



UNIVERSITÀ
DEGLI STUDI
DE L'AQUILA



DISIM
Dipartimento di Ingegneria
e Scienze dell'Informazione
e Matematica

Project Documentation

2021/2022

Course: **Service Oriented Software Engineering**

University of L'Aquila



Alessandro Di Matteo

alessandro.dimatteo1@student.univaq.it

Federico Di Menna

federico.dimenna@student.univaq.it

Gianluca Rea

gianluca.rea@student.univaq.it

Scholastique Uwizeyimana

scholastique.uwizeyimana@student.univaq.it

Summary

Domain	2
Requirements	3
Functional Requirements	3
Non Functional Requirements	3
Architecture	4
Component Diagram	4
Sequence Diagrams	4
Services Description	5
Development Details	5
Project Demo	5
API Documentation	6
Used Technologies	6
Conclusions	6
Meeting the project requirements	6
Meeting the final test specification	6

Domain

The system that we realized is aimed at managing user reviews and rankings on movies and TV series. This way users can share their ratings and opinion about the movies. The application will have a dual purpose: the first purpose is to be able to consult the existing opinions of users thus understanding which film could be more suitable for the user, the second purpose is to be able to share their experience in order to actively participate in the community.

At this point it is useful to clearly specify which are the two types of feedback that the system will grant to the user:

- **Review:** review/*textual comment* of the user: the user can write in a textbox what he thinks about the film
- **Rating / Ranking:** like/dislike and ranking based on different parameters.

Requirements

Functional Requirements

#	Functional Requirement	Priority
1	Authentication (login and registration): Since the application allows the user to publicly release reviews and ratings, the user must be registered and logged into the system. For this reason, it is appropriate to have the login and registration functionalities inside the system..	MEDIUM
2	Search the film: show the list of the films based on selected filters by the client	HIGH
3	Show all the details about a film/tv series (including ranking and reviews of the other users)	HIGH
4	Add reviews and ratings of the film/series tv. Rating parameters will be: A. Film direction; B. Actors; C. Global score; D. Dialogues; E. Costumes;	HIGH
5	System must calculate a global score through a heuristic/algorithm/linear combination (0-100) for the film. (Note: The heuristic should be able to update each time without considering all the previous reviews individually but incrementally)	LOW

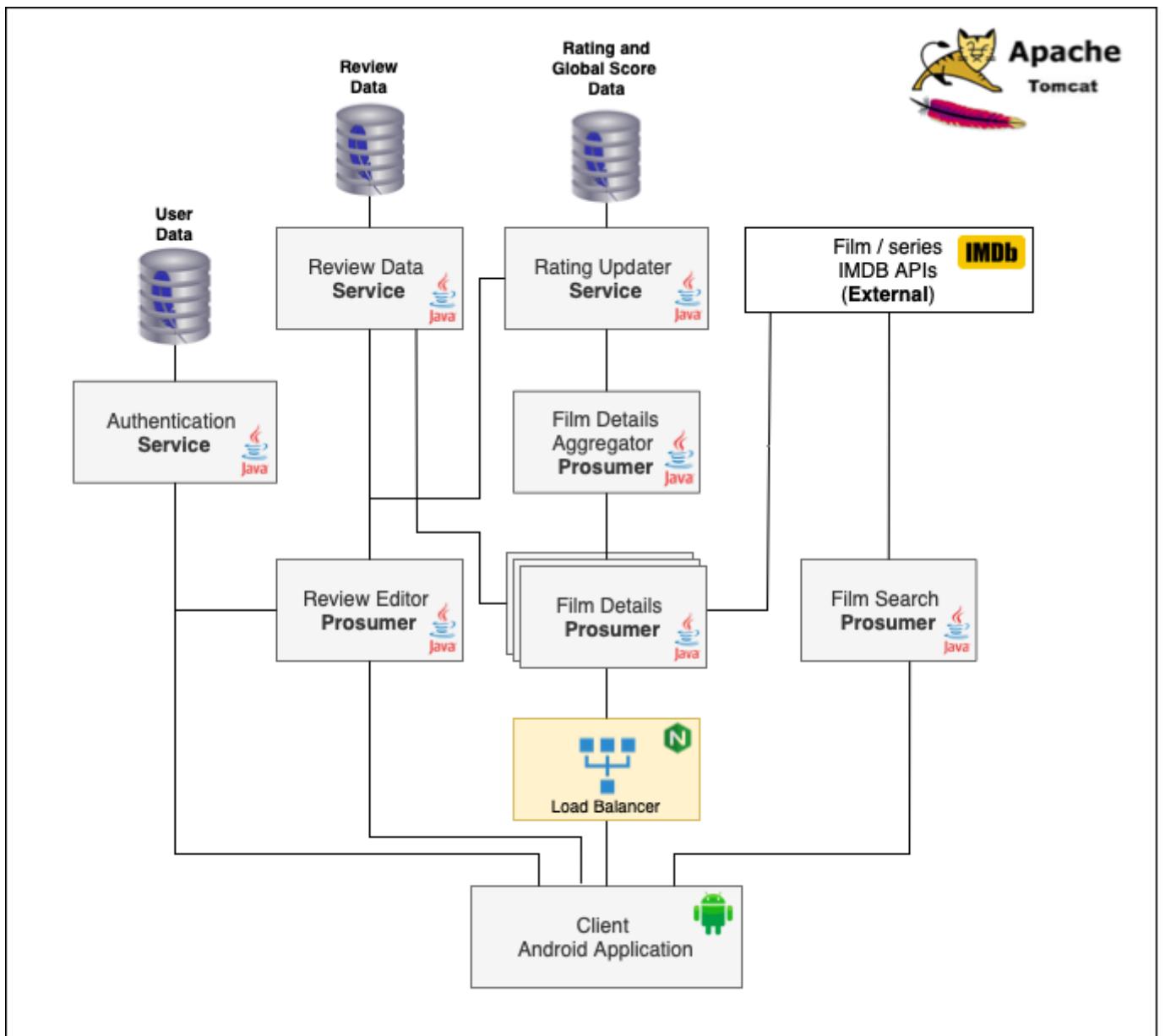
Non Functional Requirements

- Load balancers on services/prosumers are subject to numerous requests in order to improve performance.

Architecture

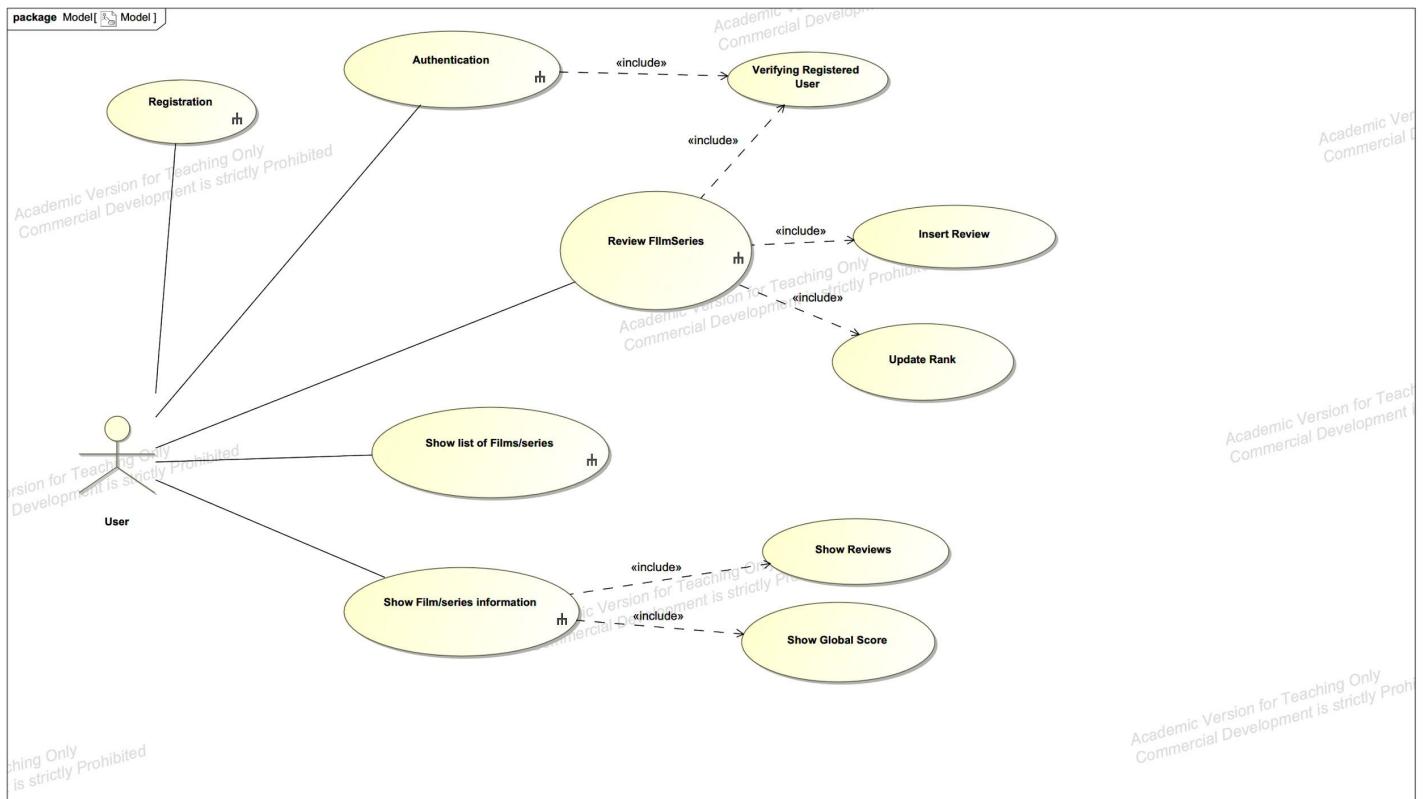
Component Diagram

Ref.: [Architecture-SOSE_Shared.drawio](#)



System Architecture - Component Diagram

Use Case Diagram



Use Case Diagram

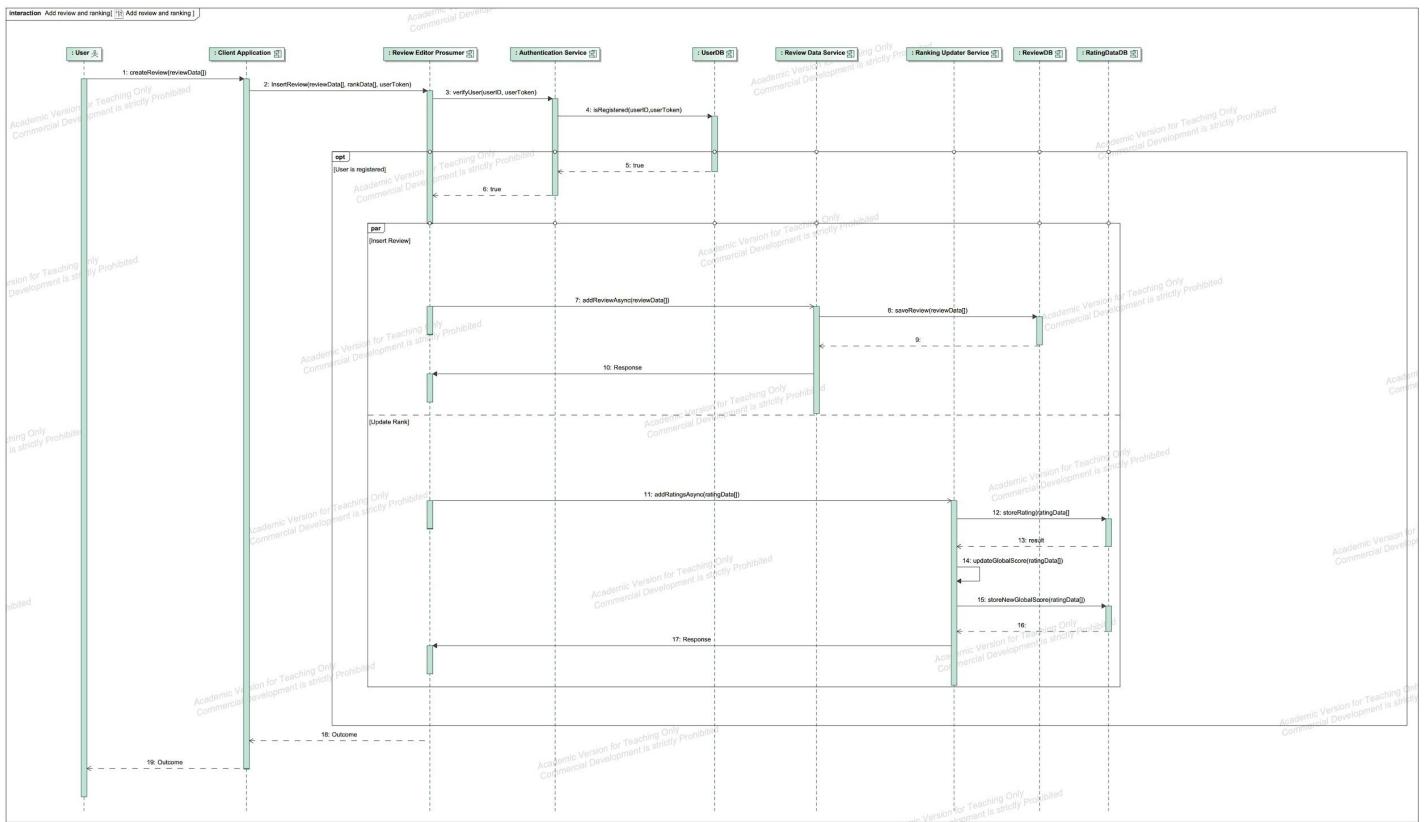
The diagram above figures out the operations supported by the application.

The operations are:

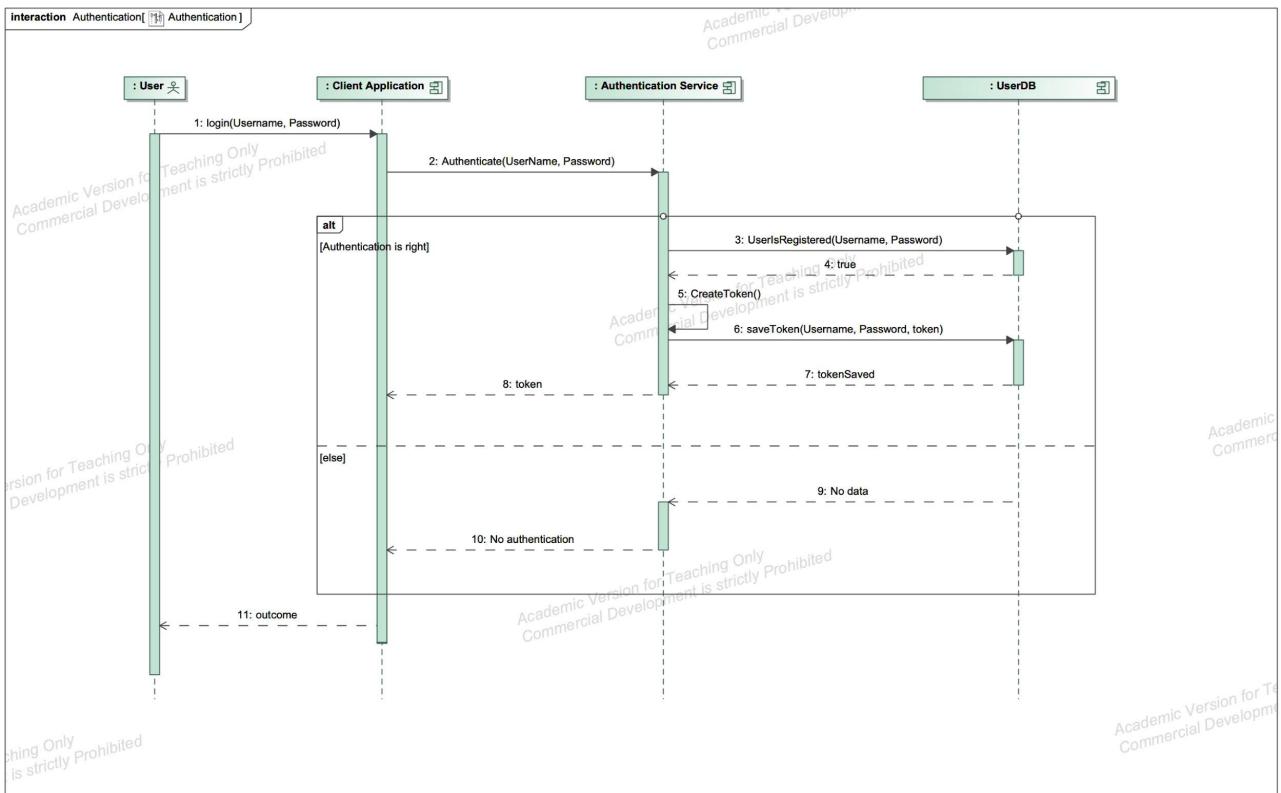
- **Registration** of the user
- **Authentication**/login of the user
- **Verifying Registered User** i.e. checking that the user is correctly registered to the application
- **Review FilmSeries**: a user can edit a review and rates of a movie and insert them to the system
- **Insert Review**: a user can insert a review
- **Update Rank**: a user can insert a rating
- **Show list of Film/series**
- **Show Film/series information**
- **Show reviews** that the users have written.
- **Show global score** calculated by the ratings.

Sequence Diagrams

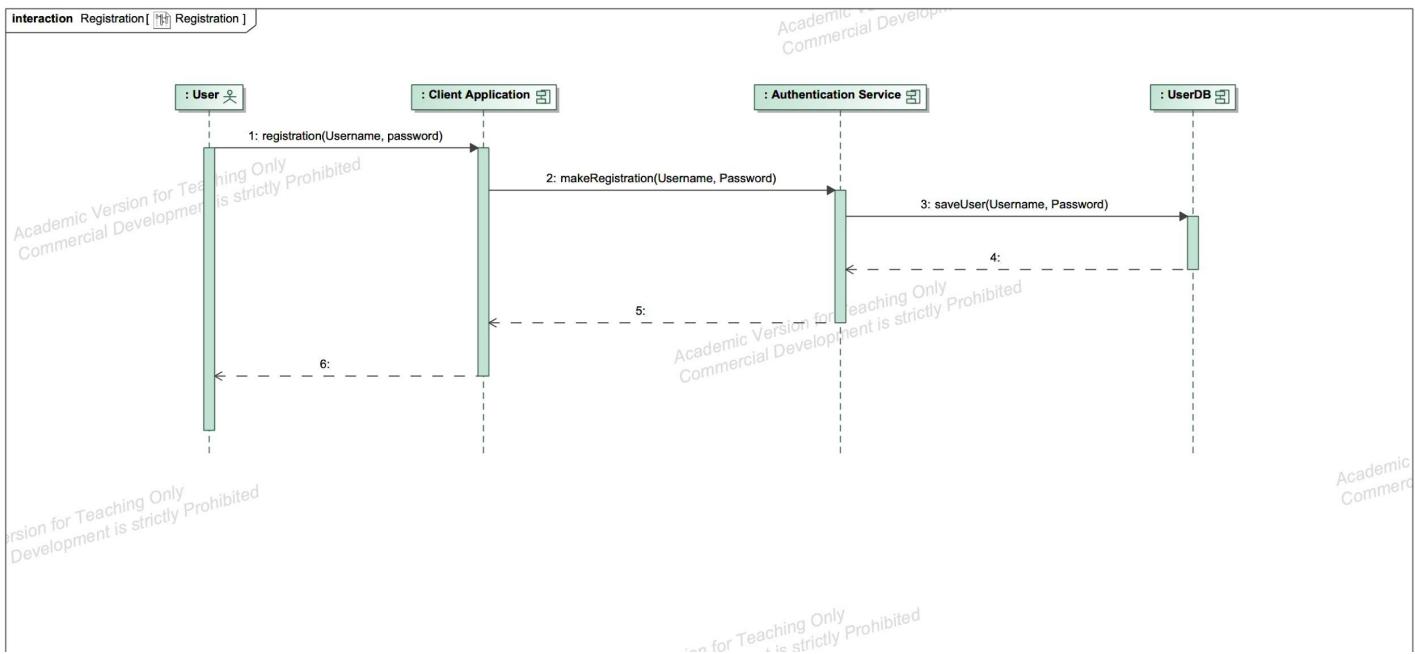
Add Review and Ranking



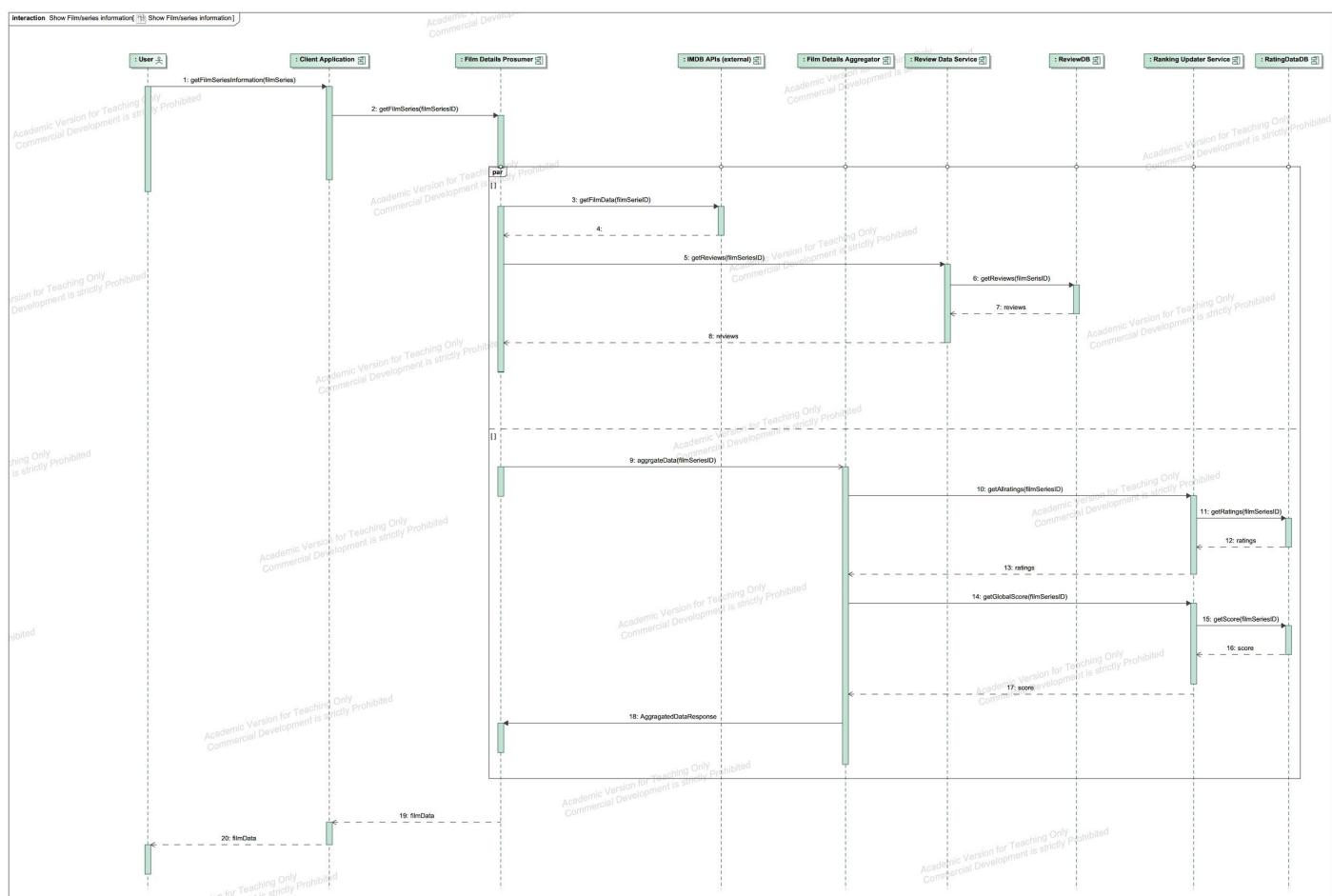
Login



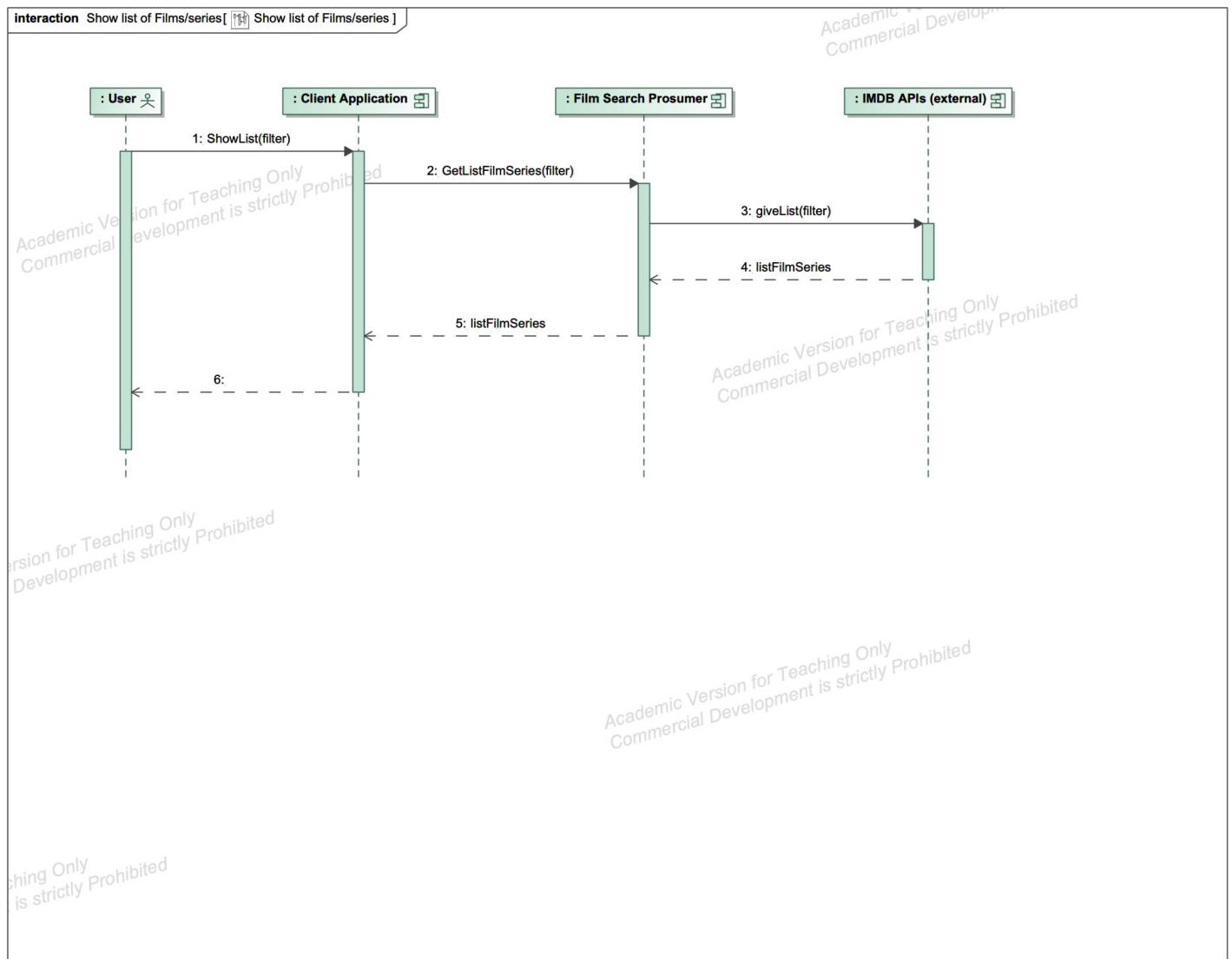
Registration



Film Details



Film Search



All the sequence diagrams are available on this google drive folder: [sose_project_diagrams](#)

Services Description

The services to be implemented in the system are those shown in the architecture and in this section, they will be described in detail.

Service Name	Type	Protocol	Sync / Async	Exposed operations
Authentication	<i>Service Provider</i>	REST	Sync	<ul style="list-style-type: none">- <i>login</i>- <i>register</i>- <i>getUsernameByld</i>
Rating Updater	<i>Service Provider</i>	REST	Async	<ul style="list-style-type: none">- <i>addRatings</i>- <i>addRatingsAsync</i>- <i>getRatingAvgs</i>- <i>getAllRatings</i>- <i>getGlobalScore</i>
Review Data	<i>Service Provider</i>	REST	Async	<ul style="list-style-type: none">- <i>getReviewsByUserID</i>- <i>getReviewsByFilmID</i>- <i>getReviewByFilmIDUserID</i>- <i>insertReview</i>
Film Details	<i>Prosumer</i>	SOAP	Sync	<ul style="list-style-type: none">- <i>getFilmDetails</i>
Film Details Aggregator	<i>Microservice</i>	SOAP	Sync	<ul style="list-style-type: none">- <i>aggregateRatings</i>
Film Search	<i>Prosumer</i>	SOAP	Sync	<ul style="list-style-type: none">- <i>searchFilmsByName</i>- <i>searchOnlyFilms</i>- <i>searchOnlySeries</i>- <i>searchEpisodes</i>
Review Editor	<i>Prosumer</i>	SOAP	Sync	<ul style="list-style-type: none">- <i>insertReview</i>

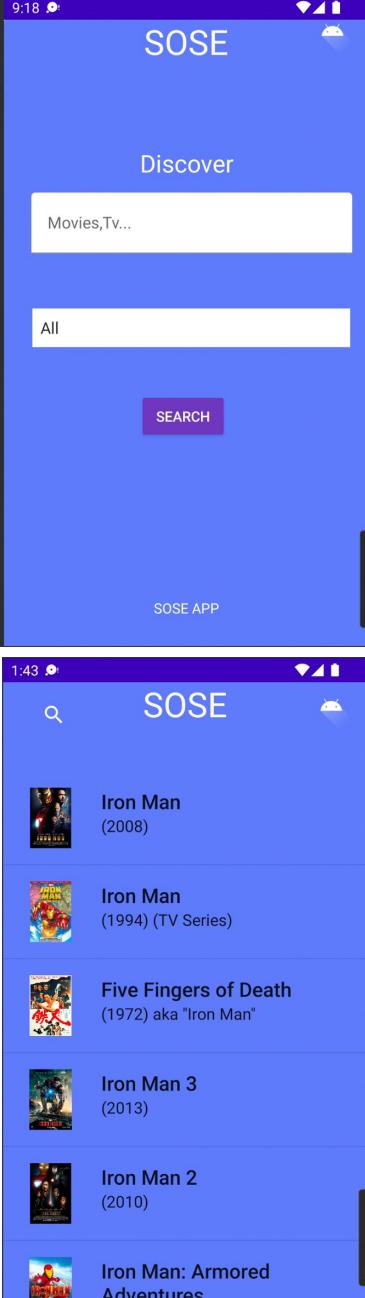
Development Details

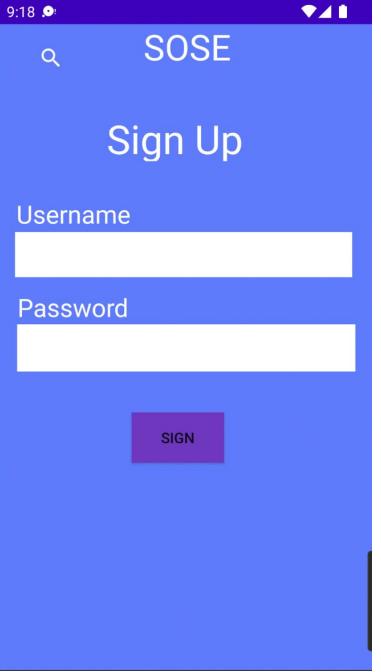
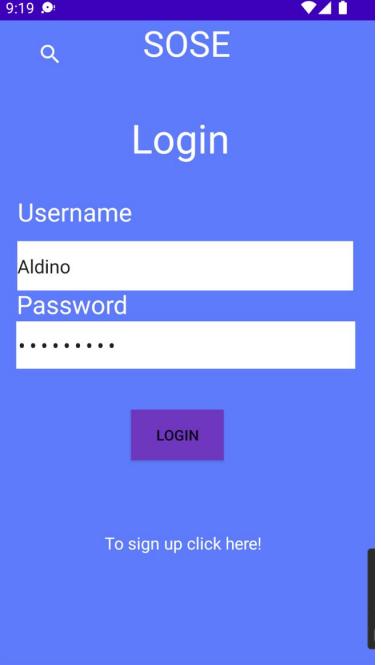
All the implementation has been stored on two GitHub Repository. The two repositories contain a **readme** file which describes all the steps to build and run the project.

Link-Repository-Servers: https://github.com/federix98/SOSE_Project

Link-Repository-Android: <https://github.com/GianlucaRea/Application>

Project Demo

Functionality	Screenshot	Description
Search Film		<p>In this couple of views we can see how the search for any kind of entertainment. On the home page of the app, we can see a search bar where we can write the title we are searching for.</p> <p>The second Item is a choice box (spinner in android) where we select the typology of the search.</p> <p>The typologies implemented are 4. All, Only Movies, Only Tv Shows and Episodes.</p> <p>In the second view, we can find the results of our search (with input Iron man and choice box on all).</p> <p>The app will show any kind, accordingly with the choice box, of entertainment with the specific name.</p> <p>On the top left we have the button to go back to the search view and on the top right we have the button for the login section of the application.</p>

Registration		<p>This view is accessible through the login view. In this case, we have two editText inputs where we gonna put our chosen username and our password. After the sign up we will be sent back to the login view. On the top left, we have the icon to go back to the search view.</p>
Login		<p>This is the login page of the application where after our user is created (if it's already created we can directly login from here) we can access the application gaining some privilege (i.e Adding a review to a movie). After the login button is pushed we will be sent back to the home page.</p> <p>On the homepage, if everything is done correctly we will not be able to go back to the login page because we will be already logged in. On the top left, there is the icon to go back on the home back. At the bottom of the page, there is clickable text to go to the sign-up page.</p>

Film Details

The image contains two screenshots of a mobile application interface for movie details. Both screenshots show a header with the title 'SOSE' and a search icon. The top screenshot shows a movie poster for 'Iron Man' and a detailed description of the film, including its release year (2008), genre (Action, Adventure, Sci-Fi), production companies (Paramount Pictures, Marvel Enterprises, Marvel Studios), cast (Robert Downey Jr., Gwyneth Paltrow, Terrence Howard), and languages available (English, Persian, Urdu, Arabic, Kurdish, Hindi, Hungarian). It also includes a 'Ratings' section and a note that the film has a global score of 0. The bottom screenshot shows the same information but with a higher global score of 10.0, indicating a review has been added. The 'Reviews' section lists one review: 'Meraviglioso Film stupendo'.

On this page there is all the information about a specific movie we have chosen to open from the result of the search. Only a few items have already reviews and stuff inside it (The Lost tv show can be filled by the python script available at the repository.).

So when we visit a movie that is never revied we will find every rating to 0 and an empty list of reviews.

The page is formed by the title, the image and the description that is generated by gathering all the information about the specific movie. In this example, a review is already added to the system. The bigger global score we encounter is generated by the servers calculating the summary score of all the rating points. On the box below there will be the average for each point of a rating including the number of ratings the films have. On the button of the page, there is the review list. In this case, there is only one review but we suggest checking the Lost detail page after using the python script.

On the top left of the page, we have the go-home button. At the top right, we have two icons available. The first one is the add review button that will work if only the user is logged in and beside it, there is the login icon.

Add rating and review		This view shows the insert review form that we can use to add a review to a specific movie. Only available for logged users. We can insert multiple parameters. The review title and its comment. The rating for each point of the review (Actors, Costume, Dialogue, Film Director) and at the top bottom the total score we give to the specific movie. After the insert of the review, the user will be sent back to the film detail page where he will be able to see the change on the rating and review box. At the top left of the page, we can see the go-home button.
-----------------------	---	---

Conclusions

Meeting the project requirements

All the functional requirements specified in the section above have been realized. The user can search for film and tv series details and make reviews and ratings. A custom global index score has been also developed to resume the ratings of the users of the film. The NFRs have been achieved too. The update of the global index score and the storage of the review are parallel asynchronous tasks. The load balancer has been added to increase the performance of the system.

Meeting the final test specification

The application that has been realized contains both REST and SOAP services developed by using Apache CXF. It is composed of:

- **4 service providers** (one is the external IMDB API Service for retrieving film and tv series information, the others have been implemented from scratch). Two service providers have been realized as asynchronous services.
- **4 prosumers**
- An **Android application** that simulates all the possible uses of system services
- A **Load balancer** for the film details prosumer (the one that is estimated to be called more times)

The client interacts with 3 prosumers and 1 service provider. Two of the prosumers interact with more than 1 service provider.

The component diagram and the sequence diagrams are inserted in this documentation in the architecture section.

All the REST services are equipped with OPEN API and Swagger for the documentation.

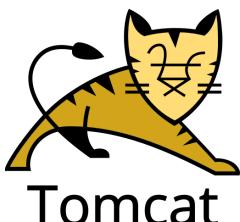
Used Technologies



Apache CXF



Swagger



Tomcat

