UNIVERSITÀ
DEGLI STUDI
DE L'AQUILA

DISIM
Dipartimento di Ingegneria
e Scienze dell'Informazione
e Matematica

Project Documentation

2023/2024

Course: **Service Oriented Software Engineering**

University of L'Aquila

EatFit&Review

Adam Bouafia adam.bouafia@student.univaq.it

Summary

Domain

The EatFit&Review system aims to manage user reviews and rankings on various foods. Users can share their ratings and opinions about the foods they consume. The application has a dual purpose: first, to allow users to consult existing opinions and understand which foods might be more suitable for them; second, to enable users to share their experiences to actively participate in the community.

The system will provide the following types of feedback to the users:

- **Review**: A textual comment where the user can describe their thoughts about a particular food.
- **Rating / Ranking**: Users can rate foods based on different parameters such as taste, nutrition, and overall satisfaction.

Requirements

Functional Requirements

#	Functional Requirement	Priority
1	**Authentication** (login and registration): Since the application allows users to publicly release reviews and ratings, users must be registered and logged into the system. Therefore, login and registration functionalities are essential.	**MEDIUM**
2	Search for Foods: show the list of the food based on selected filters by the client	**HIGH**
3	Show all the details about a food (including reviews of the other users)	**HIGH**
4	Users can add reviews and ratings for different foods. **The rating parameters include:** A. Taste B. Nutritional value C. Overall satisfaction D. Packaging E. Price	**HIGH**
5	The system must calculate a global score through a heuristic/ algorithm/linear combination (0-100) for each food. **The heuristic should be able to update incrementally without reconsidering all previous reviews individually.**	**LOW**

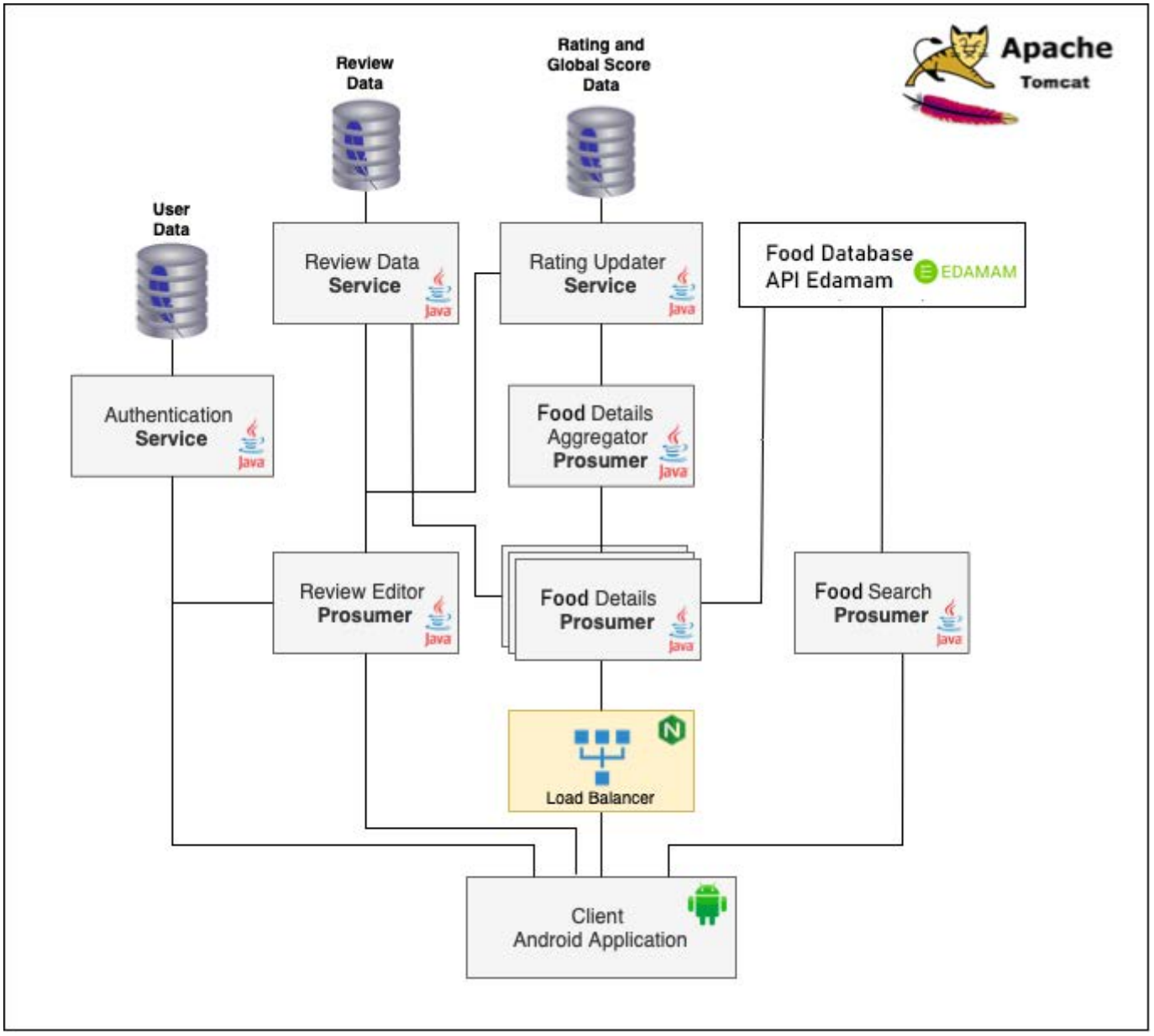
Non Functional Requirements

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

Architecture

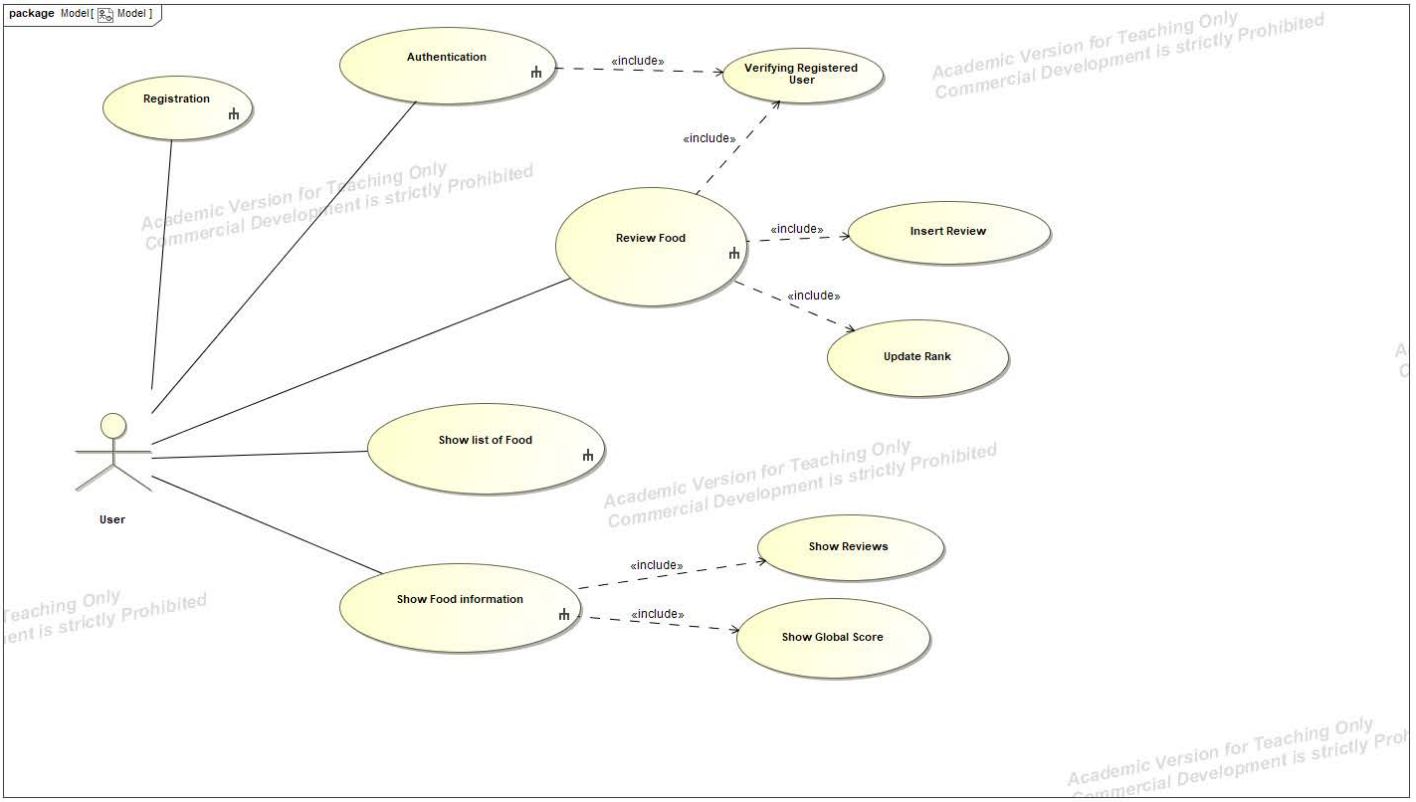
Component Diagram

Ref.: [Architecture-SOSE_Shared.drawio]



System Architecture - Component Diagram

Use Case Diagram



Use Case Diagram

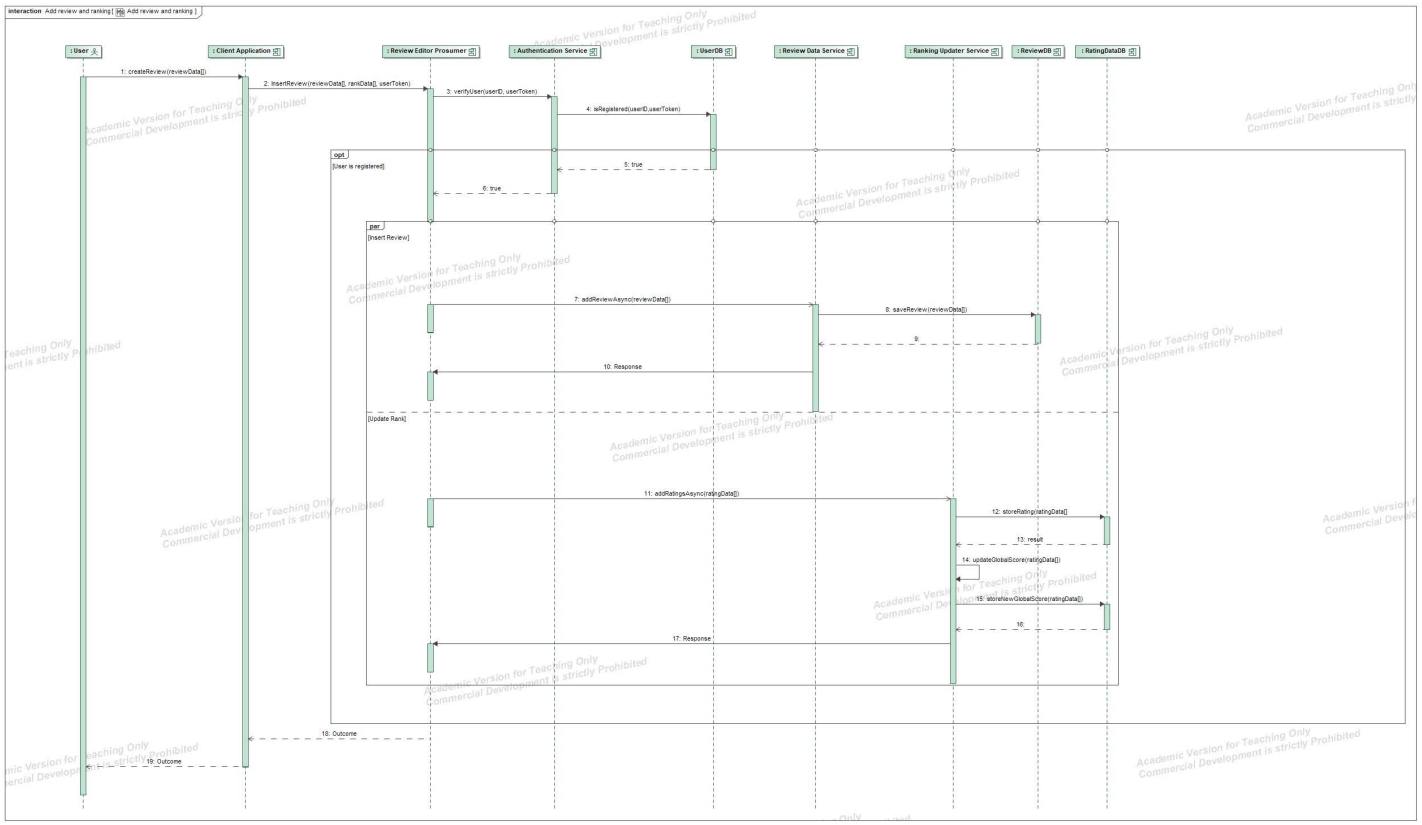
The diagram above figures out the operations supported by the application.

The operations are:

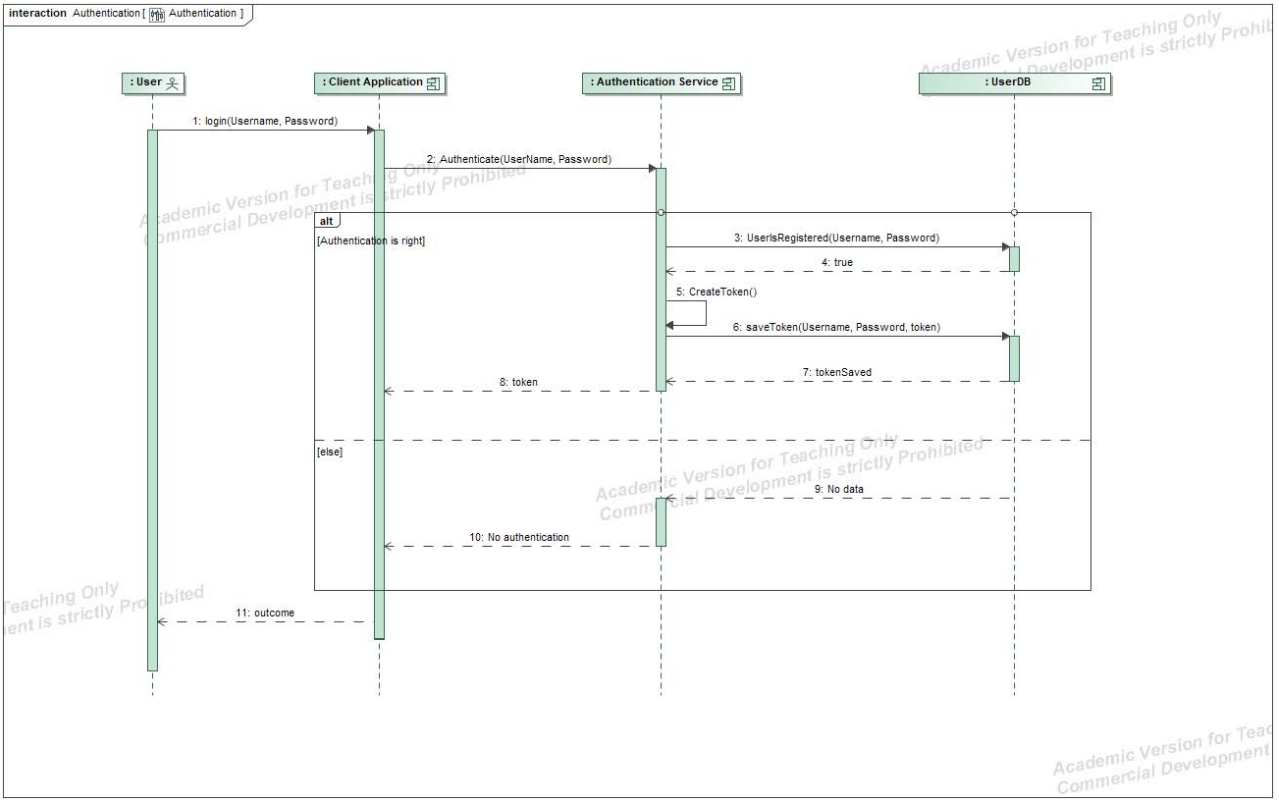
- **Registration** of the user
- **Authentication**/login of the user
- **Verifying Registered User** i.e. checking that the user is correctly registered to the application
- **Review Food:** a user can edit a review and rates of a food and insert them to the system
- **Insert Review**: a user can insert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

Sequence Diagrams

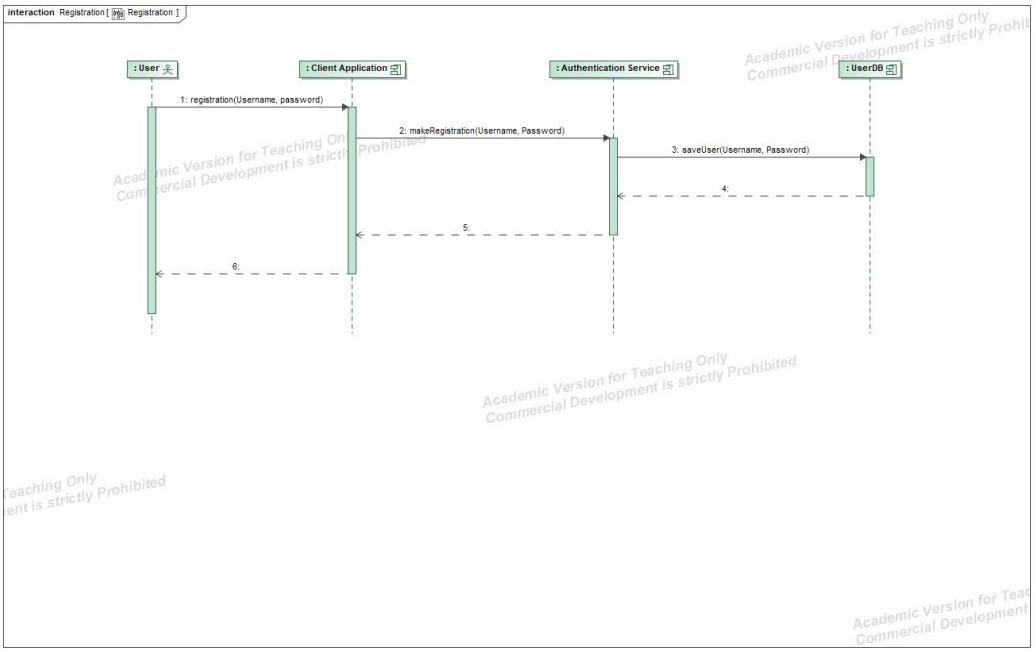
Add Review and Ranking



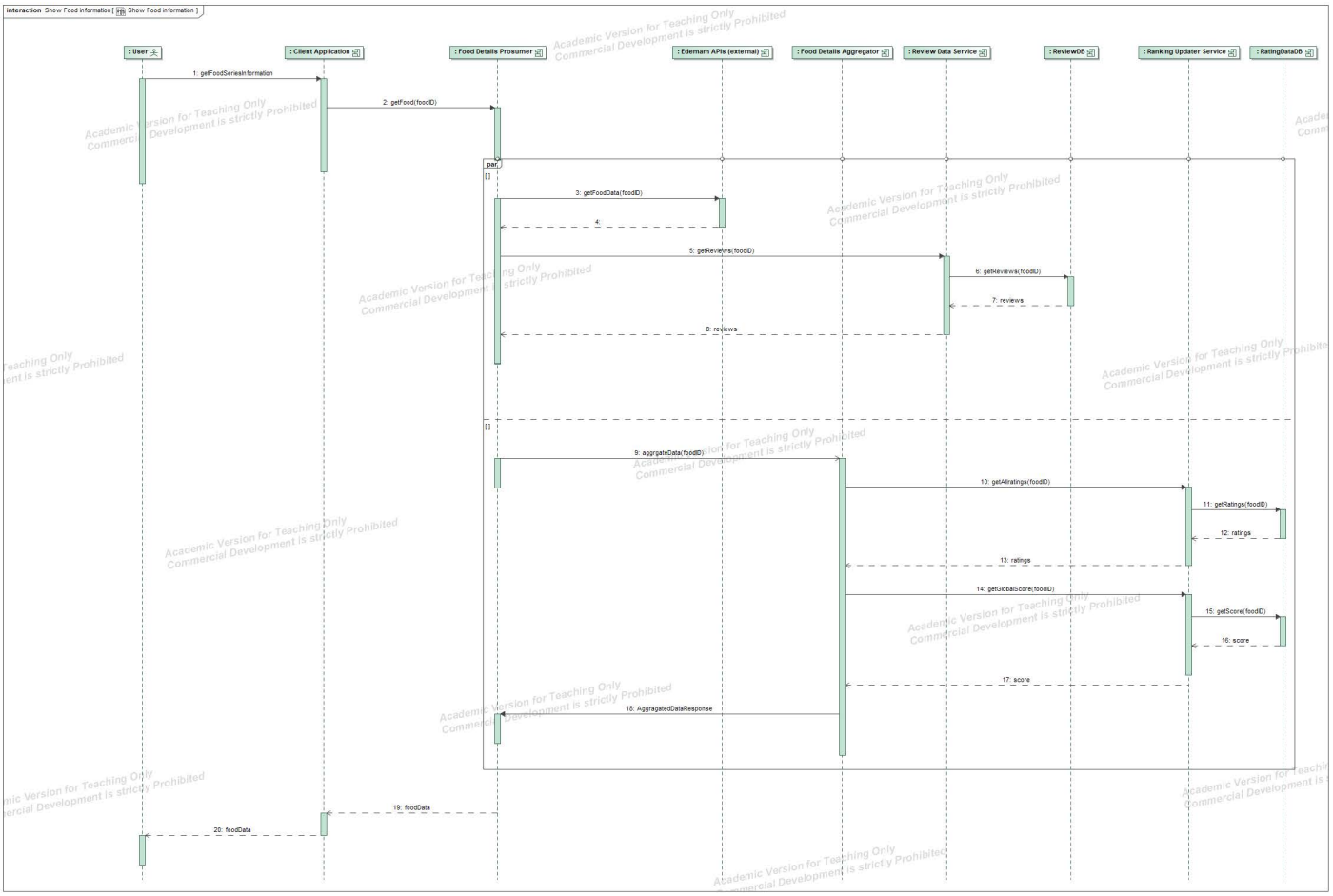
Login



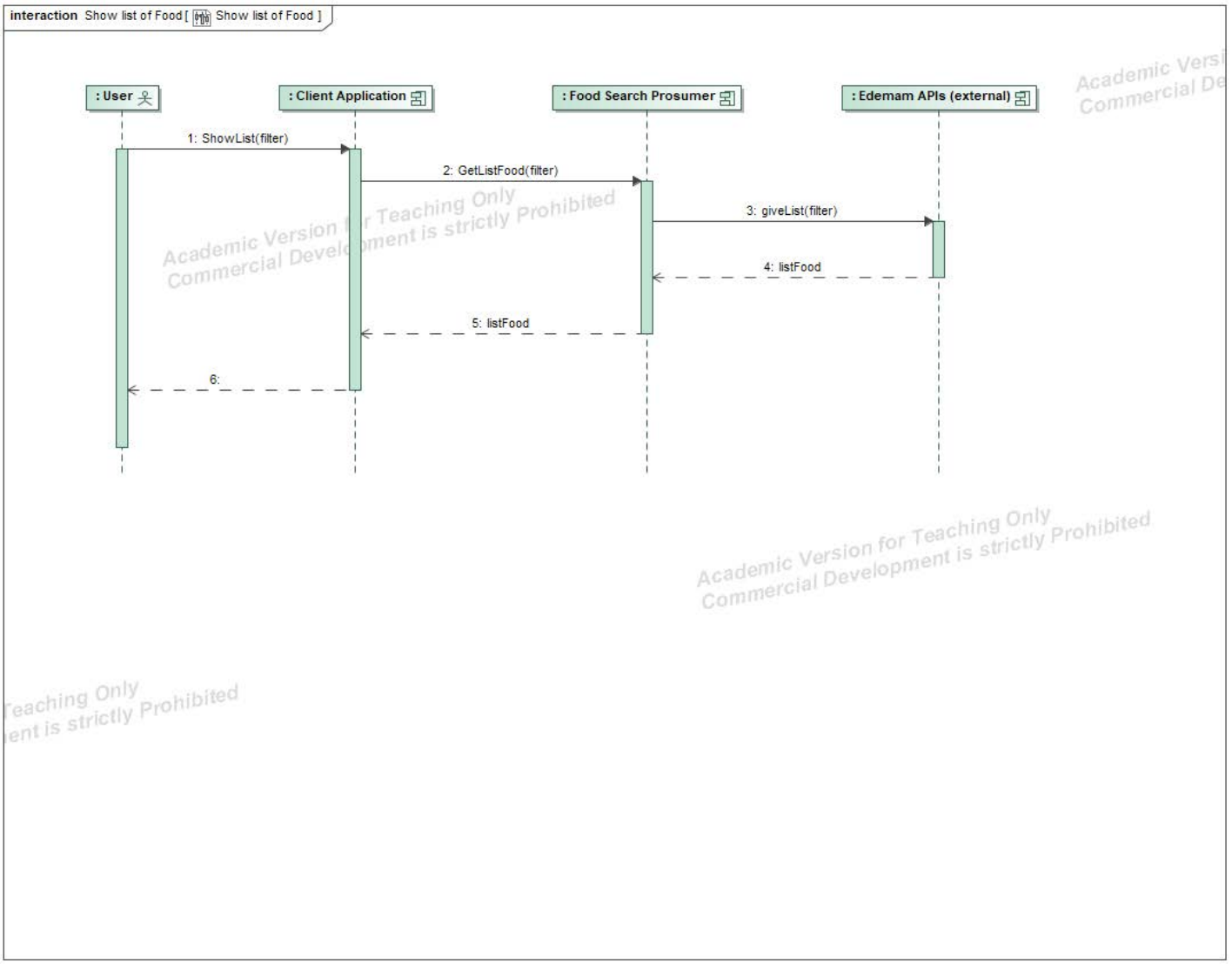
Registration



Food Details



Food Search



All the sequence diagrams are available on Github

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	*Service Provider*	**REST**	Sync	- *login* - *register* - *getUsernameById*
Rating Updater	*Service Provider*	**REST**	Async	- *addRatings* - *addRatingsAsync* - *getRatingAvgs* - *getAllRatings* - *getGlobalScore*
Review Data	*Service Provider*	**REST**	Async	- *getReviewsByUserID* - *getReviewsByFoodID* - *getReviewByFoodIDUserID* - *insertReview*
Food Details	*Prosumer*	**SOAP**	Sync	- *getFoodDetails*
Food Details Aggregator	*Microservice*	**SOAP**	Sync	- *aggregateRatings*
Food Search	*Prosumer*	**SOAP**	Sync	- *searchFood*
Review Editor	*Prosumer*	**SOAP**	Sync	- *insertReview*

Project Demo

Functionality	Screenshot	Description
Search Food		On the home page of the app, there is a search bar where users can enter the name of the food item they are searching for. Alongside the search bar, there is a choice box (spinner in Android) where users can select the category of the search. The categories implemented are: • **Generic Foods** • **Generic Meals** • **Packaged Foods** • **Fast Foods** In the second view, we can find the results of our search. For example, if the input is "apple" and the choice box is set to "Generic Foods," the app will display all generic foods that match the name "apple." **Back Button:** On the top left, there is a button to return to the search view. **Login Button:** On the top right, there is a button for the login section of the application.

Registration		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.
Login		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 food). After the login button is pushed we will be sent back to the home page. 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.

Food Details		On this page there is all the information about a specific food we have chosen to open from the result of the search. Only a few items have already reviews and stuff inside it So when we visit a food that is never reviewed 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 food. 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 food. Only available for logged users. We can insert multiple parameters. The review title and its comment. The rating for each point of the review (Taste,Nutritional value,Overall satisfaction;Packaging;Price) and at the top bottom the total score we give to the specific food. 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 Edamam API Service for retrieving food 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