłane do adaptera były przechowywane w ArrayList przechowującej elementy typu object, dzięki czemu można było w niej umieścić zarówno filmy jak aktorów.</p> <pre> private final int MOVIE = 0; private final int ACTOR = 1; @Override public int getItemViewType(int position) { if (list_items.get(position) instanceof ActorObject) { return ACTOR; } else if (list_items.get(position) instanceof Movie) { return MOVIE; } return -1; } @Override public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) { LayoutInflater inflater = LayoutInflater.from(parent.getContext()); RecyclerView.ViewHolder viewHolder = null; switch (viewType) { case MOVIE: View view1 = inflater.inflate(R.layout.movie_view, parent, false); viewHolder = new ViewHolder1(view1); break; case ACTOR: View view2 = inflater.inflate(R.layout.actor_view, parent, false); viewHolder = new ViewHolder2(view2); break; } return viewHolder; } </pre> <p>Powyższe rozwiązanie na podstawie instancji klasy przypisywało odpowiednią wartość typu int do zmiennej viewType, dzięki której był wybierany odpowiedni layout dla obiektu filmu i aktorów.</p>

Tabela 4.3 PI-03 - Wyświetlanie listy znajomych, polecających pozycje filmowe.

Lp.	PI-03
Nazwa	Wyświetlanie listy znajomych, polecających pozycje filmowe.
Opis problemu	Problem dotyczył dublowania listy znajomych, przy powrocie do widoku znajomych po usunięciu jakiejkolwiek pozycji polecanej.
Rozwiązanie	Rozwiązanie polegało na zmianie miejsca inicjowania zmiennej przechowującej informacje o znajomych. Wynikało to z asynchroniczności zapytań wysyłanych do bazy danych i kolejności wykonywania poszczególnych linii kodu w listenerze. Dzięki debugowaniu szybko rozpoznano przyczynę problemu i ją wyeliminowano.

Tabela 4.4 PI-04 – Wystawianie oceny.

Lp.	PI-04
Nazwa	Wystawianie oceny.
Opis problemu	Problem dotyczył bardzo długiego oczekiwania na wystawienie oceny dla danego filmu lub nawet zaprzestawania działania aplikacji.
Rozwiązanie	Rozwiązanie polegało na zmianie rodzaju listenera wykorzystanego do pobierania rekordów z bazy danych z „addValueEventListener” na „addListenerForSingleValueEvent”. Dzięki temu po wystawieniu wybranej oceny dla filmu nie nastąpiło już niepotrzebne przeładowanie wcześniej już załadowanych danych.

Tabela 4.5 PI-05 - Widok startowy.

Lp.	PI-05
Nazwa	Widok startowy.
Opis problemu	Problem dotyczył koncepcji przedstawienia widoku startowego w taki sposób, aby użytkownik mógł od razu znaleźć film dla siebie.
Rozwiązanie	Rozwiązanie polegało na umieszczeniu w layoucie czterech różnych RecyclerView, aby każdy z nich zawierał różne filmy: polecane na podstawie gustu, te które chce obejrzeć, ocenione pozycje jak i wybrane losowo dla użytkownika.

5 Podsumowanie

Celem pracy inżynierskiej było zbudowanie aplikacji mobilnej na system Android o tematyce filmowej. Miała ona na celu wspomóc użytkownika w wyborze pozycji kinematograficznej, a co za tym idzie oszczędzić jego czas dzięki zgromadzeniu mogących interesować go treści w jednym miejscu. Proces tworzenia aplikacji dla użytkownika składał się z następujących elementów:

- projektu aplikacji,
- implementacji aplikacji,
- testowaniu aplikacji.

Wszystkie przypadki użycia i przypisane im scenariusze, wliczając to ścieżki alternatywne zostały brane pod uwagę w trakcie prac implementacyjnych. Każdy z przypadków ma swoje w pełni działające odzwierciedlenie w aplikacji. Zostały wyeliminowane wszystkie znane błędy w działaniu. Tworząc interfejs graficzny, brano pod uwagę sugestie innych osób korzystających z tego typu aplikacji, czego efektem jest przyjemny i prosty w użyciu dla użytkownika interfejs. W porównaniu z innymi dostępnymi rozwiązaniami:

- wprowadzono możliwość samodzielnego polecenia filmu znajomemu,
- już pierwsze uruchomienie aplikacji pozwala użytkownikowi znalezienie filmu dla siebie

W efekcie końcowym uzyskano w pełni działający produkt, spełniającą wszystkie założenia projektowe.

W przyszłości aplikacja ta mogłaby zostać udoskonalona o następujące funkcjonalności:

- pisanie recenzji filmów przez użytkowników,
- dodawanie nowych filmów do bazy przez użytkowników,
- możliwość tworzenia prywatnych kolekcji filmów,
- dodanie możliwości rejestracji poprzez pozostałe serwisy społecznościowe,
- repertuar kin w pobliżu.

Spis ilustracji

Rys. 1.1 Liczba użytkowników smart fonów a komputerów stacjonarnych latach 2007-2015(w milionach)[1]	4
Rys. 1.2 Procentowy udział systemów na urządzeniach mobilnych latach 2009-2016[2]	5
Rys. 1.3 Liczba dostępnych aplikacji w Sklepie Google Play od grudnia 2009 do września 2017[4]	6
Rys. 1.4 Łączna liczba aplikacji pobranych z Google Play od maja 2016 (w miliardach)[5] ...	7
Rys. 1.5 Dystrybucja wersji Androida właścicieli smartfonów – wrzesień 2017r.[6]	7
Rys. 1.6 Architektura systemu Android[7]	9
Rys. 1.7 Widok startowy serwisu Filmweb.....	11
Rys. 1.8 Lista filmów do obejrzenia	11
Rys. 1.9 Widok startowy serwisu IMDb	12
Rys. 1.10 Lista filmów do obejrzenia	12
Rys. 2.1 Diagram przypadków użycia	15
Rys. 2.2 Wzorzec Model-View-Controller	26
Rys. 3.1 Widok ekranu logowania	27
Rys. 3.2 Widok ekranu rejestracji	27
Rys. 3.3 Widok ekranu startowego aplikacji.....	28
Rys. 3.4 Menu nawigacyjne	28
Rys. 3.5 Widok listy rankingowej filmów	29
Rys. 3.6 Widok listy filmów polecanych	29
Rys. 3.7 Widok przed wyszukiwaniem.....	30
Rys. 3.8 Widok po wyszukiwaniu.....	30
Rys. 3.9 Widok profilu użytkownika	31
Rys. 3.10 Widok ocenionych filmów	31
Rys. 3.11 Widok szczegółów filmu 1	32
Rys. 3.12 Widok szczegółów filmu 2	32
Rys. 4.1 Lista widoków zaimplementowana w aplikacji	33
Rys. 4.2 Przykładowy sposób przetrzymywania danych filmu w bazie Firebase.....	34
Rys. 4.3 Elementy składowe projektu	35
Rys. 4.4 Struktura przetrzymywania informacji o aktorach w bazie danych.....	36

Spis tabel

Tabela 2.1 PU-1 – Przeglądanie polecanych filmów na podstawie gustu.	15
Tabela 2.2 PU-2 - Przeglądanie filmów zaznaczonych do obejrzenia	16
Tabela 2.3 PU-3 - Przeglądanie filmów polecanych przez znajomych.....	16
Tabela 2.4 PU-4 - Przeglądanie filmów z rankingu Top100.....	17
Tabela 2.5 PU-5 - Przeglądanie ocenionych filmów.....	17
Tabela 2.6 PU-6 - Wyszukiwanie filmów	17
Tabela 2.7 PU-7 - Wyszukiwanie użytkowników	18
Tabela 2.8 PU-8 - Przeglądanie listy znajomych	18
Tabela 2.9 PU-9 - Przeglądanie zaproszeń do grona znajomych	18
Tabela 2.10 PU-10 - Przeglądanie profili użytkowników	19
Tabela 2.11 PU-11 - Dodanie użytkownika do znajomych	19
Tabela 2.12 PU-12 - Niezaakceptowanie zaproszenia	19
Tabela 2.13 PU-13 - Usunięcie użytkownika ze znajomych	20
Tabela 2.14 PU-14 - Przeglądanie ocenionych filmów użytkownika	20
Tabela 2.15 PU-15 - Przeglądanie filmów, które użytkownik chce obejrzeć	21
Tabela 2.16 PU-16 - Wysłanie zaproszenia do znajomych.....	21
Tabela 2.17 PU-17 - Przeglądanie szczegółów filmu	22
Tabela 2.18 PU-18 - Ocenianie filmu	22
Tabela 2.19 PU-19 - Zaznaczenie filmu do obejrzenia	22
Tabela 2.20 PU-20 - Polecanie filmu znajomym	23
Tabela 2.21 PU-21 - Wybór znajomych, którym film zostanie polecony.....	23
Tabela 2.22 PU-22 - Przeglądanie aktorów grających filmie	24
Tabela 2.23 PU-23 – Usunięcie filmu z listy polecanych	24
Tabela 2.24 PU-24 – Wybranie użytkownika, który polecił film	24
Tabela 2.25 PU-25 - Rejestracja.....	25
Tabela 2.26 PU-26 - Logowanie	25
Tabela 4.1 PI-01 - Czas wyszukiwania aktorów do filmu.	36
Tabela 4.2 PI-02 - Wyświetlanie aktorów i opisu filmu w jednej aktywności.	37
Tabela 4.3 PI-03 - Wyświetlanie listy znajomych, polecających pozycje filmowe.....	38
Tabela 4.4 PI-04 – Wystawianie oceny.....	38
Tabela 4.5 PI-05 - Widok startowy.	38

Bibliografia

- [1] Dave Chaffey, Mobile Marketing Statistics compilation, 01.03.2017, [Dostęp w dn.03.12.2017], <https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- [2] Felix Richter, The Smartphone Platform War Is Over, 20.02.2017, [Dostęp w dn.03.12.2017], <https://www.statista.com/chart/4112/smartphone-platform-market-share/>
- [3] Agnieszka Serafinowicz, Android - historia prawdziwa, 12.09.2015, [Dostęp w dn.03.12.2017], <http://softonet.pl/publikacje/poradniki/Android-historia.prawdziwa,1162>
- [4] Statista Inc., Number of available applications in the Google Play Store from December 2009 to September 2017 [Dostęp w dn.03.12.2017], <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>
- [5] Statista Inc., Cumulative number of apps downloaded from the Google Play as of May 2016 (in billions), [Dostęp w dn.03.12.2017], <https://www.statista.com/statistics/281106/number-of-android-app-downloads-from-google-play/>
- [6] Statista Inc., Android version market share distribution among smartphone owners as of September 2017, [Dostęp w dn.03.12.2017], <https://www.statista.com/statistics/271774/share-of-android-platforms-on-mobile-devices-with-android-os/>
- [7] Android, Dokumentacja techniczna, [Dostęp w dn.03.12.2017], <https://developer.android.com/guide/index.html>
- [8] Michał Śmiałek, Zrozumieć UML 2.0, 2007
- [9] Zigurd Dornin, Laird Meike, G. Blake Mednieks, Laird Dornin, Zigurd R Mednieks, Programming Android, 2nd ed, 2012
- [10] Smith, Dave, Hellman, Erik „Android Recipes: A Problem-Solution Approach”, Apress, 2016
- [11] MacLean, Dave, Komatineni, Satya, Allen, Grant „Pro Android 5”, Apress, 2015
- [12] Chacon, Scott, Straub, Ben, Pro Git, Apress Open, 2014
- [13] W3C, Extensible Markup Language (XML), [Dostęp w dn. 03.12.2017] , <https://www.w3.org/XML/>
- [14] Robert C. Martin, „Clean Code”, Prentice Hall, 2008
- [15] Glide, Dokumentacja techniczna, [Dostęp w dn.03.12.2017], <https://bumptech.github.io/glide/>

[16] SDK Facebook, Dokumentacja techniczna, [Dostęp w dn.03.12.2017],
<https://developers.facebook.com/docs/android>

[17] Firebase, Dokumentacja techniczna, [Dostęp w dn.03.12.2017],
<https://firebase.google.com/docs/>