Technical Project Report - Android Module

Movie Emotions

Course:	44139- Computação Móvel
Date:	Aveiro, November 5th, 2017
Authors:	71883: João Melo 73046: Tiago Henriques
Project abstract:	Movie Emotions is a mobile application developed for the Android System. This app main objective is to trace the psychophysiological profile of movies through a collection of emotions/feelings felt by the users, this are acquired by a facial recognition mechanism while the user watches several movie trailers. This is used to create personalized movie suggestions that take into account people's personalities preferences by a certain movie type/genre.

Table of contents:

1 Introduction

This project is related to the Android Module of the CM course and it's main objectives are to improve our skills on the Android Language.

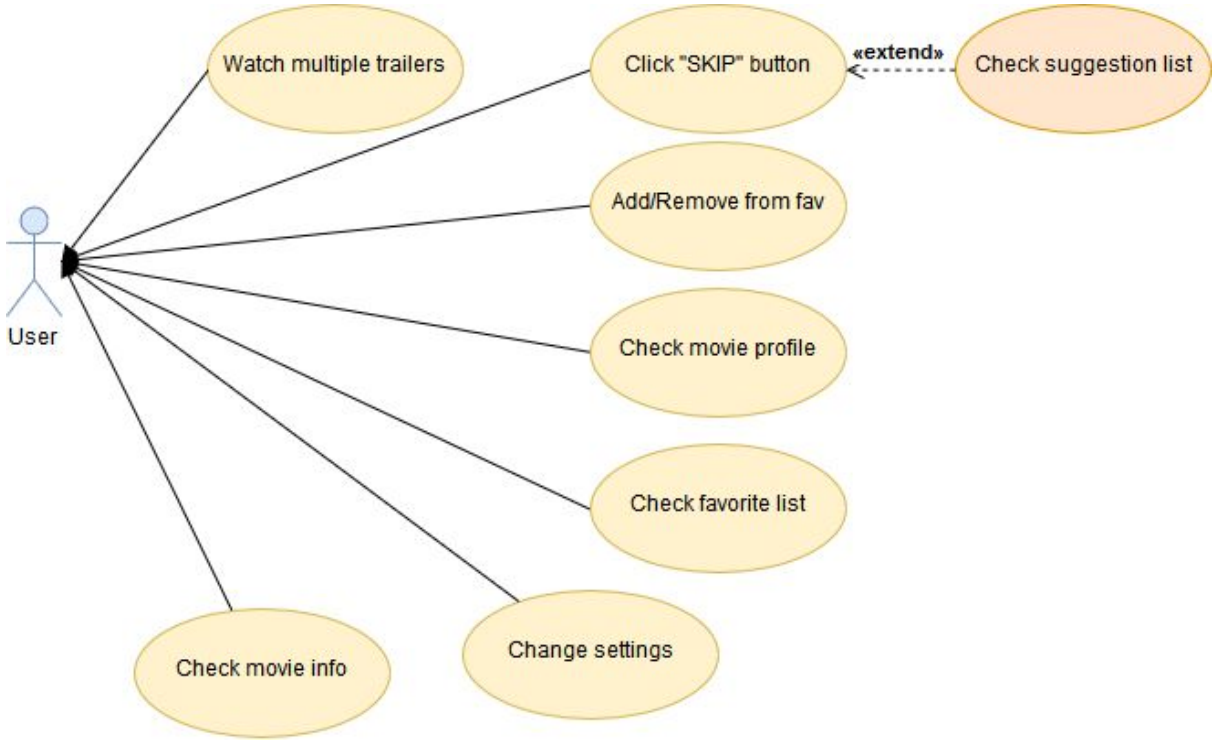
The first project that was proposed was a kind of project that would turn your cellphone emotional. So, it is possible to say that it would react depending on the emotion that the user would demonstrate. But this felt decoupled from the user experience, so when talking with the Professor, it was decided to put the emotion analysis part more hidden from the user experience and create an app that uses this, but in an transparent manner to the user.

With this, it was decided to create an app that would recommend movies. To allow this feature, we show several movie trailers to the end user, random trailers each time, and from different genres/types. While watching the trailer, several emotions of the user are being detected by a facial recognition system with the support of the front camera. We are detecting emotions like Joy, Fear, Attention, Smile... When the user decides to stop watching trailers, based on the values collected, our app is going to suggest a list of movies. The user can then check movie informations like title, synopsis, release date, vote average, and add specific movies to the favorites list so he can check them out later. Also, the user can check the psycho emotional profile of a specific movie.

This app chooses the right movie for you based on your emotions/feelings on several random movie trailer. It solves the need that people usually have to do a deep search or go to netflix/imdb to be able to listen to other's people movie review. With this app, every user can have a personalized list of movies to watch that matches his movie's favorite characteristics.

2 Application concept

In the next diagram it's possible to check several of the most important use cases that a user can perform in our app.

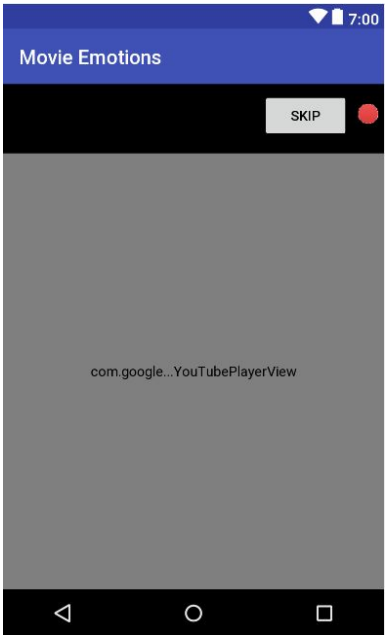


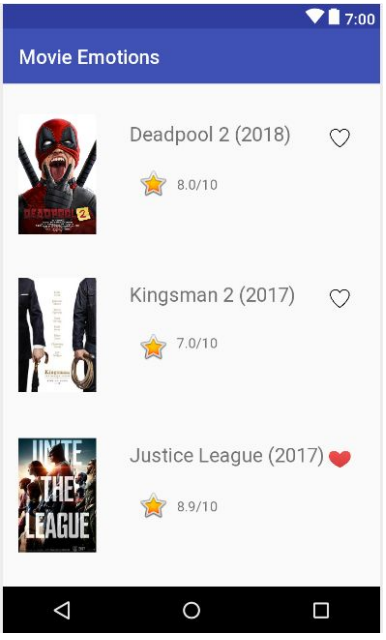
The application constantly monitors users' emotions/feelings using Affectiva facial recognition SDK, a company co-founded by Rana el Kaliouby and Rosalind Picard who specialize in software to handle and sense human emotions.

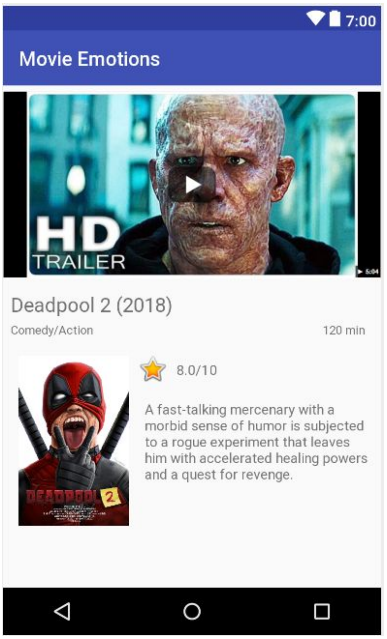
The users of the app are the smartphone users themselves. The only requirement asked to the user is that his face is aligned accurately with the front camera, so that it is possible to use the front camera for detecting emotions/feelings while the user is watching a random movie trailer. The user, while watching a movie trailer, is aware that facial recognition is actually working through visual feedback represented by a circular symbol, green colored when it is detecting, and red colored when it is not detecting the user face because it is not centrally aligned with the front camera. Other purpose of this application is to trace psychophysiological movie profiles through the collection of user's emotions/feelings.

3 User experience design process

One of the things that it was decided to do was to show the minimal amount of information of the recognition pattern to the end user, because this would be boring and meaningless to him. The user has always a colored circle to check if the recognition mechanism is working (green/red), and he can choose to see the camera surface or not. The most curious ones can see the emotion values on a file stored on the system's internal storage. The system was designed to be very simple and intuitive to use. On the main screen, if the user clicks on the "start" button, he can start watching movie trailers (Image 1). When he has watched a movie trailer (Image) or if he wants to check the movies suggestions list , he then can click the "skip" button. A list of suggested movies (Image 3) will appear and by clicking on a movie on the list, the user gets the relevant information for that movie. If the movie actually meets the user's preference, he can add the movie to the favorites through a click on the "heart" icon.







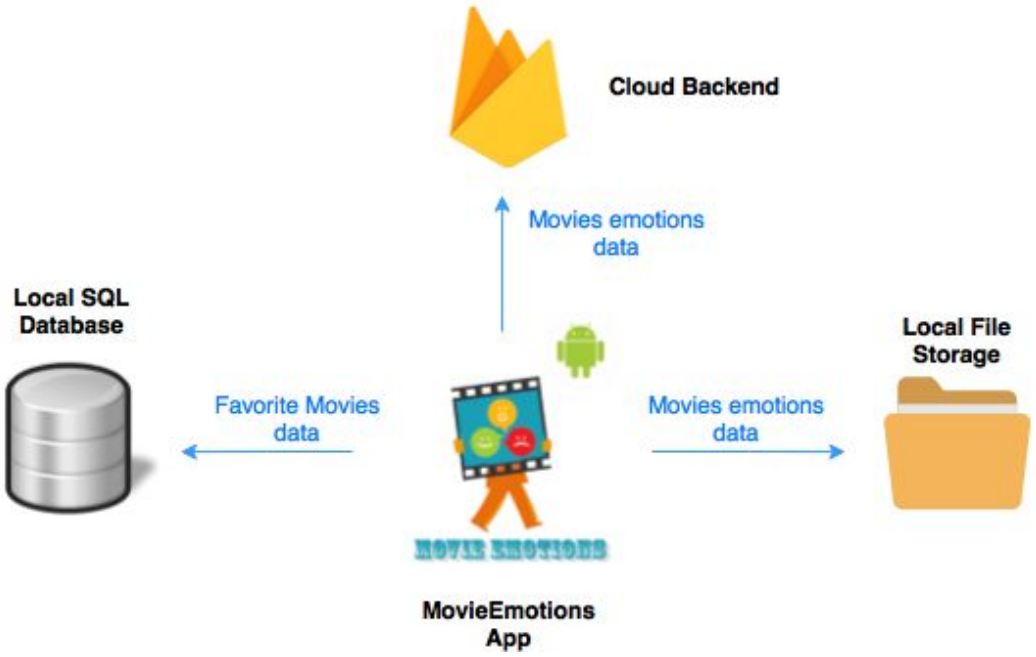
On the previous images, are shown the first activities prototypes of the application to be developed. Our final product is slightly different but the main features remained untouched. On the images we can see the main screens that the user will get on his experience. The user will start on the first image that is the home page, from here he can start the experience, go to the favorites, check a movie profile and even go to the settings, to show or hide the camera surface.

After starting the experience, the user will get the second image that will display a trailer. If he wants to check the movies suggestion list, he should click on the “skip” button.

After clicking the button, it will change to the third image, that shows a personalized list of suggestions. On the final product, the information in it is slightly different and the favorite button is inside the movie information window and not on the list.

If the user likes a movie and clicks on it, it will change to the fourth image that has relevant information of the movie. Just like the previous screen, here the changes were minimal and we only took the trailer off to best fit the screen layout.

4 Architectural plan for the solution



The image immediately above represents the global architecture of our application. The Backend is controlled by the Firebase platform, allowing to storage the emotional information from movies. The mobile application developed in Android, which will be installed on the user's mobile devices, saves the favorite movies on internal storage in a SQL database and saves movies' emotional information also on the internal storage in a text file.

Persistence:

Regarding persistence, and has also already mentioned, it was used a SQL Database for local storage of favorite movies. The data related to certain movies' emotions/feelings was also stored in the local storage, in .txt file format.

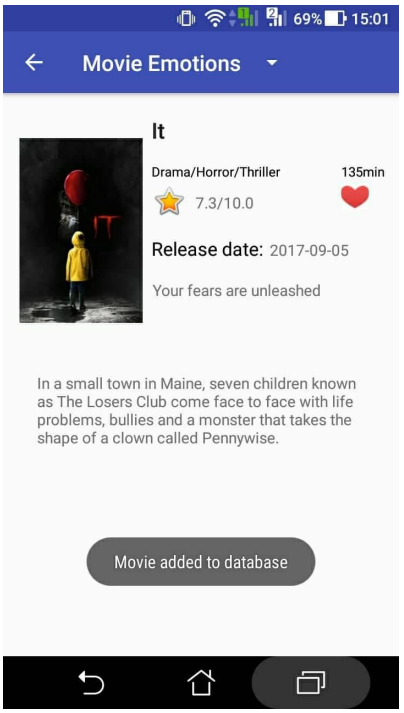
We also use a firebase database, in this database every time a user stops watching the movie trailer the emotions he felt on that particular movie are sent to the database to later analysis and creation of an emotion profile.

Synchronization strategies:

The main data synchronization strategy used was AsyncTasks. This are able to perform tasks in the background thread in order to later update the User Interface. Examples of this use are the video trailers' upload from youtube and collecting information from the API to populate the list.

Notification mechanisms:

As a notification mechanism, Toasts were used for instant visual feedback of an action, for example, when movies are being added or removed from the favorites list (SQL Database).



Data Models:

For data models, we have 7 different data types. Four of this (EmotionValues, SaveToFile, ToFirebase, ValuesToStore), are used mainly to store data on the device's internal storage or on the cloud database. The other three are MovieInfo, that it's used to represent a movie when we get it from the API. It is stored the title, id, poster, overview, release date and vote average. The other one is MoviesSuggestionInfo where we have the list of trailers the user will watch and the trailers that we had watched in this session. The last one, MovieUserProfile, is used when we get the movie profile from the firebase database and we store the username of the watcher, movie name, movie genre and map with the name of the emotion and it's value.

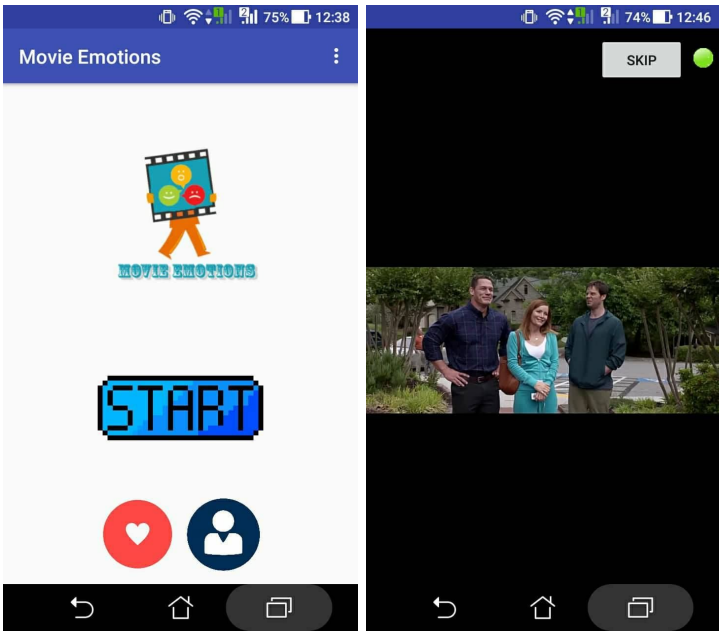
For the content update, it is used the Movie Database API (https://www.themoviedb.org/), and we are getting the data in real time every time we do a suggestion list.

5 Implemented solution in Android

In the Android application that was developed were used several technologies/strategies acknowledged in the practical classes of the curricular unit.

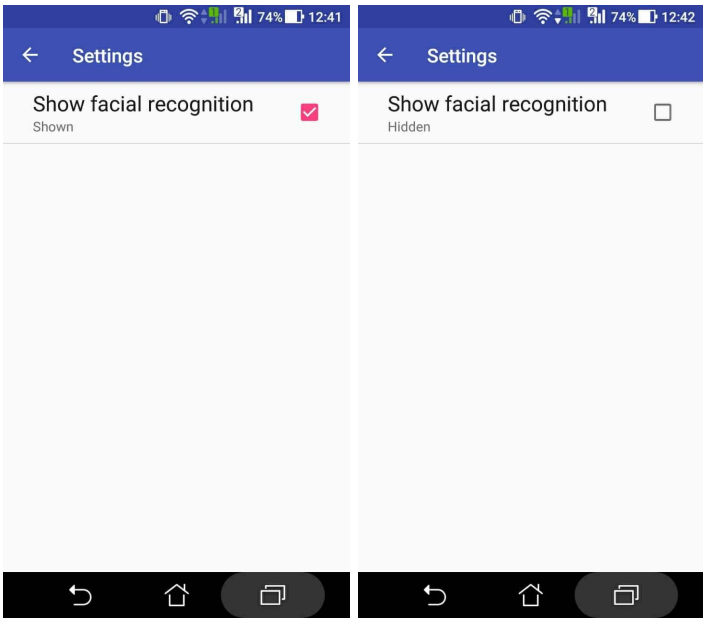
Activities and Intents:

The application's presentation layer consists in several Activities that are interconnected with each other and that were declared in the "AndroidManifest.xml" file. The launcher Activity is called "MainActivity" and allows, through "Intents", to initialize other activities such as the Activity responsible for launching the Camera (Camera), SettingsActivity, FavoritesActivity and MovieEmotionalProfileActivity.



Shared Preferences:

"SharedPreferences" was used in the Settings Activity to save the Boolean value depending on whether or not the face recognition process is presented to the user.

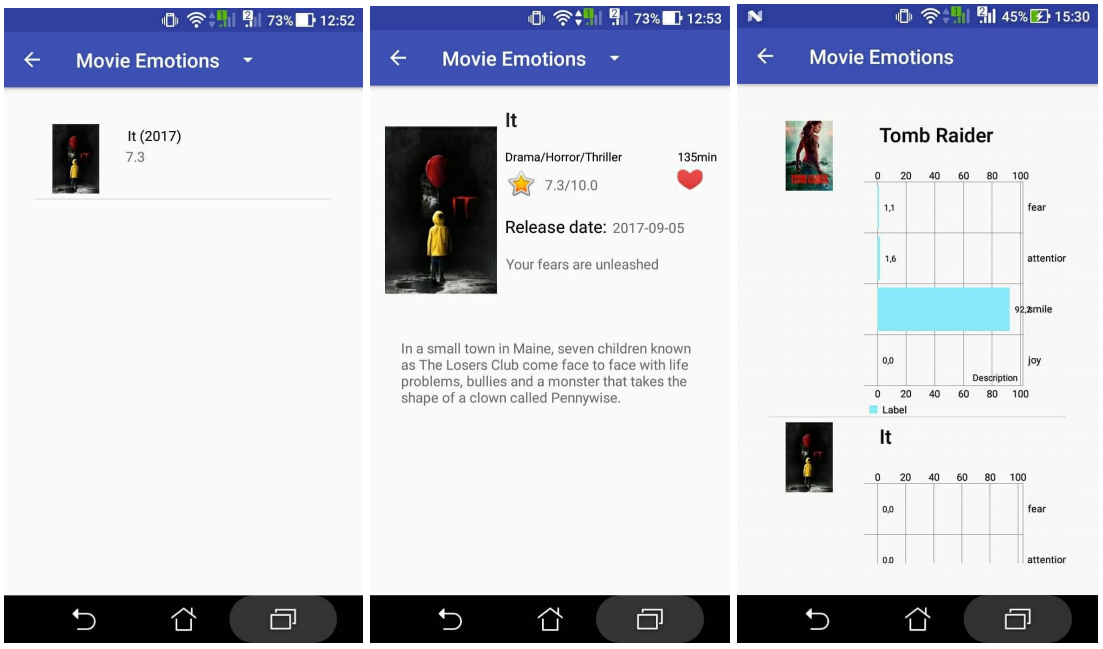


AsyncTasks strategies:

AsyncTasks strategy is widely used in the application for example when uploading videos from Youtube and loading the data from the api to collect movie information. The api used was "The Movie Database API" as mentioned above because it provided in a simple and fast way, all the information relevant and necessary for our application. The api allowed, for example, to search for films by name and by genre (useful for the Activity of suggested films).

Fragments:

Fragments were also used on the project, when populating activities in runtime (example of FavoritesActivity and SuggestionsActivity), that is, fragments transactions were used to allow passing movies' information between fragments and consequent Activity, using callbacks implemented through interfaces.



Databases:

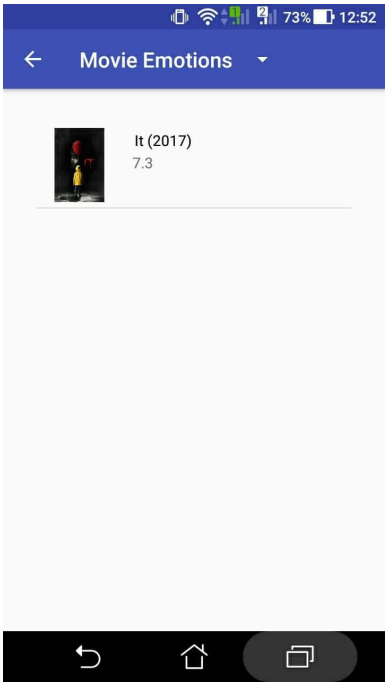
SQL databases were also used for internal storage of favorite movies for each user. For this, a Contract was initially defined (MovieContract) where the tables and the columns of the Database were specified and later it was declared a class denominated MovieDbHelper that extends SQLiteOpenHelper responsible for the creation of the database and corresponding tables.

After watching the trailers all of the emotion values of the user will be sent to a database NoSQL on the firebase cloud, this is done by simply sending a json object to the database previously configured.

This is used when the user accesses the movie profile. It will get in real time all the emotions displayed for each movie seen and make an average value of them, showing then the movie and average values of each emotion. This is done by first getting every value with json, putting it in appropriate data models and in the end calculating the average of them.

No Internet connection:

If the user does not have Internet access, the user will not be able to access all the features offered by the application, because much of the data to be obtained depends on the viewing of trailers of videos uploaded from Youtube. However, the user will be able to view the information stored in internal storage such as favorite movies and the emotional movie data recorded when viewing trailers when with Internet access.



On the picture above, even if the user has no internet connection, he still is able to access data stored on the device's' internal storage such as movies on favorites list.

6 Conclusion

The main project goals were fulfilled because with the application developed it is possible to trace psychophysiological movie profiles through the collection of information coming from the detection of emotions/feelings by facial recognition.

The main problems were that the youtube player doesn't let overlay components to its fragment, so we had to change the fragment size and put everything on top of it. We would like to change the design, put it more user friendly and more material design. Another thing that we would like to improve is the suggested movie algorithm and use a neural network to this end, this way the suggestions would be even more personalized.

7 References and resources

https://developer.android.com/index.html

https://developer.affectiva.com/

https://www.themoviedb.org/

https://developer.android.com/design/material/index.html

https://firebase.google.com/docs/android/setup

Project resources:

Project resources for the Android module:

- Code repository: https://github.com/daweasel27/CM_17_18
- Ready-to-deploy APK: https://github.com/daweasel27/CM_17_18/blob/master/ProjectoFinal/ProjectoFinal/app/release/app-release.apk