# Movie rating and recommendation app

Eugeni Suaña Barranco 1526274

**Resum**— Aquest projecte neix de la necessitat per part de les persones interessades d'una aplicació totalment enfocada en recomanar i compartir opinions sobre contingut digital (Principalment pel·lícules i sèries). L'objectiu principal d'aquest projecte és el desenvolupament d'una aplicació mòbil per al sistema operatiu Android que permeti crear usuaris, agregar altres usuaris com a contactes i poder compartir opinions, llistes i recomanacions sobre contingut digital. Aquesta aplicació s'ha desenvolupat a través del entorn de programació Android Studio, utilitzant els llenguatges Kotlin i Java, amb el suport de Firebase i seguint la metodologia agile Kanban. El desenvolupament d'aquest sistema ha ocupat un total de 4 mesos, dels quals un mes i mig ha sigut dedicat a la planificació inicial del projecte i la resta al desenvolupament del mateix. En aquest període de temps s'ha pogut desenvolupar una aplicació funcional que compleix amb la majoria dels objectius inicialment proposats, satisfent d'aquesta manera les necessitats dels usuaris involucrats.

**Paraules clau**— Kanban, Java, Kotlin, Firebase, aplicació, contingut digital, sèries, pel·lícules, llistes, opinions, recomanacions, contactes, grups.

**Abstract**— This project was born out of the need from the people interested of an application totally focused on recommending and sharing opinions about digital content (Mainly movies and series). The main goal of this project is the development of a mobile application for the Android operating system that allows the creation of users, adding other users as contacts and being able to share opinions, lists and recommendations about digital content. This application has been developed through the Android Studio programming environment, using the Kotlin and Java languages, with the support of Firebase and following the agile Kanban methodology. The development of this system has occupied a total of 4 months, of which a month and a half has been dedicated to the initial planning of the project and the rest to the development of the same. During this time it has been possible to develop a functional application that meets most of the objectives initially proposed, thus meeting the needs of the users involved.

**Index Terms**— Kanban, Java, Kotlin, Firebase, application, digital content, series, movies, lists, opinions, recommendations, contacts, groups.

## 1 Introduction

NOWADAYS with more and more digital platforms appearing each day, people consume a huge quantity of series and movies. When choosing what to see, most of the time, the doubt appears due to the immense quantity of digital content available, and, after a long time of searching, we end up choosing a movie or series that it is not worth the time. To not end up in this situation, we often rely on asking our friends and family for some advice on what to see, or we ask them if they have recently seen any movie or series that they have liked. The problem is that we usually end up forgetting these recommendations, maybe because we simply forget about them after a while or because our contacts have recommended it through some social network chat, and it gets lost among other messages.

With this project, we want to solve this problem with the creation of an application focalized in recommending series and movies to our contacts. This way, our social relationships will be able to help us choose what to watch, and all these recommendations will be centralized in one application instead of being scattered among chats of different social networks.

In this document the implementation of the project will be exposed and explain. First, the context and main objectives will be presented, as well as the planification and methodology that has been followed during the development. Following this, there will be a detailed explanation of the implementation as well as an overall discussion of the obtained results.


- *E-mail de contacte: eugeni.suana@e-campus.uab.cat*
- *Menció realitzada: Enginyeria del Software*
- *Treball tutoritzat per: Helena Bolta Torrell (Ciéncies de la Computació)*
- *Curs 2021/22*


## 2 Context

There exist different mobile applications for Android that allow you to recommend movies or series to your friends and family, create your own lists and opinions and share them with your contacts.

An example of this could be the IMDb [1] and Letterboxd [2] applications that have a lot of different utilities like adding movies or series to lists, create your own

opinions, or even read news about the upcoming movies and series. The only problem is that these applications do not directly allow you to recommend movies to your contacts and to do so you must rely on other applications or social networks.

Then, we have the Friendspire [3] application. With this application you can recommend movies and series to your contacts as well as create your own lists, opinions and share them. The only problem is that the application is not only focused on movies and series but also in other fields like restaurants and bars.

So, if you are looking for an application to share recommendations, opinions and lists with your friends and family and that it is only focused on movies and series, it does not exist yet, and this is what we are trying to solve by developing this project.

## 3 Objectives

The main objective of this final degree project is to develop an Android application where the users can recommend movies and series to other users. This application will allow the creation of users with whom recommend and share opinions with other users individually or together with the creation of groups. Also, the users will be able to create lists of movies and series (for example a list with the pending movies to watch) as well as rate movies and series and add comments with their own opinions.

In the next table we can see the main objectives and their priority:

| Objective | Priority |
|---|---|
| Develop a functional application in Android that allows the user to recommend series and movies to other users individually or in groups | Critical |
| Allow the users to create lists of movies and series inside the application | Critical |
| Allow the users of the application to rate and create opinions about movies and series | Critical |
| Include in the application a wide catalogue with the maximum number of movies and series premiered to date | Optional |

Table 1: Main project objectives

## 4 Methodology

For this project the chosen methodology is an agile methodology for the advantages that this type of methodologies apport at the time of project development and for the adaptability that these have to the different ways of working. [4] The agile methodology chosen it is the Kanban methodology because it is believed that it is the one that will adapt the best to this project by allowing a visualization of the pending tasks as well as the work done, focusing on a continuous flux of work and not in sprints, not require roles and allowing for a non-change reluctant work planification. [5][6]

In order to follow this methodology, the Trello tool has been used. This tool has been selected because it is a free task tracking tool, simple, visual, easy to learn and to use and that, even having some limited features, its is enough for a personal project and allows to obtain a good organization and planification of the required tasks. [7][8] On Figure 1 we can see the Trello board used during the development of the project. The image was taken at the end of the first codification phase.

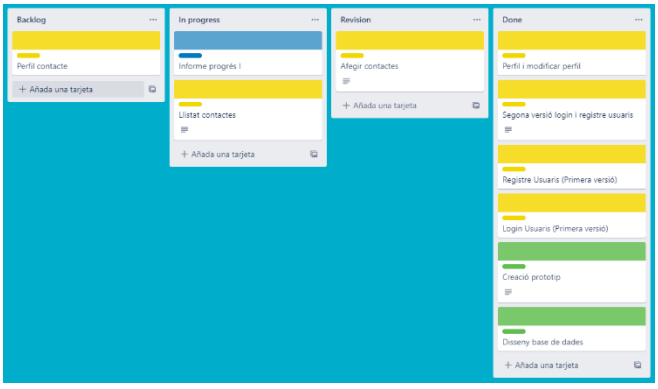



Fig. 1: Trello board

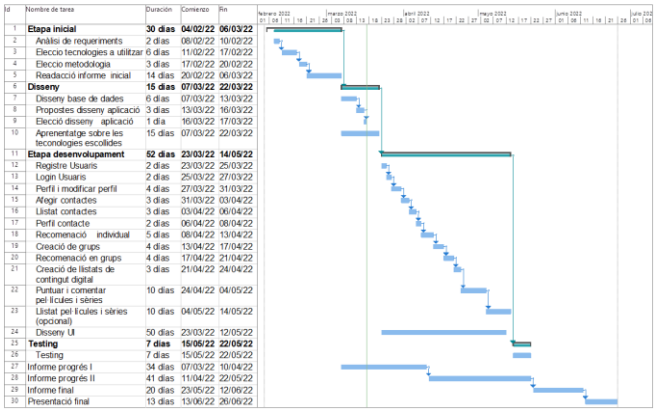
As it can be seen in Figure 3, 4 different columns were created. The first column called “Backlog” is the column where uninitiated tasks were located. Then we have the “In progress” column that had the tasks that were being worked on. Finally, we have the “Revision” column where the finished tasks that needed revision before passing to the “Done” column were placed.
In addition to the columns, the tasks were also sorted by class. This classification was made with the use of colors. Green corresponded to the design, yellow to the development of the application and the blue to the documentation.

As to the version control, GitHub [9] has been used. During the development of the application, the newest versions of the code have been uploaded to this version control system as well as the documentation regarding the project. This way we have a tracing and a record of the changes made to the code and, at the same time, an organization of the documentation. Also, thanks to this, both the code and documentation will be saved and updated on the cloud in case of loss or corruption on the working machine.

## 5 Planification

The planification of the project consists of 4 different phases: Initial phase, Design, Development phase and Testing.

To organize these phases and their tasks, they were grouped on a Gantt diagram (Figure 2) created with the Microsoft Project [10] tool.



| Id | Nombre de tarea | Duración | Comienzo | Fin |
|---|---|---|---|---|
| 1 | **Etapa inicial** | **30 días** | **04/02/22** | **06/03/22** |
| 2 | Anàlisi de requeriments | 2 días | 08/02/22 | 10/02/22 |
| 3 | Eleccio tecnologies a utilitzar | 6 días | 11/02/22 | 17/02/22 |
| 4 | Eleccio metodologia | 3 días | 17/02/22 | 20/02/22 |
| 5 | Readacció informe inicial | 14 días | 20/02/22 | 06/03/22 |
| 6 | **Disseny** | **15 días** | **07/03/22** | **22/03/22** |
| 7 | Disseny base de dades | 6 días | 07/03/22 | 13/03/22 |
| 8 | Propostes disseny aplicació | 3 días | 13/03/22 | 16/03/22 |
| 9 | Elecció disseny aplicació | 1 día | 16/03/22 | 17/03/22 |
| 10 | Aprenentatge sobre les tecnologies escollides | 15 días | 07/03/22 | 22/03/22 |
| 11 | **Etapa desenvolupament** | **52 días** | **23/03/22** | **14/05/22** |
| 12 | Registre Usuaris | 2 días | 23/03/22 | 25/03/22 |
| 13 | Login Usuaris | 2 días | 25/03/22 | 27/03/22 |
| 14 | Perfil i modificar perfil | 4 días | 27/03/22 | 31/03/22 |
| 15 | Afegir contactes | 3 días | 31/03/22 | 03/04/22 |
| 16 | Llistat contactes | 3 días | 03/04/22 | 06/04/22 |
| 17 | Perfil contacte | 2 días | 06/04/22 | 08/04/22 |
| 18 | Recomenació individual | 5 días | 08/04/22 | 13/04/22 |
| 19 | Creació de grups | 4 días | 13/04/22 | 17/04/22 |
| 20 | Recomenació en grups | 4 días | 17/04/22 | 21/04/22 |
| 21 | Creació de llistats de contingut digital | 3 días | 21/04/22 | 24/04/22 |
| 22 | Puntuar i comentar pel lícules i sèries | 10 días | 24/04/22 | 04/05/22 |
| 23 | Llistat pel lícules i sèries (opcional) | 10 días | 04/05/22 | 14/05/22 |
| 24 | Disseny UI | 50 días | 23/03/22 | 12/05/22 |
| 25 | **Testing** | **7 días** | **15/05/22** | **22/05/22** |
| 26 | Testing | 7 días | 15/05/22 | 22/05/22 |
| 27 | Informe progrés I | 34 días | 07/03/22 | 10/04/22 |
| 28 | Informe progrés II | 41 días | 11/04/22 | 22/05/22 |
| 29 | Informe final | 20 días | 23/05/22 | 12/06/22 |
| 30 | Presentació final | 13 días | 13/06/22 | 26/06/22 |

Fig. 2: Gantt diagram of the project

This way, the tasks were organized by phase, time, and dependencies.

The project starts with the Initial phase. This phase begins with the analysis and elicitation of requirements, followed by the choice of the technologies and methodology that will be used during the development, establishing the bases of the project. This phase ends with the drafting of the initial report where the objectives that have been discussed with the people interested in the project are also established.

Next, we have the second phase where the overall design of the application is decided. In this phase, the database design is generated as well as some proposals for the application design, one of which is chosen as the final design. Also, during this phase, an investigation of the elected technologies will be followed through documentation and tutorials.

Following this, we have the largest phase of the project known as the Development phase. Here, as the name says, the development of the application takes place. This phase is extensive, and it is divided by tasks regarding the different application functionalities.

Finally, we have a testing phase where the developed functionalities will be tested to see if they work correctly and as expected.

Meanwhile, apart from the Initial phase, the project documentation will be carried out in parallel to the other phases.

## 6 ELICITATION OF REQUIREMENTS

In this section of the document the elicitation of requirements will be exposed. This is one of the cornerstones of every project, having a major impact on the design and posterior phases of development and helping with the communication between the client and the users and understand their needs.

This analysis of requirements was done before the start of the development based on the objectives of the project agreed at the Initial phase.

### 6.1 Functional requirements

In the next table (Table 2), the functional requirements of the project are presented. These requirements define the functionality that the developed system has to offer.

| ID | Requirement |
|---|---|
| RF-01 | The application must allow the registration of users |
| RF-02 | Registered users must be able to log in and log out of the application |
| RF-03 | The user must be able to visit its own profile as well as that of their contacts |
| RF-04 | The user must be able to modify its own profile |
| RF-05 | The application must allow other users to be added or removed as contacts |
| RF-06 | By visiting the profile of its contacts, the user should be able to view the public lists and opinions created by them |
| RF-07 | The user must be able to create, modify, and delete public or private lists of digital content |
| RF-08 | The user must be able to create, modify, and delete public or private opinions about digital content |
| RF-09 | The application should allow the user to create groups of contacts |
| RF-10 | The group administrator should be able to edit the group |
| RF-11 | Users should be able to recommend digital content to their contacts or contact groups along with a descriptive message |
| RF-12 | The user should be able to send a recommendation to more than one contact simultaneously |
| RF-13 | The application must allow the user to see all contacts public lists on a single screen |
| RF-14 | The application must allow the user to see all contacts public opinions on a single screen |
| RF-15 | The user must be able to delete the recommendation messages sent |

Table 2: Project's functional requirements

### 6.2 Non-functional requirements

In Table 3, the non-functional requirements are presented. These requirements define the wished qualities of the developed system, influence the design more than the functional requirements but not determine the functionality of the application.

| ID | Requirement |
|---|---|
| RNF-01 | The application will work on the Android operating system |

| | |
|---|---|
| RNF-02 | In order to access the application, the user must be registered |
| RNF-03 | Application screens need to be adapted to different resolutions |
| RNF-04 | Switching time between screens should never exceed 2 or 3 seconds |
| RNF-05 | The application must have a user learning time of less than 4 hours |
| RNF-06 | The application must provide informative and user-oriented error messages |
| RNF-07 | It may take no more than 2 seconds to update the data in the database |

Table 3: Project's non-functional requirements

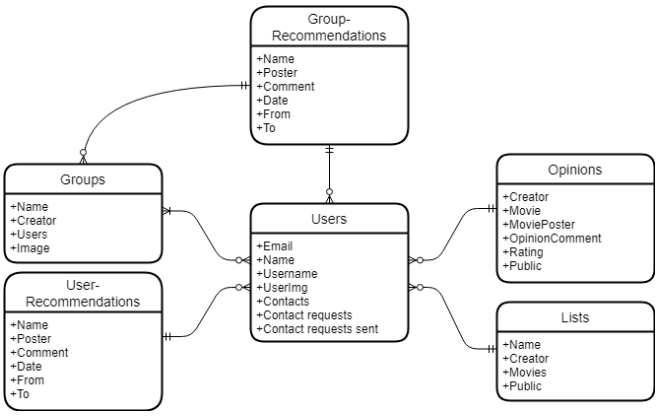



Fig. 3: Database diagram of the project

## 7 SYSTEM DESIGN

In the following section, the system design is shown through some diagrams that will allow a better understanding of the system's behaviour as well as the user's possible actions. Also, a prototype created before the development of the application will be presented.
Here is where, having the bases of the project stablished with the agreement of the objectives and the elicitation and analysis of requirements, the Design phase started.

### 7.1 Use case diagram

On the Annex 1, the use case diagram of the system generated with the PlantText [11] tool is presented. This diagram shows the main functionalities that the user has access to and how they interact with each other.

### 7.2 Database diagram

The database of the project is in charge of saving the data referring to the users as well as their opinions and lists of movies and series. Also, the recommendations and groups created by the different users are stored in the database.

At the start of the project, an initial database diagram was presented using the AppDiagrams [12] tool, but, after some iterations it was decided that the best design for the database was the one that can be seen on Figure 3.

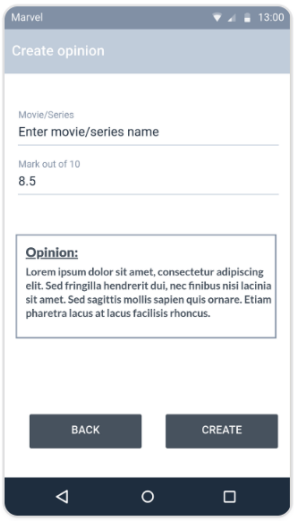
As it can be seen on the diagram, two different types of recommendations tables have been stored on the database. This has been done because it simplifies the job of getting the different recommendations for both individual users and groups.

### 7.3 System prototype

In the next section, some of the key screens and functionalities of the application will be presented as part of the prototype generated with the MarvelApp [13] tool.
This prototype was created to try to simulate the behaviour of the application so that the people interested on the project could perform some usability tests and give feedback to the development team. Also, this prototype helped to establish, more or less, the overall design of the application and the screens that would be necessary to generate the application.

Even though on the planification it was stated that after the designing the database diagram the development team would proceed to create different design proposals, it was finally decided to directly generate a prototype of the application. This option was chosen because it was thought that with this prototype the application design would already be established without needing to implement any sketch boards.

On the next figures (Figure 4 and Figure 5) we can see the prototype screens for the Create opinion and Create list functionalities.



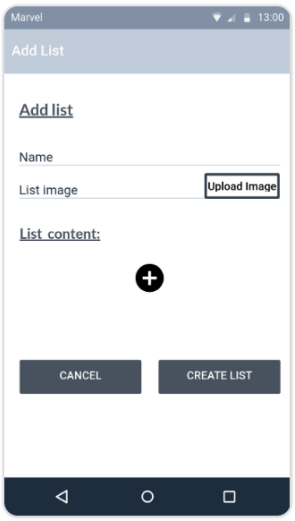




Fig. 4 and 5: Prototype screens for Create opinion and Create list

Next, on Figure 6 and Figure 7 the prototype screens for the Mange opinions and Manage lists functionalities are presented.

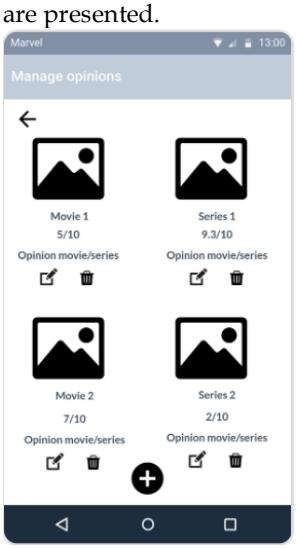
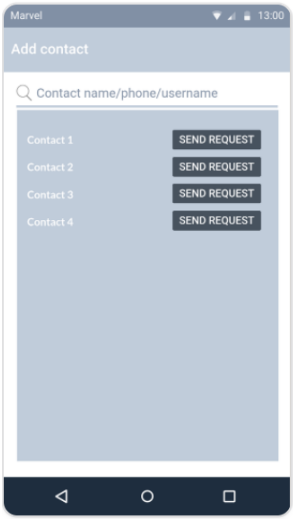




Fig. 6 and 7: Prototype screens for Manage opinions and Manage lists

On Figure 8 and Figure 9 we can see the prototype screens for the functionalities Add contact and Create group.



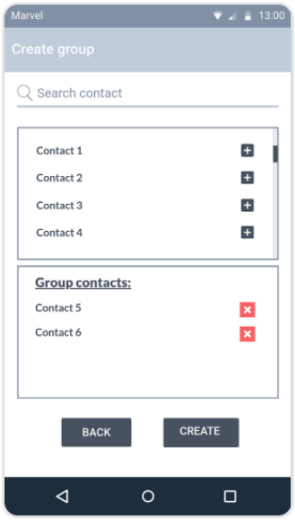
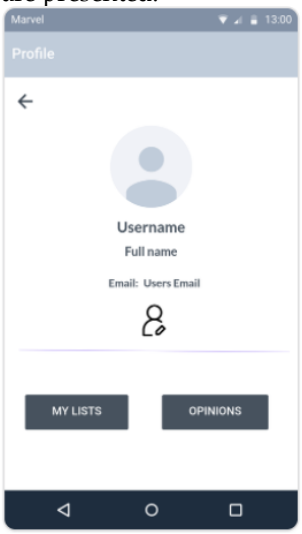




Fig. 8 and 9: Prototype screens for Add contacts and Create group

Finally, on Figure 10 and Figure 11 the prototype screens for User's profile and Contact recommendation are presented.



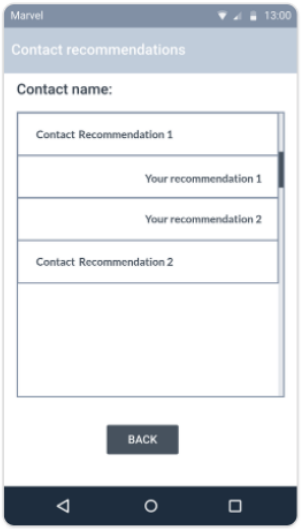




Fig. 10 and 11: Prototype screens for User's profile and Contact recommendation

As it will be seen later in the document, some of the application final screens maintained the style and functionality of the prototype screens, while others are completely different. This is because at the time of developing the functionalities, it was chosen an easier or better way of implementing the same screens while maintaining the overall style of the application.

## 8 DEVELOPMENT PHASE

In this section of the document, the largest and most important phase of the project will be presented. Here, the main technologies chosen for the project will be discussed as well as the procedure to develop the most im-

portant functionalities of the system.

### 8.1 Technologies

At the time of deciding which technologies to use, initially, it was thought to develop the application through ReactJS. This decision was rapidly changed when a lot of information was found throughout the internet about the major advantages granted by Android Studio when developing applications for the Android OS. This is why it was finally decided that the application would be generated with the Android Studio environment and the Kotlin and Java languages. [14][15]

For the backend, since the initial phases of the project, it was very clear that Firebase from Google would be used. It was decided to use this tool because it has a lot of services that grant great advantages while developing an application. These advantages go from the ease of implementing a secure user authentication system to allowing to connect our application with the database without needing to use an API. Moreover, Android Studio and Firebase complement each other very well because Android Studio comes with built-in functionality of allowing to easily connect your project to Firebase, this way being able to access all the advantages that this tool provides in a quite simple way. This factor also influenced the choice of Android Studio over ReactJS. [16]

The only problem that was found with Firebase was the fact that it is a free tool only until certain point. If the scale of the application ended up rising to high levels (more than 10.000 monthly user verifications or more than 20.000 writing operations per day), it would probably be needed to pay in order to keep using the Firebase services. [17]

### 8.2 Creation of the project

The first step on this development phase was the creation of the project on Android Studio and bind it with Firebase. To do this, the instructions on the Google Firebase official website [18] where followed.

Once the project was connected with Firebase, in order to access the functionalities that this tool offers, it was simply needed to add the desired dependencies to the Gradle file. This way, again following the instructions on the Google Firebase official website, the needed dependencies in order to use the data base and the Firebase user authentication system were added.

Having now access to the Firebase tools, the creation of the application's different functionalities and screens started. It needs to be mentioned that it was hard to start programming because never before it had been worked with Kotlin, but with previous knowledge of Java, following some informative tutorials via YouTube [19][20], and with the help of StackOverflow [21], it was possible to start developing the first screens effectively.

### 8.3 Log in and Sign up

Firstly, the user Log in and Sign up screens where programmed. Initially, it was not easy to implement these screens because of the reasons mentioned previously, but, once obtained knowledge of the Kotlin language and the Android Studio programming environment, this two functionalities where quickly and effectively implemented (Figures 12 and 13). This way, new users registered in the app were generated and stored correctly in the database.

The Firebase Authentication tool must really be taken into account for this task. This tool made it very easy and timesaving to register and authenticate users to our application.

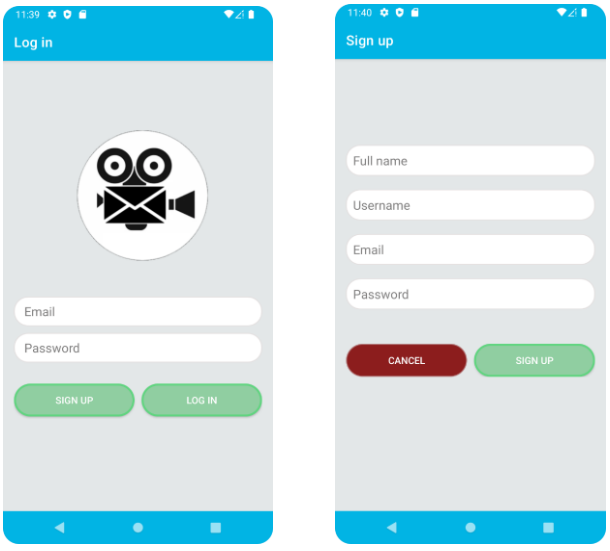



Fig. 12 and 13: Log in and Sign up screens

### 8.4 Chat with contact

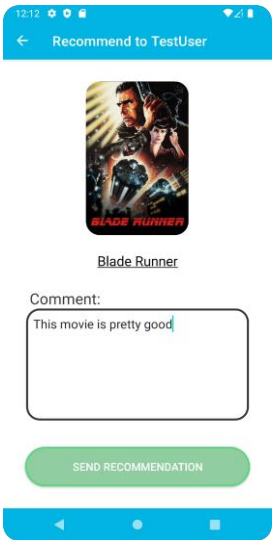
The first main functionality of the application that was developed was the one to allow recommendations between users.

Since the beginning, it was clear that the recommendations between users would be shown in the application as a chat, so the first step was to create this chat screen.

In order to do this the tutorials made by the YouTube channel "Lets Build That App" [22] were followed. Thanks to this YouTube channel, a lot of knowledge of Android Studio and Kotlin was obtained as well as the importance of using libraries that highly simplify the amount of work.

The most important libraries discovered through this channel were the Picasso library [23], which greatly simplifies the code when displaying images on the screen, and the Groupie library [24], which is the most important library of the project, as it makes it extremely easy to display lists on the screen and has been used in many of the functionalities of the application (user list, chats, movie lists, opinions lists...).

The final result of the create recommendation and chat screens can be seen on figures 14 and 15 respectively.



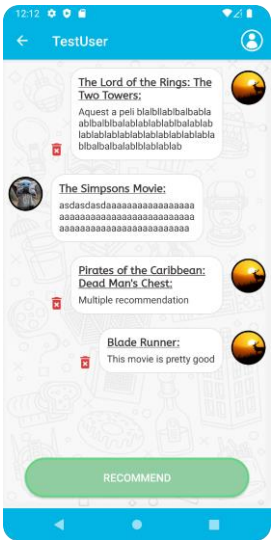




Fig. 14 and 15: Create recommendation and chat screens

### 8.5 Implementation of the API

In the middle of the application development, during a reunion with the tutor, it was decided that the application without a search engine of digital content would be too simple and would not meet the standards. Also, the recommendation functionality allowed the user to manually enter the name of the movie or series that he or she wanted to recommend, which did not really make sense.
For these two reasons it was decided to implement a search engine of digital content using an API.

After this reunion, APIs that provided listings of digital content were searched and it was seen that there weren't many and they were hard to find. The RapidAPI website [25] was accessed and there several APIs were discovered, but none came close to what it was wanted. In addition, many of these APIs needed to be paid in order to be used or only consisted of one type of digital content, whether they were just movies or just series.

After some more searching, the API for TheMovieDatabase website [26] was found. This API was perfectly suited because it had all kinds of digital content. In order to access this API an account was created on the website and a request to use the API was sent. Once the permission to use the API was received and having access to all the data from TheMovieDatabase through a private key, the only thing left was to find a way to pass this data to the application.

Searching for information on the Internet it was found that the best way to retrieve API data in Kotlin was through the Okhttp3 [27] and Gson [28] libraries. The request to the API is made with Okhttp from which a JSON is received with the results. This JSON is transformed with the Gson library so that the information it carries can be read within the application.
When a movie or series was searched, the API returned a lot of information. This is why when picking up the information, apart from the title and poster, the popularity was also picked up so the results could be filtered and only show the most popular.

Once the functionality for receiving the information from the API was implemented and worked properly, the screen that would be used to search for movies and series was created.
As it can be seen on Figure 16, the screen has a search bar where the user can search for a title and the results are printed underneath.


Fig. 16: Digital content searcher screen

The user could now access an extensive catalog of digital content inside the application. This search engine screen was added to the recommendation functionality. This way, the user could no longer invent a title to recommend but search up for the digital content and select it.

### 8.6 Digital content lists

Having now implemented the API, the next step was to develop another of the application's key functionalities: The creation of digital content lists.
With this feature, the user should be able to create custom lists of digital content for personal use or to share with other users.

When generating a list, the first step would be to choose the name of the list and whether you wanted it to be public or not. Next, the user, through the digital content search engine, (Figure 16), would choose the first digital content item to be included in the list. Done this, the list would be generated with this first element.
Once created the list, the user would be able to choose what to do with it, whether to edit it, rename it, change its privacy, add more digital content to it or simply delete it.

Thanks to what was learned developing previous functionalities and the use of the Groupie and Picasso libraries, it was possible to implement these features without major complications.

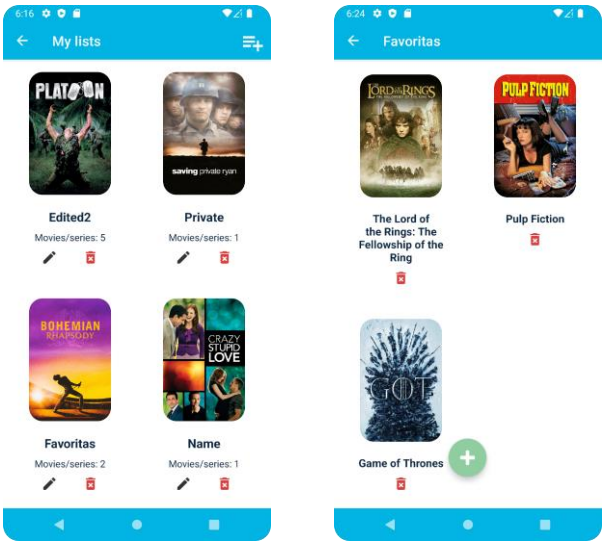



Fig. 17 and 18: Screen with user's lists and screen with the content of a list

Now that the user could create its own content lists, the screens necessary for users to be able to access the public lists of their contacts were generated. This functionality was quickly implemented, as the code is very similar to the one used to display the user's own lists.

### 8.7 Rate and comment digital content

The last major functionality to implement was the user's ability to create opinions about digital content.
It was agreed with the people interested in the project that the user could choose a score from 1 to 10, without decimals, to rate digital content and could also add a comment to the opinion if wished. With this in mind, the necessary screens were created to implement this new functionality.

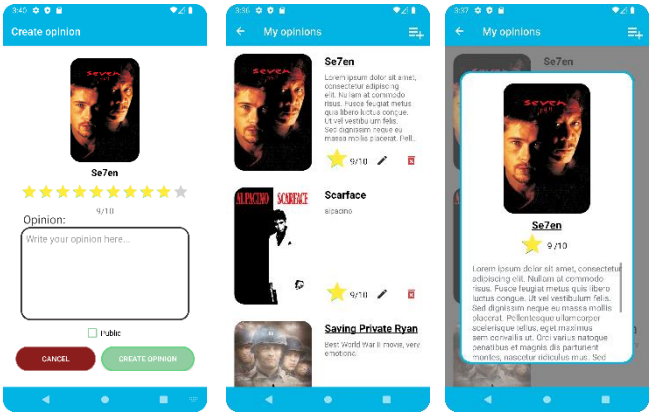



Fig. 19, 20 and 21: Create an opinion, list of opinions and opinion pop-up screens

When the user decides to create a new movie or series review, the first screen that he or she would encounter is, again, the digital content searcher screen (Figure 16). On this screen the user should select a movie or series. Once selected, the user would go to the Create opinion screen (Figure 19), where he or she could choose the rating through a star bar, add a comment to the review if wished and select if the review is wanted to be public or not.
This created opinion would then be added to the list of opinions (Figure 20) where the user could edit it, changing the score, comment, or privacy, or could simply delete it. Also, in this same screen (Figure 20) the user could select one of the opinions to see it in more detail through a Pop Up as seen in Figure 21. Thus, if the comment of opinion was too long and could not fit on the opinions list screen, the user could see the full comment in a scrolled text box. Seeing the result of implementing these Pop Ups, they were also implemented on the chat screen, both for groups and contacts. This way, when a user clicks on a recommendation, a Pop Up appears with the movie poster and title.

As with digital content lists, now that the user was able to create opinions, it was necessary to generate the screens so that users could access the public opinions of their contacts. As with the lists, this functionality was quickly implemented, because the code is very similar to the one used to show the user's own opinions.

## 9 TESTING

### 9.1 Exploratory testing

In exploratory testing test cases are generated and executed to simultaneously learn about the system while testing its main functionalities. This type of testing is really powerful because it allows to freely test the application, identify, document and discover errors that otherwise would not be found, investigate its causes and correct them with the objective of finding related errors.

All the tests were performed on Android emulators through Android Studio and with my own mobile phone (Samsung Galaxy A8).

On annex A2 some examples of the generated test cases can be found.

### 9.2 User testing

The user testing is the process through which real users, without many directions, perform specific tasks to test the interface and functionalities of the application. This way, the usability of the application is evaluated, and feedback is received.

To perform this type of testing, some people interested in the project were needed to perform some simple tasks inside the application. In my case, my family and friends served as testers.

Some of the results obtained during these tests

where:

- **Design:** A lot of changes were implemented to the design (colours, button shapes, background images, …) thanks to feedback received from the users while testing.
- **Delete message button:** Some of the users suggested that a button to delete the messages on the chat screen would be needed in case an error was committed while sending a recommendation. That is why a little button was included with every recommendation so that the message could be deleted if needed.
- **Public or private option:** While creating a list or opinion, the majority of testers suggested to include the possibility of choosing whether it to be public or not. To satisfy this need, a checkbox was included both while creating and editing a list or opinion to be able to choose if you wanted it to be public or not.

## 10 RESULTS

Right now, the project meets with the objectives and most of the functional requirements specified in sections 3 and 6.

The application allows the creation of users and the possibility of adding and removing other users from the circle of contacts. Through the application, the users can access a list with all their contacts and send digital content recommendations to them.

Moreover, the users can create groups of contacts if they wish so. This way they can send and receive recommendations to more than one user at the same time. This groups can be edited by the group administrator if needed. The administrator can change the group image, name, add and remove group members or even pass the administrator role to another group member.

At the same time, users can generate their own lists of digital content. These lists can be private or public, include both movies and series items, and can have as much items as wanted.

Also, the system offers the possibility of generating your own public or private opinions by rating and commenting digital content. The user can choose a rating between 1 to 10 and add a comment if wished.

Both opinions and lists can be edited or deleted.

The user has access to an extensive list of digital content through the application thanks to TheMovieDatabase API. This content can be recommended to contacts, added to lists and rated and commented with the creation of opinions.

Apart from the main functionalities, other smaller but necessary features have been developed. Examples of this could be, the possibility of visiting your own profile and editing it, visiting your contacts profiles, sending the same recommendation to multiple contacts without the need of a group or visiting all the opinions and lists of your contacts at the same time without needing to enter inside their profiles.

It is also important to remark as a result the experience obtained by developing this project. Initially I had no experience with Android Studio or Kotlin. Through documentation, videos, guides and coding, knowledge about this environment and language has been obtained. As more experience was obtained, complex functionalities could be developed faster and in a more agile way without many problems, errors or bugs that could delay the project.

Simultaneously, generating this application, administrating the time available, the different phases of the project, having to document the development, … All of this has helped to acknowledge how to organize a project at the time of development.

## 11 CONCLUSIONS

The main objective of the project was the generation of an application that allowed the users to send recommendations to their contacts, create digital content lists as well as opinions about digital content. A first version of the application has been developed that fulfils these objectives with the help of TheMovieDatabase API.

Overall, the project has gone much better than what was expected initially. As mentioned other times on the documentation, at the begging of the project I had very low close to no experience with Android Studio and Kotlin and I was quite anxious of not achieving the objectives initially proposed.

It is necessary to say that the same functionalities could have been developed faster and without that many problems if I initially had had more experience with the program and language. However, I also think that the whole learning process is precisely one of the key factors of the Final Degree Project and should not be overlooked.

Another conclusion that could be drawn from the project, would be the realization of the importance of establishing the project basis from the begging as well as a clear work scheme. This way, the objectives of the project are clear from the beginning, avoiding doubts during the implementation of the application that could delay the development.

### 11.1 Future lines of the project

Even though the application meets the objectives initially proposed, the project is far from over. One of the conclusions that has been drawn while developing the application is that there is always room for improvement.

Some of the application functionalities could be better adapted to the user, simplifying the application's use. For example, the addition of a search bar for the contacts while creating the groups, adding support for different languages, adding a search bar for the lists and opinions or improving the design and screen composition.

The next main functionality that would be worked on if the project followed its development would be the addition of the phone number while registering new users. This way the users could more easily add as contacts

inside the application their mobile phone contacts.

Even if the application can still be improved, the overall feedback from the people interested in the project is positive. If these people finally like how the project has resulted, talks could be made to upload the application to the Google Play Store so that the people interested could download and use it. Only as a supposition, and as mentioned in the Technologies section (8.1), if the application ended up having a lot of users, Firebase maybe should be replaced by another type of database or pay the price to use Firebase.

## ACKNOWLEDGMENTS

First of all, I would like to thank my tutor Helena for all the support she has given me during the development of this project. She has always been available, has been close to me, has given me good advice and has known how to motivate me at all times.

Secondly, I would like to thank my family who have always been by my side during this project, supported me when I needed it most and have served as test users whenever they were asked for.

Lastly, thanks to my friends as well. Some have helped me to come up with ideas and solutions to certain problems that have arisen. Others have been my escape route when I needed to be distracted from the project. Thank you all so much for enduring my endless complaints during the development of the work.

## BIBLIOGRAPHY

[1] IMDB, "IMDB Cine & TV", https://play.google.com/store/apps/details?id=com.imdb.mobile&hl=es&gl=US [Accessed: 20-02-2022]

[2] Letterboxd, "Letterboxd", https://play.google.com/store/apps/details?id=com.letterboxd.letterboxd&hl=es&gl=US [Accessed: 20-02-2022]

[3] Friendspire, "Friendspire: Movies, TV, Books" , https://play.google.com/store/apps/details?id=com.friendspire&hl=es_419&gl=US [Accessed: 20-02-2022]

[4] John C. Goodpasture, "Project Management the Agile Way, Second Edition", http://projanco.com/Project%20Management%20the%20Agile%20Way%20Making%20It%20Work%20in%20the%20Enterprise,%202nd%20Edition.pdf [Accessed: 21-02-2022]

[5] Dan Radigan, "Kanban –A brief introduction", https://www.atlassian.com/agile/kanban [Accessed: 21-02-2022]

[6] Ana Lamelas, "Top 5 main Agile methodologies: advantages and disadvantages" https://www.xpand-it.com/blog/top-5-agile-methodologies/ [Accessed: 21-02-2022]

[7] Trello, "Trello para la productividad personal", https://trello.com/es/teams/personalproductivity [Accessed: 21-02-2022]

[8] Teamhood, "21 Best Kanban Board Software of 2022", https://teamhood.com/kanban/best-kanban-board-tools/ [Accessed: 21-02-2022]

[9] GitHub, "Github – Code Repository", https://github.com/ [Accessed: 23-03-2022]

[10] Microsoft Project, "Software de administración de proyectos", https://www.microsoft.com/es-ww/microsoft-365/project/project-management-software?market=bz [Accessed: 21-03-2022]

[11] PlantText, "The expert's design tool", https://www.planttext.com/ [Accessed: 10-02-2022]

[12] App Diagrams, https://app.diagrams.net/ [Accessed: 09-03-2022]

[13] Marvelapp, "Rapid prototyping, testing and handoff for modern design teams", https://marvelapp.com/ [Accessed: 16-03-2022]

[14] Platzi (2018) "Cómo crear tu primera app en Android y iOS". [online video], https://youtu.be/6MfO1uU8Avk [Accessed: 12-02-2022]

[15] Quora, "Should I go with React Native or Java for Android app development?", https://www.quora.com/Should-I-go-with-React-Native-or-Java-for-Android-app-development [Accessed: 14-02-2022]

[16] [4] Firebase,"Documentación", https://firebase.google.com/docs [Accessed: 15-02-2022]

[17] Firebase, "Comienza a usarlo sin costo y luego usa un plan prepagado", https://firebase.google.com/pricing?hl=es-419 [Accessed: 15-02-2022]

[18] Firebase, "Agrega Firebase al proyecto de Android", https://firebase.google.com/docs/android/setup [Accessed: 18-03-2022]

[19] Youtube, https://www.youtube.com [Accessed: 04-02-2022]

[20] Youtube, "MoureDev by Brais Moure", https://www.youtube.com/channel/UCxPD7bsocoAMq8Dj18kmGyQ [Accessed: 15-02-2022]

[21] Stackoverflow, "A public platform building the definitive collection of coding questions & answers", https://stackoverflow.com/ [Accessed: 15-03-2022]

[22] Youtube, "Lets Build That App", https://www.youtube.com/c/LetsBuildThatApp [Accessed: 15-02-2022]

[23] Picasso, https://github.com/square/picasso [Accessed: 03-04-2022]

[24] Groupie, https://github.com/lisawray/groupie [Accessed: 03-04-2022]

[25] RapidAPI, "Discover and connect to thousands of APIs", https://rapidapi.com/hub [Accessed: 18-03-2022]

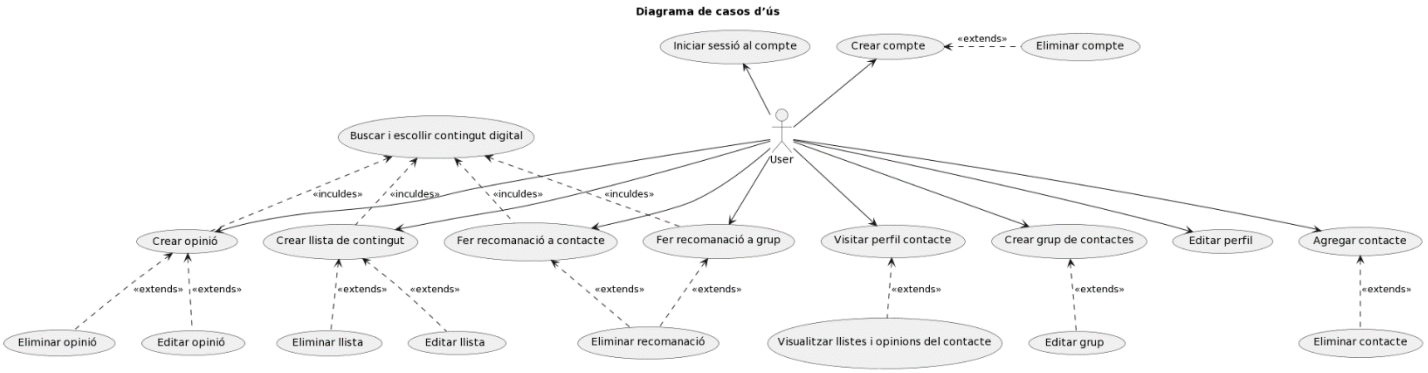
[26] The Movie Database, "Millones de películas, programas de televisión y personas por descubrir. Explora ahora", https://www.themoviedb.org/?language=es-ES [Accessed: 19-04-2022]

[27] Okhttp3, https://github.com/square/okhttp [Accessed: 19-02-2022]

[28] Gson, https://github.com/google/gson [Accessed: 19-04-2022]

# APPENDIX

## A1. Use case diagram




## A2. TEST CASES EXAMPLES

| Test Case ID | Description | Precondition | Assumption | Test Data | Expected result | Obtained result | Status (pass/fail) |
|---|---|---|---|---|---|---|---|
| TC1 | The system must generate a user when the data entered at the time of Sign Up is correct | The user must have e-mail | The application supports gmail.com | - Full name: "Test User"<br>- Username: "TestUser"<br>- Email: test@gmail.com<br>- Password: "123456" | The user is successfully generated in the database | The user is successfully generated in the database | Pass |
| TC4 | The system should not generate a user when the "Email" field is empty | | | - Full name: "Test User"<br>- Username: "TestUser"<br>- Email: ""<br>- Password: "123456" | The user is not generated in the database | The user is not generated in the database | Pass |
| TC5 | The system should not generate a user when the "Email" field is not email formatted | | | - Full name: "Test User"<br>- Username: "TestUser"<br>- Email: testgmail.com<br>- Password: "123456" | The user is not generated in the database | The user is not generated in the database | Pass |
| TC9 | The system should not generate a user when the "email" field matches that of another user | | | - Full name: "Test User2"<br>- Username: "TestUser2"<br>- Email: test@gmail.com<br>- Password: "123456" | The user is not generated in the database | The user is not generated in the database | Pass |
| TC10 | The system must not allow you to log in if the email you entered does not match any user | | | - Email: testX@gmail.com<br>- Password: "123456" | The system does not allow you to log in | The system does not allow you to log in | Pass |
| TC12 | The system must not allow you to log in if the password entered does not match the email | | | - Email: test@gmail.com<br>- Password: "78910" | The system does not allow you to log in | The system does not allow you to log in | Pass |
| TC18 | The user must not be able to edit the profile if the "Username" field is empty | The user must be logged in | | - Full name: "Test User"<br>- Username: "" | The user is not modified in the database | The user is not modified in the database | Pass |
| TC22 | The user must be able to create a digital content list if the "List name" field is not empty | The user must be logged in | | - List name: "TestList" | The system goes to the next screen of the list creation | The system goes to the next screen on list creation | Pass |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| TC23 | The digital content that the user chooses is successfully added to the list of digital content | - The user must be logged in<br>- The user must have a list created | | - List: "TestList"<br>- Movie: "Scarface" | The digital content is added to the list | The digital content is added to the list | Pass |
| TC29 | The user must be able to create an opinion on digital content | The user must be logged in | | - Movie: "Avatar"<br>- Rating: "8/10"<br>-Opinion: "Test test" | Opinion is generated correctly | Opinion is generated correctly | Pass |
| TC31 | The user must be able to edit an opinion on digital content | - The user must be logged in<br>- The user must have an opinion created | | - Opinion: "Scarface" | Opinion is edited correctly | Opinion is edited correctly | Pass |
| TC32 | The user must be able to send a contact request to another user | The user must be logged in | - There is more than one user in the application | - Request to: "Full name" | The request is sent successfully | The request is sent successfully | Pass |
| TC41 | If the user chooses any type of digital content when recommending to a group, the system should go to the next screen of the recommendation generation | - The user must be logged in<br>- The user must have at least one contact<br>- The user must be inside one group | | - Group: "Test group"<br>- Series: "S.W.A.T" | The system moves to the next recommendation generation screen | The system moves to the next recommendation generation screen | Pass |
| TC42 | If the recommendation comment to a contact is empty, the recommendation cannot be sent | - The user must be logged in<br>- The user must have at least one contact | | - Contact: "Full name"<br>- Series: "S.W.A.T"<br>- Comment: "" | The system does not generate or send the recommendation | The system does not generate or send the recommendation | Pass |
| TC45 | If there is a comment on the recommendation to a group, the user should be able to send the recommendation | - The user must be logged in<br>- The user must have at least one contact<br>- The user must be inside one group | | - Grup: "Test group"<br>- Sèrie: "S.W.A.T""<br>- Comentari: "Test comment" | The system generates and sends the recommendation | The system generates and sends the recommendation | Pass |
| TC46 | The system must allow the user to visit a contact's profile by clicking on the profile icon | - The user must be logged in<br>- The user must have at least one contact | | - Contact: "Full name" | The system takes you to the contact profile screen | The system takes you to the contact profile screen | Pass |
| TC47 | Within a contact's profile, the user must be able to visit their digital content lists | - The user must be logged in<br>- The user must have at | | - Contact: "Full name"<br>- Lists: "Contact list, Contact list test, list 2" | The system takes you to the contact lists screen and loads them correctly | The system takes you to the contact lists screen and loads them correctly | Pass |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | least one contact | | | | | |
| TC49 | Within the profile of a contact, the user must be able to visit the opinions on digital content of the contact | - The user must be logged in<br>- The user must have at least one contact | | - Contact: "Full name"<br>- Opinions: "Star Wars V, Game of Thrones" | The system takes you to the contact opinions screen and loads them correctly | The system takes you to the contact opinions screen and loads them correctly | Pass |