

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: tassioauad

Movie Check

Description

Who has never wanted to find a great movie to see in a weekend or holiday? Who has never started to watch a movie thinking it would be good but it was terrible? Movie Check can help you to stop to lose your time watching movies that you won't like or trying to find a good one.

Movie Check allows you to be informed about upcoming movies, movies that is being played now at cinemas, top rated movies, popular movies and be recommended to watch some movies that you possibly will like. If you liked a recommendation of a movie after get informations about it like description, celebrities that worked in cast or crew, videos of trailers and reviews, you can add it in a list of interests to watch on your free time. Or else, if you has already watched a movie, you can classify it and help the app to learn what you really like.

Intended User

Movie Check is designed for fan movies of any age and sex.

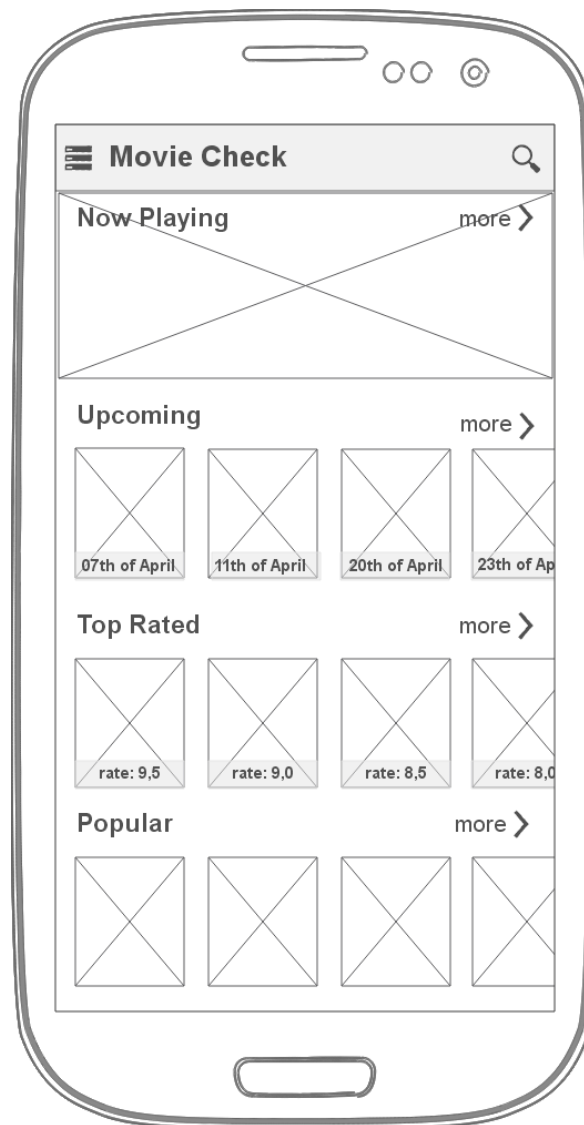
Features

- Informing what movies are upcoming
- Informing what movies are playing now at cinemas
- Informing what movies are top rated
- Informing what movies are popular
- Informing cast and crew of each movie
- Informing works as cast or crew of each celebrity
- Allow the user to inform his classification of an watched movie
- Allow the user to create a list of movies that he wants to see.
- Allow the user to discover movies that he would like to watch and add easily to his list of interests, through of a simple mining of data of classifications and interests.

User Interface Mocks

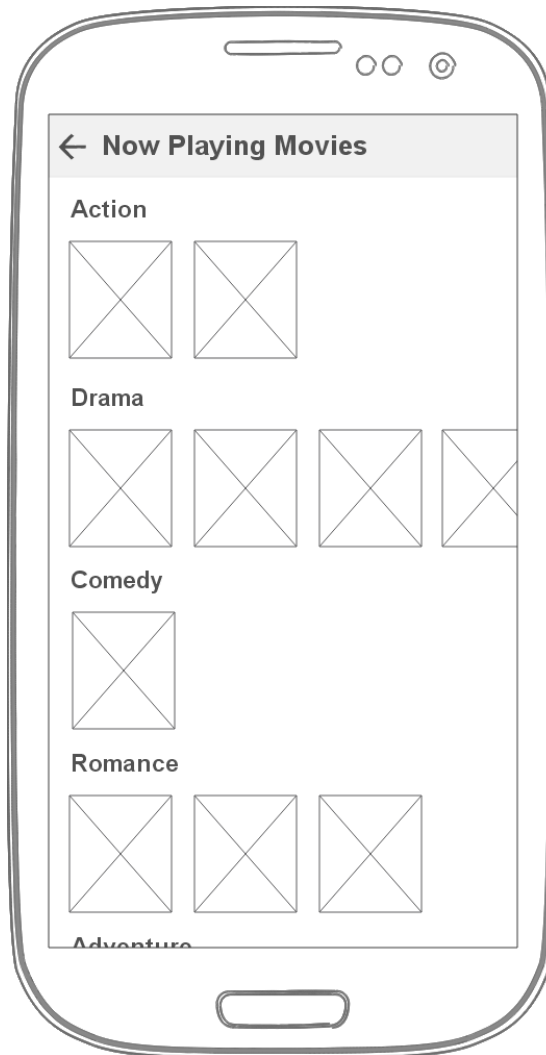
Interface mocks created using [NinjaMock](#)

Screen 1



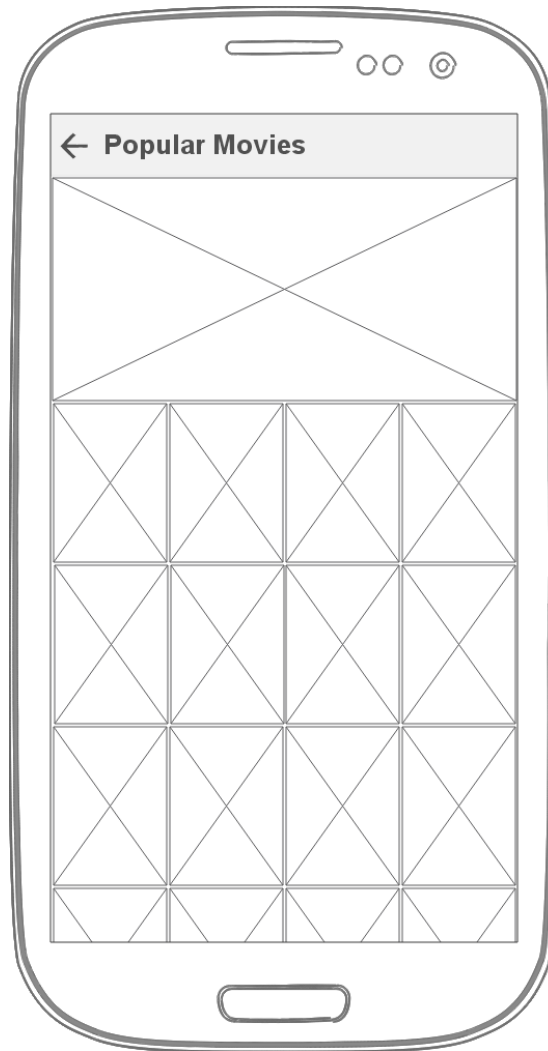
This screen shows the main screen, where the user can understand all options of filter movies (Now Playing, Upcoming, Top Rated and Popular). Clicking in the button "more", the app will show a screen where there are more movies with the filter, for example, clicking in "more" of the filter "Popular", the app will show the screen with all of the most popular movies. Clicking in a film, the app will show a new screen with details about the movie. The user can start a search for a movie or celebrity using the single icon in the actionBar. A NavigationDrawer with all main options of navigations can be called using the icon at right of the application title in ActionBar.

Screen 2



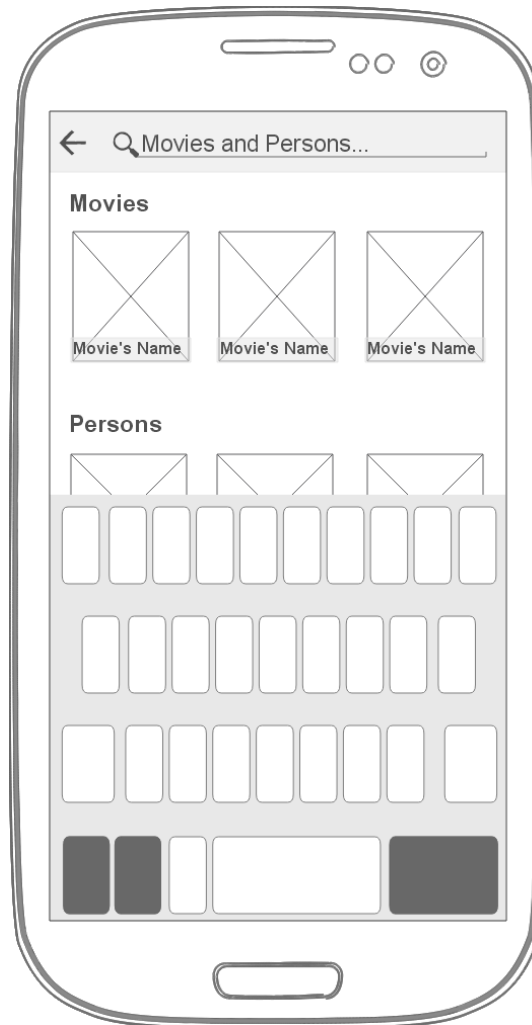
This screen shows the movies that are premiering at cinemas, grouping them by category like Action or Drama. Clicking in one of this image, the app will open a new screen with details about that referenced movie.

Screen 3



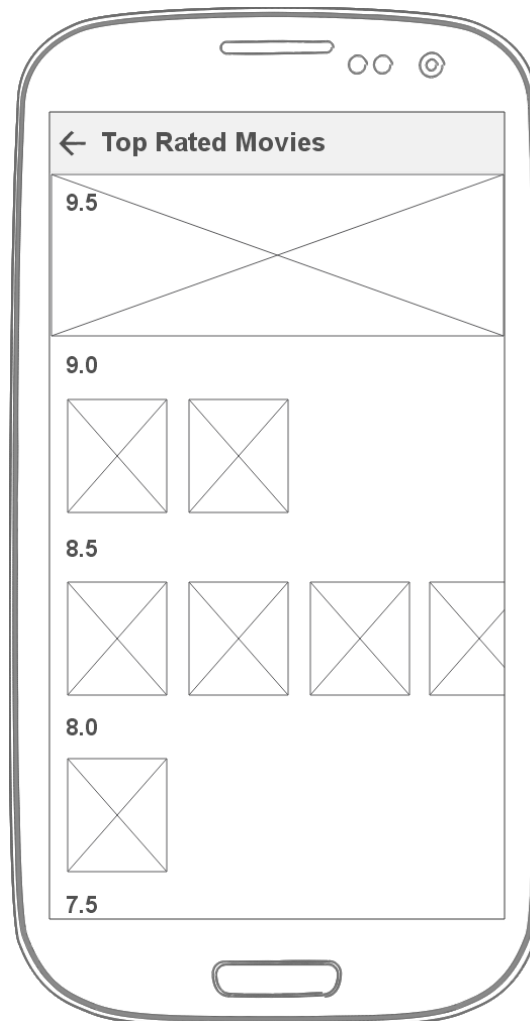
This screen shows the movies that are popular, highlighting the most popular of them. Clicking in one of this image, the app will open a new screen with details about that referenced movie.

Screen 4



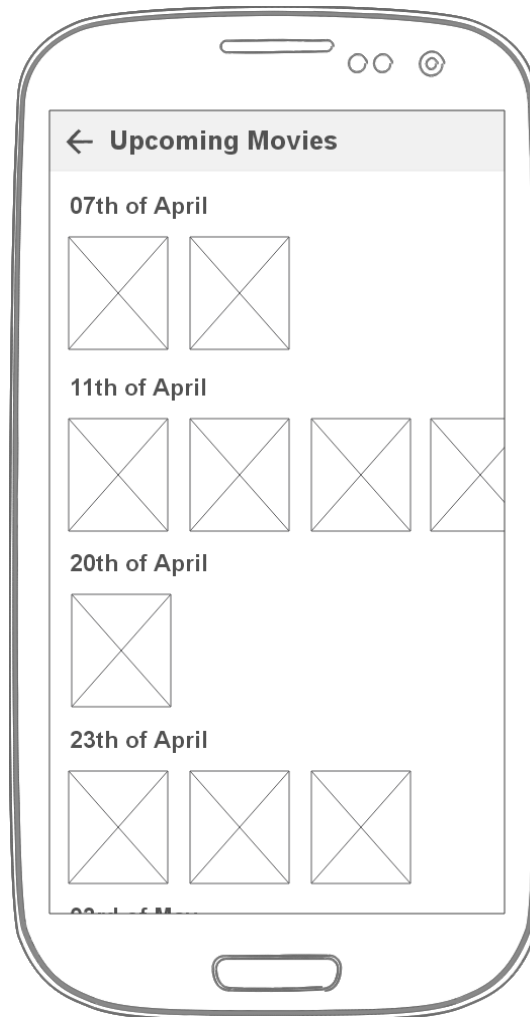
This screen shows a result of a search for movie or celebrities. The user can inform a movie's name or celebrity's name and the app will find movies and celebrities with that match at least approximately with the name informed. The result is grouped by "movie" and "celebrity" and the user has the possibility to click in one of the result to get more information.

Screen 5



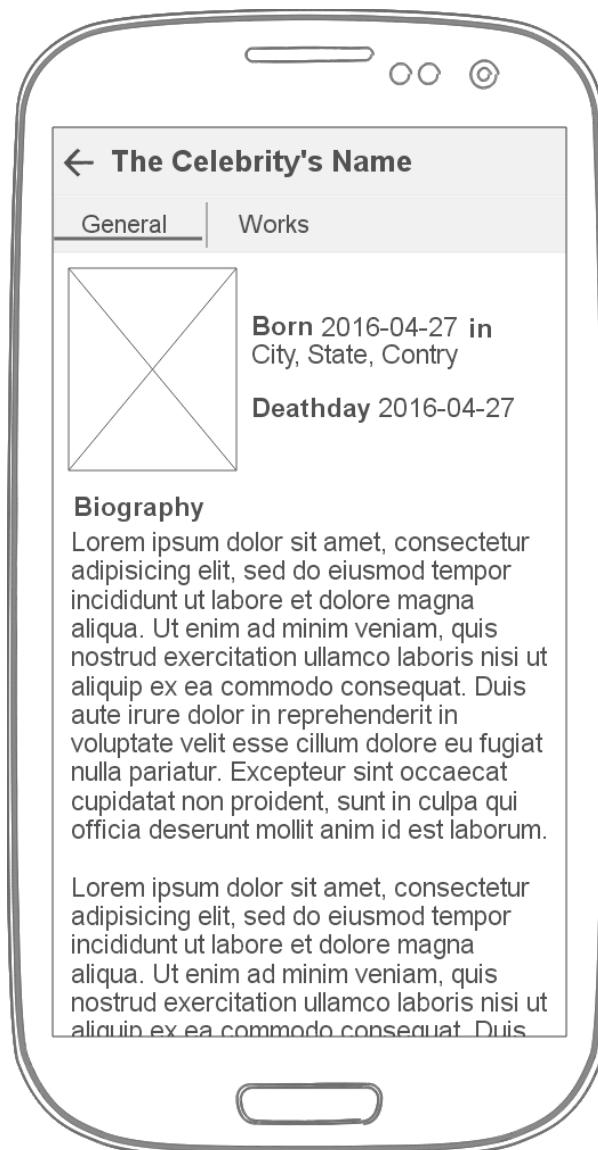
This screen shows the movies that top rated, grouping them by rate like Action or “9.0” or “8.0”. Clicking in one of this image, the app will open a new screen with details about that referenced movie.

Screen 6



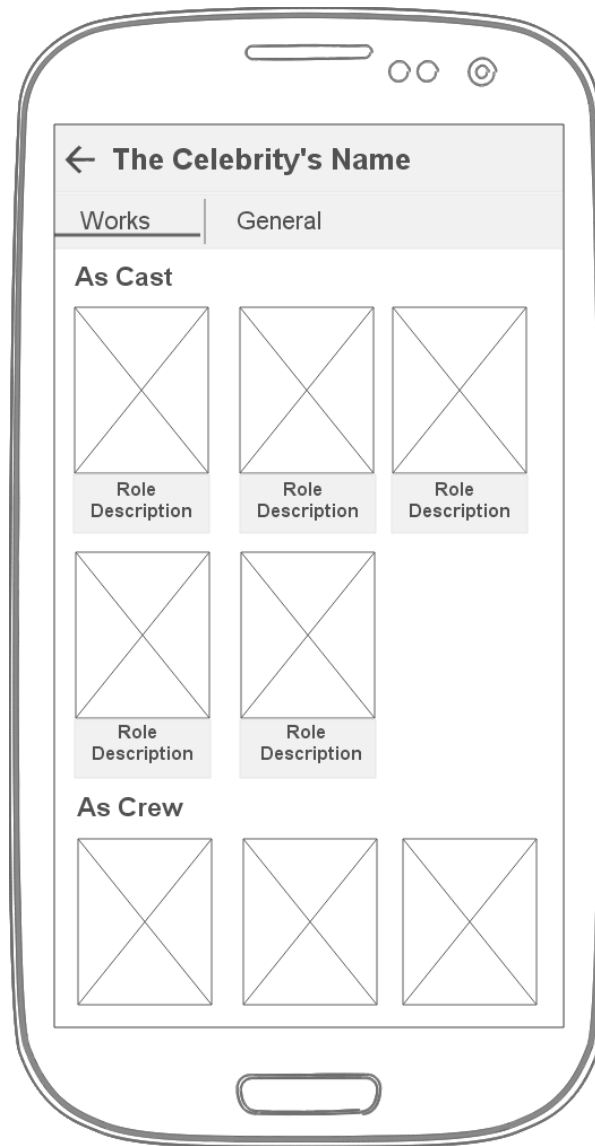
This screen shows upcoming movies, grouping them by release date like “07th of April”. Clicking in one of this image, the app will open a new screen with details about that referenced movie.

Screen 7



This screen shows general information about a celebrity, like biography, born date, where the celebrity was born and, if he has died, the death date. Clicking on the tab “work”, the user will be redirect for the screen where is detailed all of works of this celebrity.

Screen 8



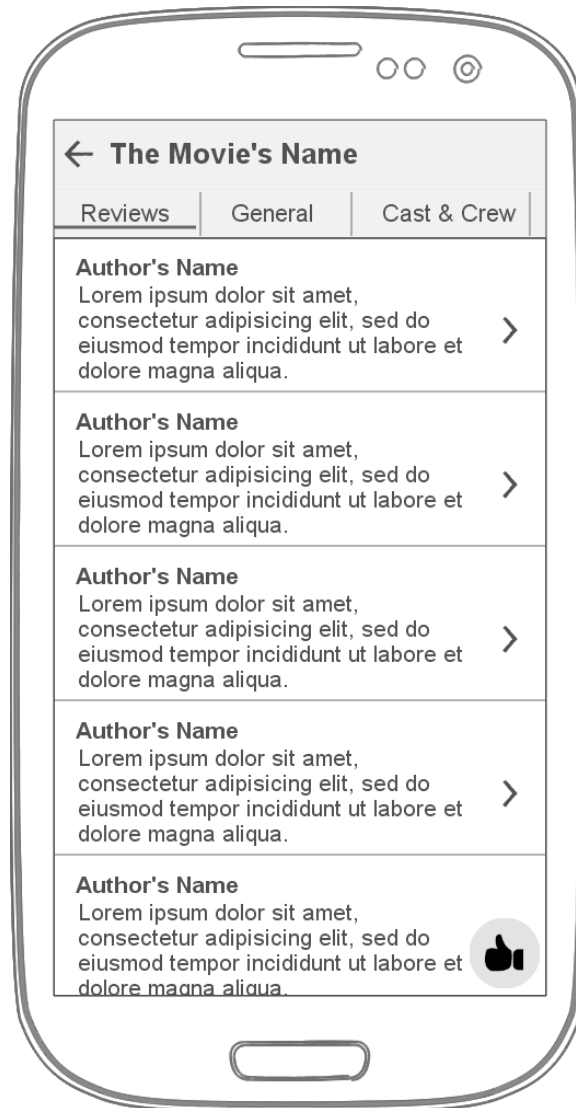
This screen shows all movies that a celebrity has worked as part of cast or crew. For each movie is shows a image and what role the celebrity has performed in that movie. Clicking on a movie, the app will show an screen with some information about the movie.

Screen 9



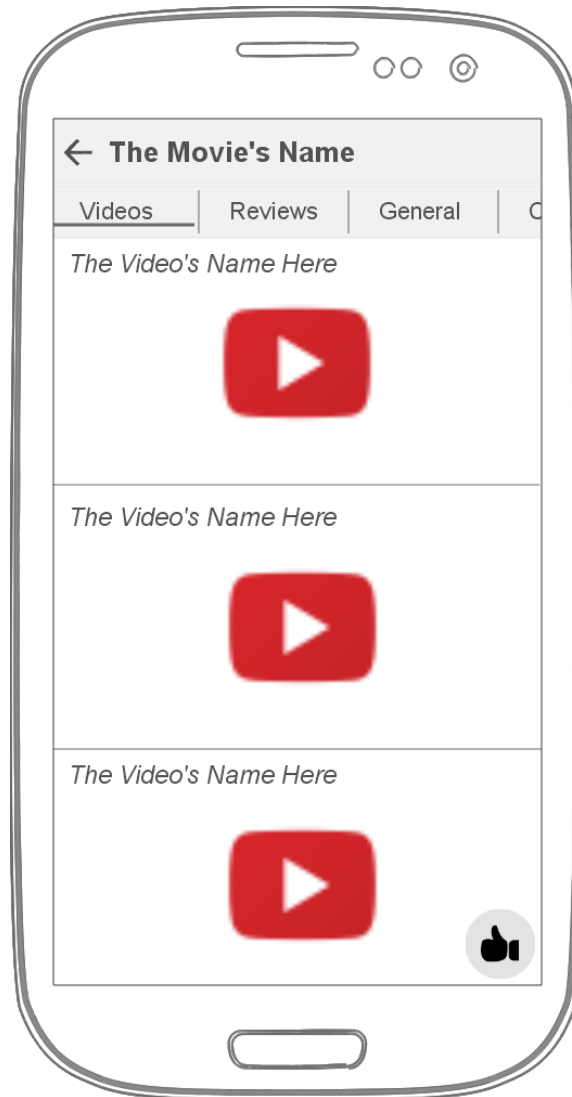
This screen shows general informations about a movie, like its classification, release date, categories and an overview. Clicking in one of the categories, the app will open a screen with good movies of that specific category. The user can classify the movies clicking on the stars that at first shows the average of ratings. Clicking in one of the images, it will be shown in a fullscreen mode. The like button allows the user to check this movie as interesting and add it into the list of movies to watch.

Screen 10



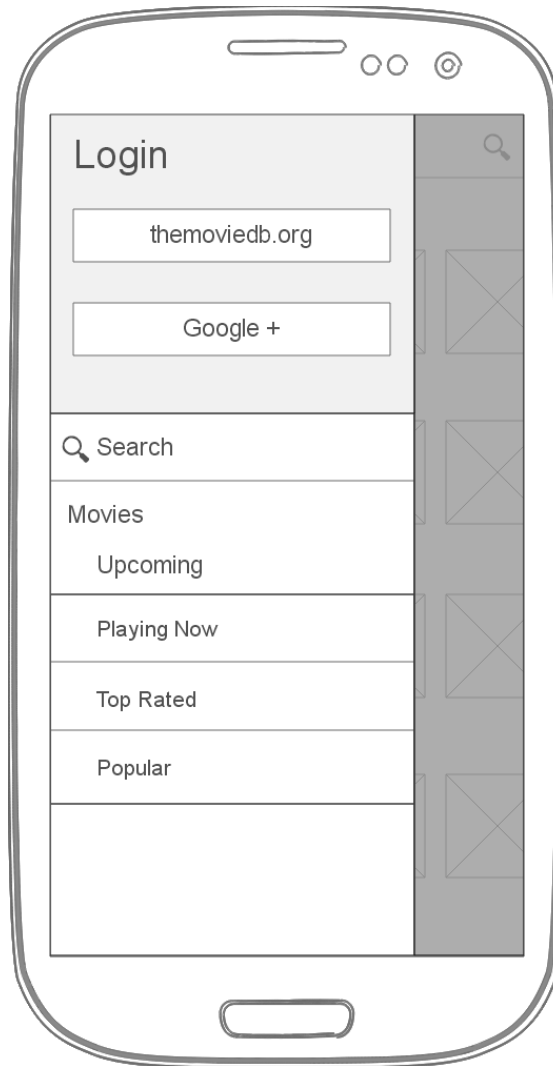
This screen show reviews of the movie, where each review has the author's name and his comment. If the user click in a review, he will be redirect to the web site where the review were made. The like button allows the user to check this movie as interesting and add it into the list of movies to watch.

Screen 11



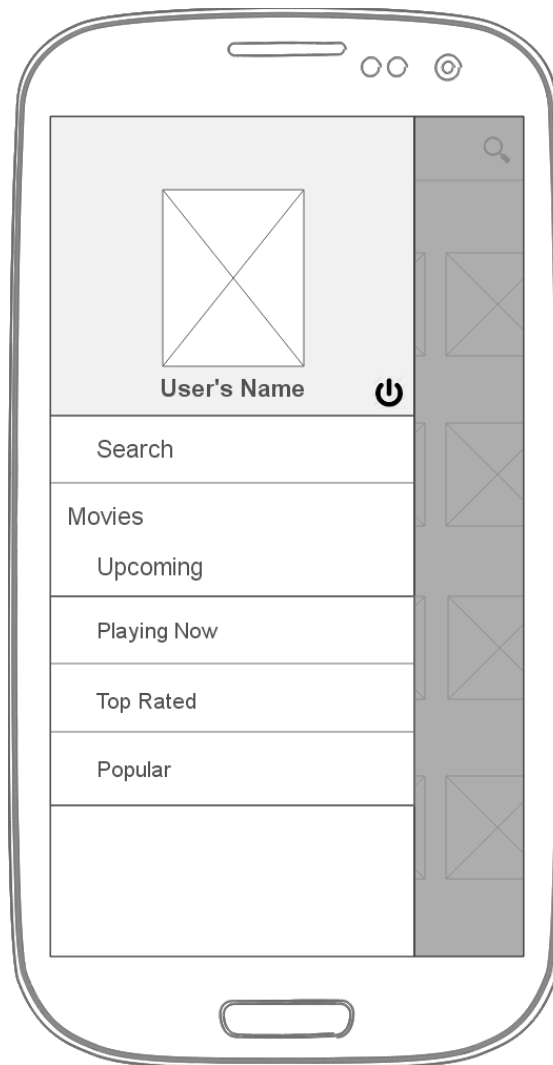
This screens shows videos of the movie, like trailers and teasers that can help the user to get interest to watch it. Clicking in a video, the app will show it in a fullscreen mode. The like button allows the user to check this movie as interesting and add it into the list of movies to watch.

Screen 12



This screen shows the NavigationDrawer that allow the user to see and go to any of the main options of navigation. In this situations, the user is not logged but has the options to do it using themoviedb account or Google+ account. When the user login, he can access his profile with interests, classifications and recommendations.

Screen 13



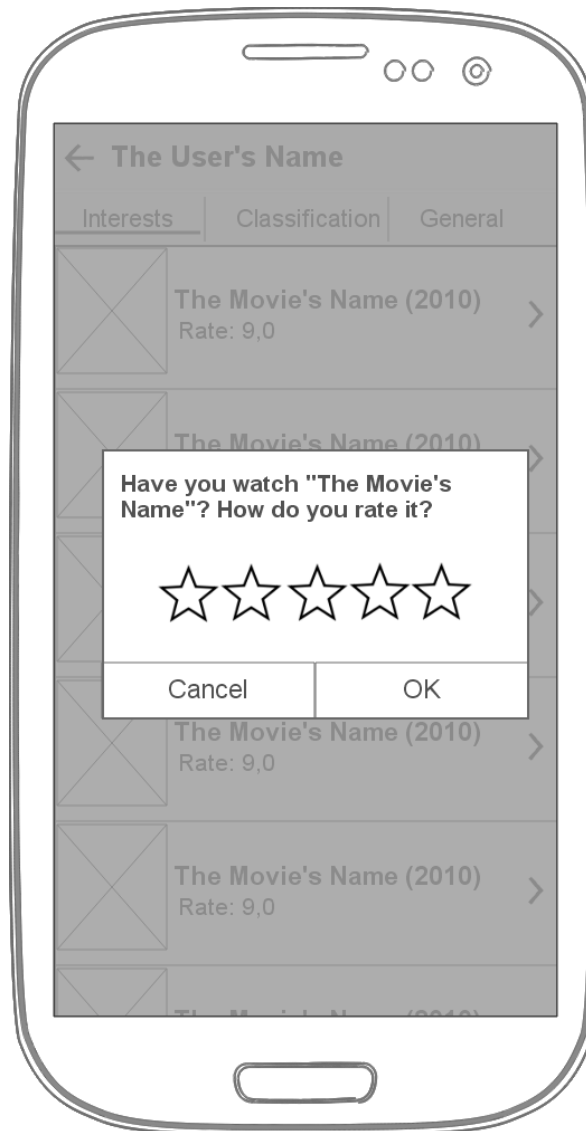
This screen shows the NavigationDrawer that allow the user to see and go to any of the main options of navigation. In this situations, the user is logged and you profile with interests, classifications and recommendations can be accessed clicking in the area of you picture and name. The user can at any moment logout of the app clicking in the “power” button.

Screen 14



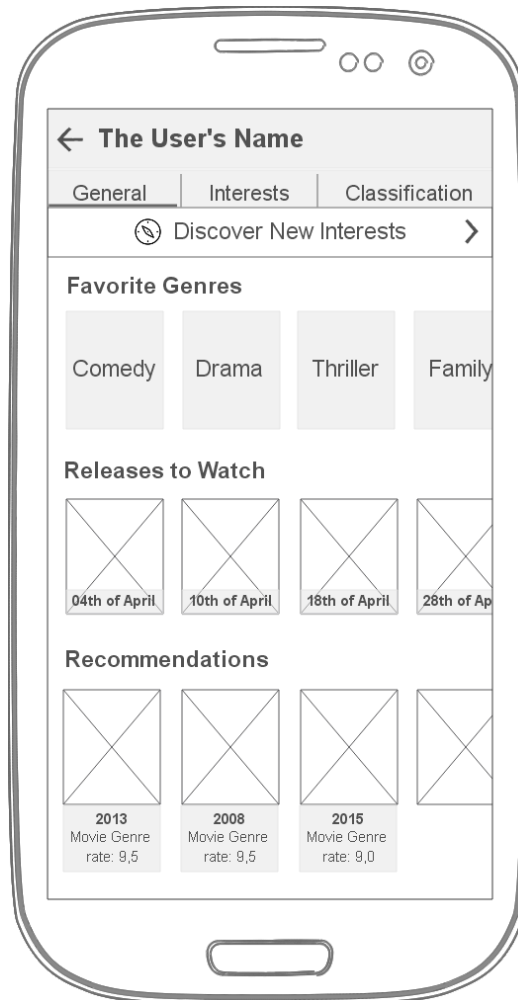
This screen shows all classification done by the logged user. The user can access the movie's profile with more information about it clicking on item of the list. The user can also update your classification clicking directly at the start.

Screen 15



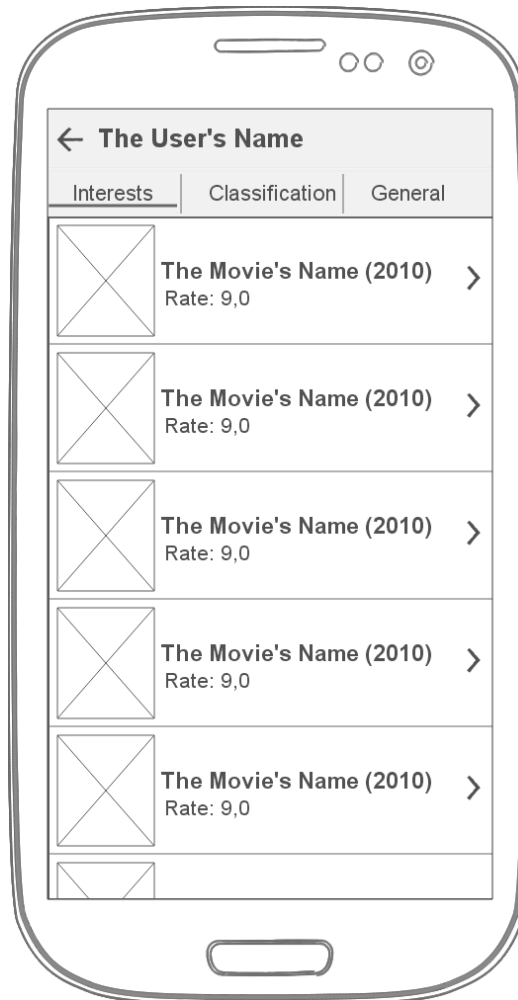
This screens shows a dialog where the user can classify a movie that he has watched. The dialog can be closed clicking outside of it or clicking at the button "cancel". Clicking on the button "OK" the classification will be saved.

Screen 16



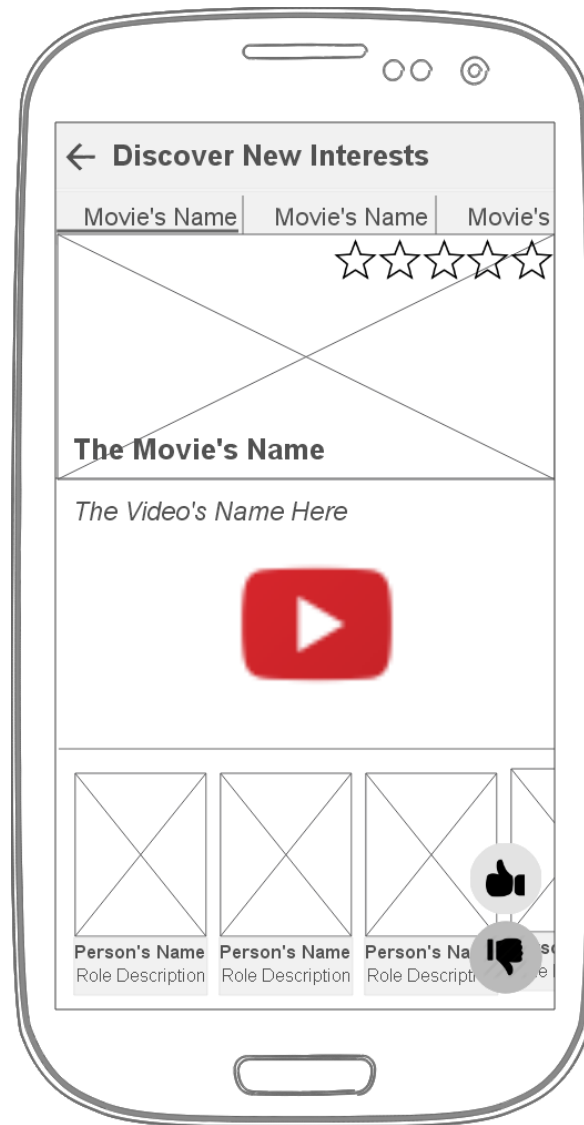
This screen is the main screen of the logged user profile. This screen shows the user favorite genres, next release of movies that the user has check as interesting and the user can get some simple recommendations of movies. Clicking in one of the genres, the app will show a screen with other movies of that specific category. Clicking in one of this movie image, the app will open a new screen with details about that referenced movie. The user can start the discover process clicking on the button right below the tabs.

Screen 17

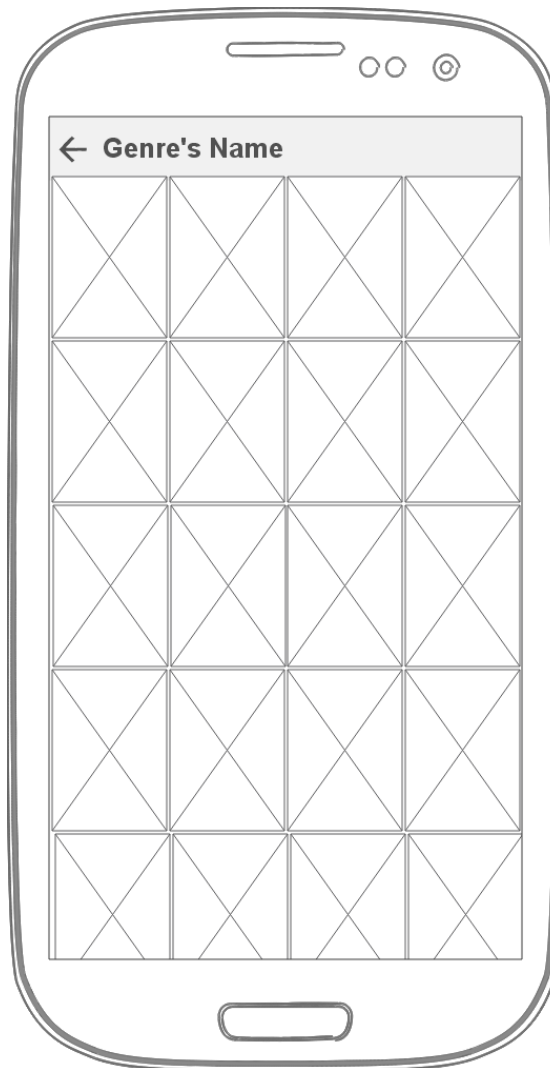


This screen shows all movies checks as interesting (like button) done by the logged user. The user can access the movie's profile with more information about it clicking on item of the list.

Screen 18

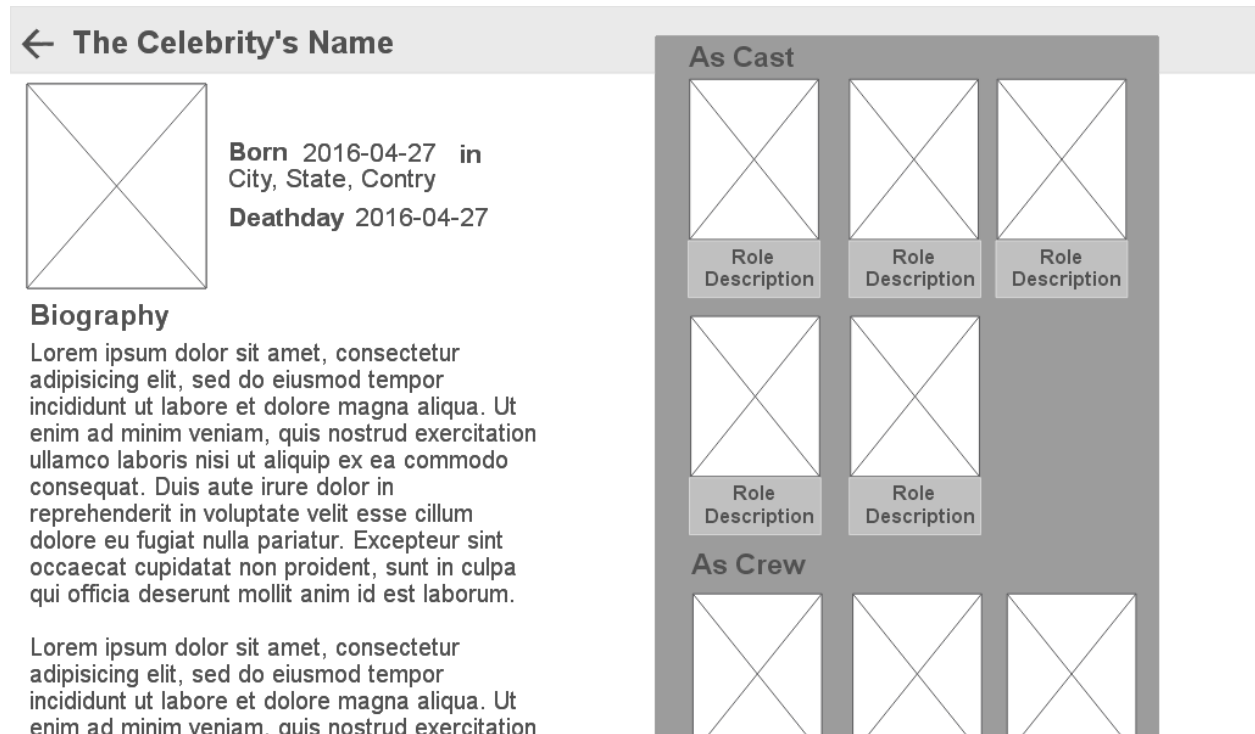


In this screen movies are recommended to help the user to discover one that probably the he will like. The most necessary details about a movie are shown in this screen in a way that get easier to the user decide if he is interested to watch or not. Clicking in one of the images, it will be shown in a fullscreen mode. Clicking in a video, the app will show it in a fullscreen mode. The “like” button allows the user to check this movie as interesting and add it into the list of movies to watch. There is also a “dislike” button that inform to the app the user are not interesting in that movie. The user can classify the movies clicking on the stars that at first shows the average of ratings. In any case, clicking on “like”, “dislike” or classifying a movie, the app will automatically move forward and show the next movie (next tab).

Screen 19

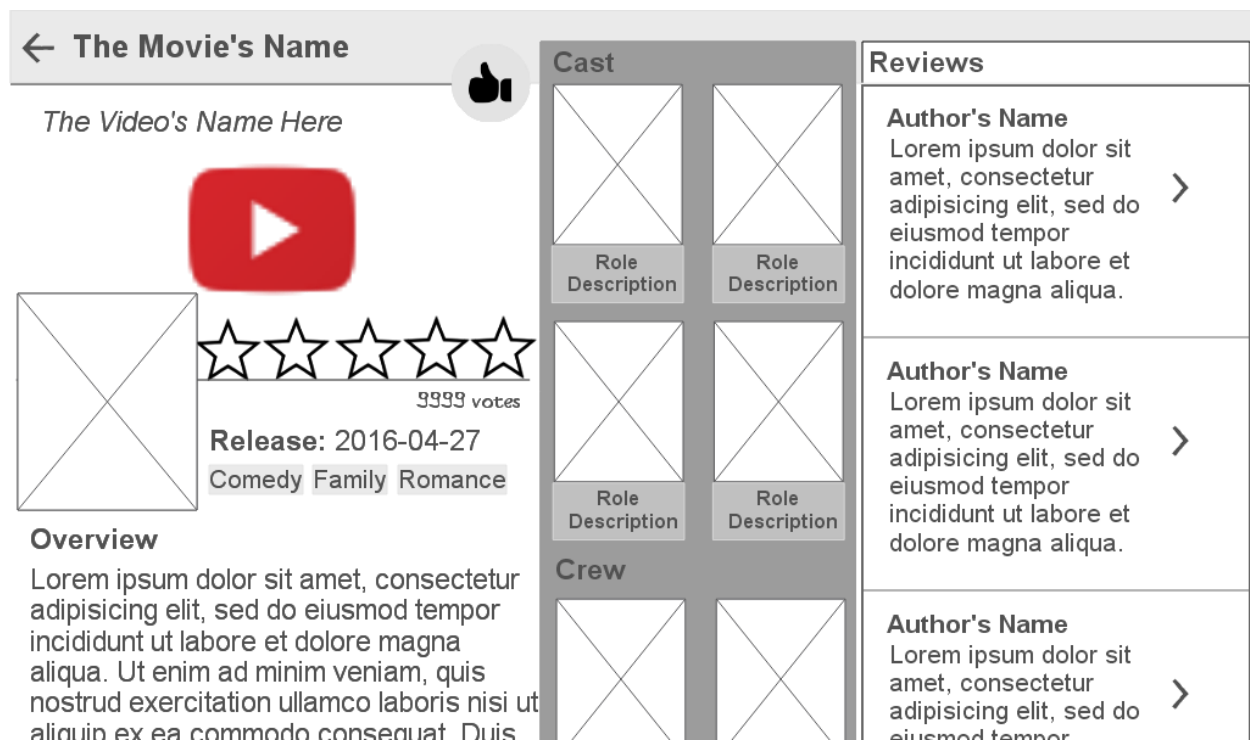
This screen shows all movies of a specific genre. Clicking on a movie, the app will show an screen with some information about the movie.

Screen 20



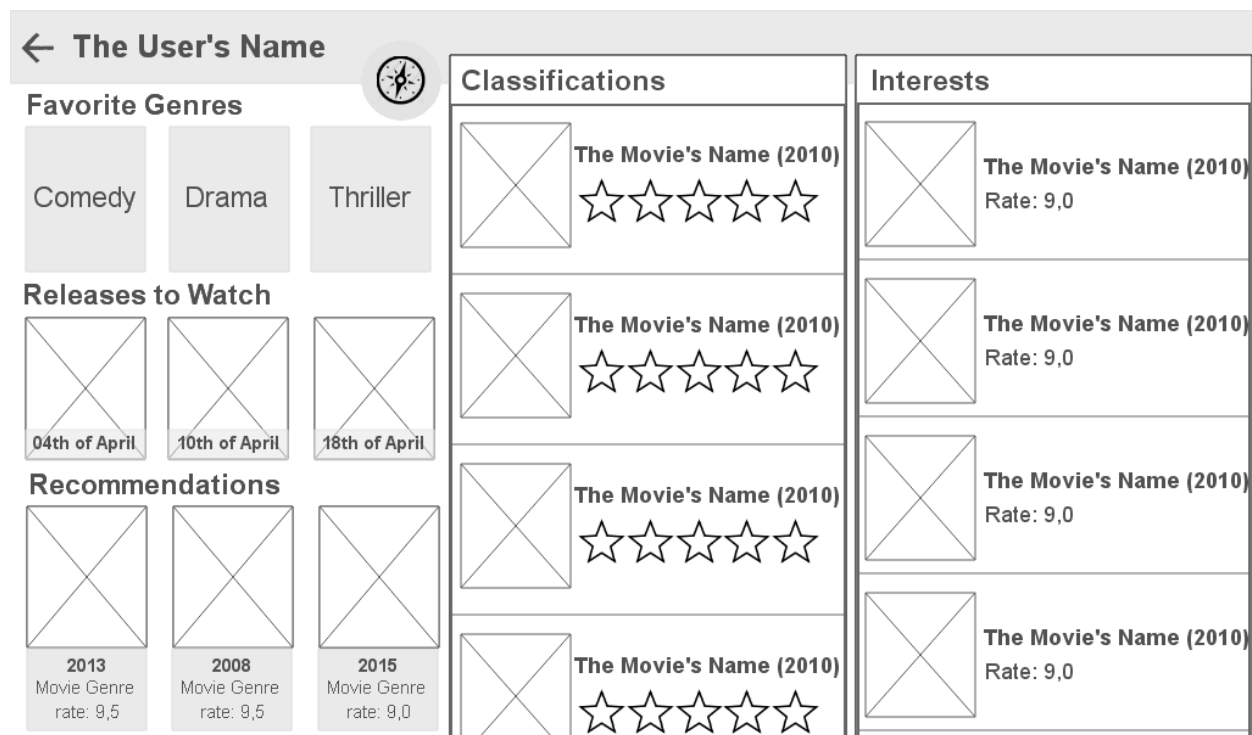
This screen designed specifically for tablets in a landscape mode. This screen shows all of informations of a celebrity. This screen shows general information about a celebrity, like biography, born date, where the celebrity was born , all movies that the celebrity has worked as part of cast or crew and, if he has died, the death date. Clicking on a movie, the app will show an screen with some information about the movie.

Screen 21



This screen designed specifically for tablets in a landscape mode. This screen shows all of informations of a movie. This screen shows general informations about a movie, like its classification, release date, categories and an overview. Clicking in one of the categories, the app will open a screen with good movies of that specific category. The user can classify the movies clicking on the stars that at first shows the average of ratings. Clicking in one of the images, it will be shown in a fullscreen mode. This screen also show reviews of the movie, where each review has the author's name and his comment. If the user click in a review, he will be redirect to the web site where the review were made. This screens also shows videos of the movie, like trailers and teasers that can help the user to get interest to watch it. Clicking in a video, the app will show it in a fullscreen mode. The like button allows the user to check this movie as interesting and add it into the list of movies to watch. The like button allows the user to check this movie as interesting and add it into the list of movies to watch.

Screen 22



This screen designed specifically for tablets in a landscape mode. This screen is the main screen of the logged user profile. This screen shows the user favorite genres, next release of movies that the user has check as interesting and the user can get some simple recommendations of movies. Clicking in one of the genres, the app will show a screen with other movies of that specific category. Clicking in one of this movie image, the app will open a new screen with details about that referenced movie. The user can start the discover process clicking on the button right below the tabs. This screen also shows all movies checks as interesting (like button) done by the logged user. The user can access the movie's profile with more information about it clicking on item of the list. This screen also shows all classification done by the logged user. The user can access the movie's profile with more information about it clicking on item of the list. The user can also update your classification clicking directly at the start.

Key Considerations

All informations about movies are get from [TheMovieDB API](#). Classifications of movies and movies that the user has interest to watch are saved in a local database.

At the first time the app is started, will be allowed to user set some preferences. This preferences will be used to find better recommendations of movies and will be saved in Shared Preferences.

Even with the app closed, the user will be notified when a movie of his interest is premiering at cinemas.

The project depends of libraries like [Dagger](#) to dependency injections and [ButterKnife](#) UI dependency injection, [Gson](#) to handle Json data get from API, [Retrofit](#) to API requests, [Picasso](#) to handle the loading and caching of images. Thinking on unit tests will be used libraries like [Espresso](#) to Android UI tests, [Mockito](#) to unit tests with mock objects, [JUnit](#) to traditional unit tests, [JaCoCo](#) to coverage unit tests.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Setup project on Android Studio, configura library and start to manage the tasks.

- Create a board com Trello to manage the project
- Create and configure project on Android Studio
- Set internal libraries on Gradle.

Task 2: Main Screen

Implement the main screen with all related screens.

- Implement Main Screen
- Implement Upcoming Movies Screen
- Implement Top Rated Movies Screen
- Implement Now Playing Movies Screen
- Implement Popular Movies Screen

Task 3: Movie Profile

Implement the movie profile screen with all related screens.

- Implement the Screen with General Informations about the Movie
- Implement the Screen with Cast and Crew Informations
- Implement the Screen to Watch Trailers and Teasers
- Implement the Screen with Reviews about the Movie

Task 4: Celebrity Profile

Implement the celebrity profile with all related screens

- Implement the Screen with General Informations about the Celebrity
- Implement the Screen that Shows All Works of the Celebrity.

Task 5: Search and Genres

Implement the search of celebrity and movie screen and the screen of movies by genre.

- Implement the Search Screen
- Implement the Movie by Genre Screen

Task 5: User Profile

Implement the user profile with all related screens

- Implement the Screen with General Informations about the user preferences and recommendations.
- Implement the Screen with User Classifications
- Implement the Screen with User Interests
- Implement the Discover Screen.