

# FACULTY OF INFORMATION TECHNOLOGY BRNO UNIVERSITY OF TECHNOLOGY

## Application Development for Mobile Devices MOVIE SWIPER

Adam Švenk, Bc. (xsvenk00)

Jakub Smejkal, Bc. (xsmejk28)

Šimon Šesták, Bc. (xsesta06)

Peter Vinarčík, Bc. (xvinar00)

30.12.2021

[Github repository link](#)

## APPLICATION DESCRIPTION

The goal of this project - Movie Swiper, is to offer the user an application, where the users can easily choose a movie to watch based on their preferences. The application works in a very straightforward way. It forces the user to swipe left and right. If the user swipes left it means, that he does not want to watch the offered movie, and if he swipes right, it means the opposite - he prefers that movie and the user wants to add it to his favorite movies.

Moreover, it offers users to create their private groups with friends, partners, etc. In those groups, it will match their marked favorite movies and the application shows only movies, which fulfills preferences and selection of all members situated in a group.

### Recommendations

For the recommendations, we made a decision that we will use the genre of a movie as a key factor. This means that if the user swipes left/right the value for all the genres that the movie is described with will be updated in the database based on the decision. More movies are selected based on these genre preferences as well as the group recommendation. A possible upgrade would be to use more information about the movie like actors, directors, runtime, rating, etc. All this information can be extracted from the TMDB but it would take a lot of database data to be stored.

To get more movies we used the inbuild API call that TMDB provides. You just have to put the data together and send the request. As a response, you will get movies based on the data that you provided. We use liked and disliked genres.

## APPLICATION UI

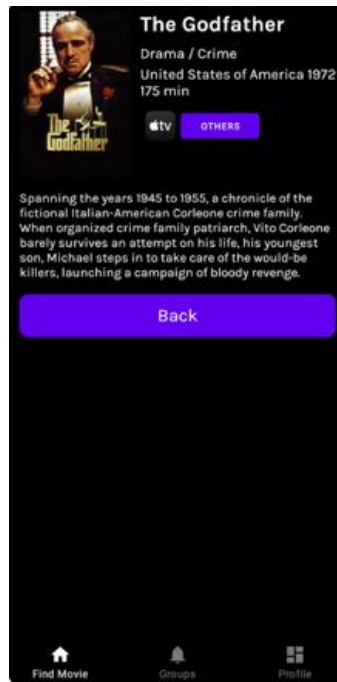
The core of the application is not too sophisticated, but on the other hand, it provides a complex solution to a frequent problem especially for a younger generation. Because of that, one of our biggest goals became to keep the user interface simple, friendly, and straightforward. We discussed a couple of ideas about the simplicity of UI. The best idea

seemed to be to combine the name of the application - Movie Swiper with the Home Page. When a user runs the application for the first time, it is very obvious, that the application expects swipe gestures on offered movies. The main page is represented in Figure 1. It does not provide only the movie name but also some brief information such as IMDb rating, duration, and a short recap about the movie, which we added after some discussion on the project workshop.



Figure 1 - Main Page

Users can also get more information about the offered movie by clicking on the „Details“ button. After that, it shows another screen, where more detailed information is provided. As is represented in Figure 2, it contains information about the genre and release year. Furthermore, it provides information about the availability of the movie on streaming platforms such as Netflix, Apple TV, HBO, and others. Because we use TMDB to get the streaming services, we had to include a link to the website where there is more information. Thanks to that, users can go directly to the platform and stream/rent/buy the movie that is shown. This is an alternative solution. We wanted to make it possible for the user to click the icon and go straight to the streaming service, but due to the license that TMDB uses, we had to do it this way.



*Figure 2 - Movie Details page*

Another part of the application is Groups. Every user in our application can create his groups. For now, the user can specify only the name of the group, it is because of the free hosting version of our used technology - Firebase which will be discussed later, and we did not want to overload the database with unnecessary information such as passwords, or specific permissions inside the groups. Other users can join existing groups with the Join Group form, you just have to know the group name to join. After a successful join, the user can see other members. The main function of our application (to find preferred movies for all members of the group) is available right below this user list and after the user clicks on the „Recommend a Movie“ button, it finds a match based preferences of all the users in the chosen group. After the movie is recommended, the user can get more information about it just like in the main swiping screen of the app that was described before. The group functionality is represented in Figure 3.

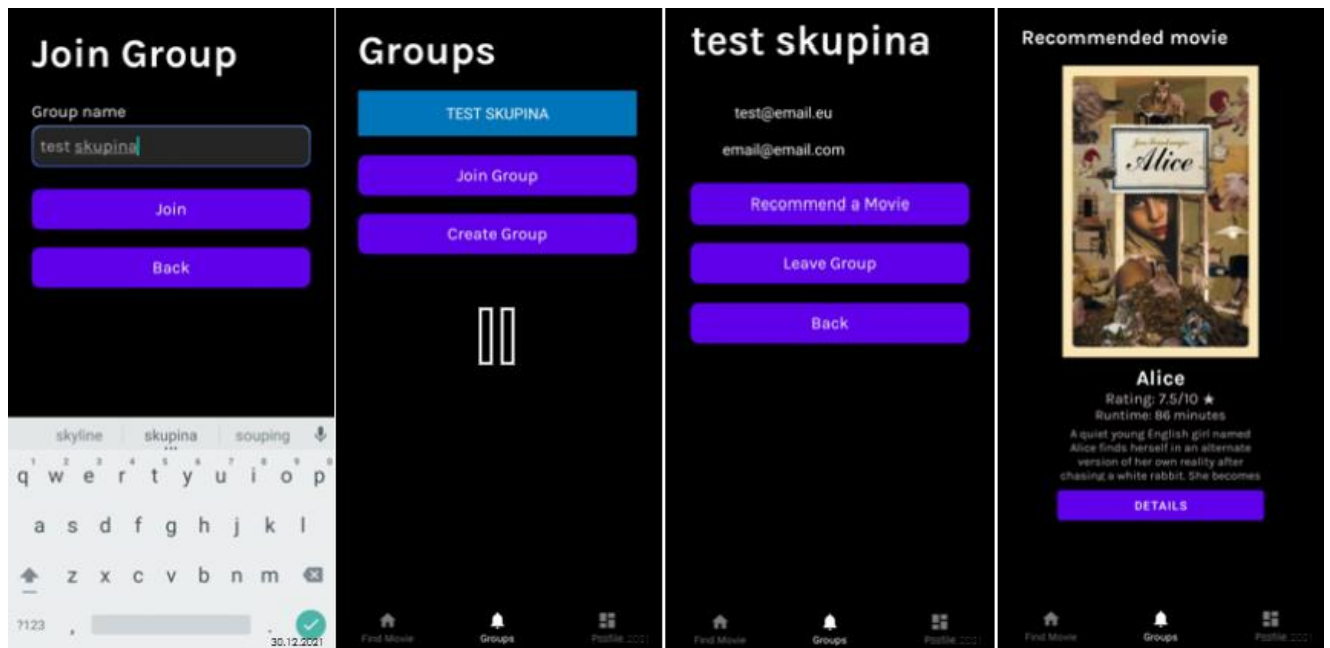


Figure 3 - Groups functionality

## USED TECHNOLOGIES

- Android Studio
- Kotlin
- Firebase
- TMDB

### Android Studio

Android studio was used as IDE because it offers many features for android development but mostly, it contains an android emulator for various types of available androids, so the final version of the application could be tested on different devices. On one hand, it is a powerful tool, but on the other hand, it is extremely slow, especially on less powerful computers. We had a difficult time sometimes when we were modifying code, especially the Kotlin part because it keeps reinstalling the application (in the emulator) and often makes Android Studio unusable because of lagging.

### Kotlin

As a programming language, we decided to go for Kotlin. Same as with Android Studio, we encountered so many difficulties. But we believe, that it was caused mostly by the fact, that no one of us had any experience with development in Kotlin so we probably did a lot of things the wrong way.

### TMDB

TMDB which stands for "The Movie Database" is the best alternative for IMDB API that we found. The reason why we did not use IMDB API which is without any doubt the most popular API is that IMDB is not free to use. After we discovered TMDB it quickly seemed to be the right choice for our project. It is free for our purposes (personal use, not application on the store), TMDB also has a great API that enabled us to extract information for the application easier.

Also, Kotlin has a library for the TMDB so after importing it, it was just about calling the methods from the library and providing arguments for the methods.

## Firestore

Another big part of our project was to create and connect the database with the application. We achieved it by using Firestore, which offers Kotlin SDK. With the help of Firestore, we created our free, but limited database. The database is primarily used for authentication of the user and to store users, and their preferences, or groups created in the application.

## SUMMARY

In summary, we created a user-friendly application. We learned, that it is not too easy to transfer theoretical concepts into a finished application. Overlooked from hard back-end development, we quickly realized the fact, that it is not so easy to create a friendly and straightforward environment and it takes time to find a way, how to keep the interface great in all ways.

We used the information that we learned in the project workshops to improve the application but it is of course not perfect as it is only a prototype after all.

The final application is not yet ready to be published on Google Play. The reasonable fact is that we are using free and limited versions of Firestore, which limits us in space, and a number of requests.