Beer-Wine Recommendation

Participants

- Raffaele Perini - 196339 - raffaele.perini@studenti.unitn.it
- Giovanni Rafael Vuolo -196502 - giovannirafael.vuolo@studenti.unitn.it

GitHub Repository

- https://github.com/intro2sde-project

Heroku Services

- https://orchestrator-sde.herokuapp.com/ - REST
- https://beer-recom.herokuapp.com/ - SOAP
- https://wine-recomm-rest.herokuapp.com/ - REST

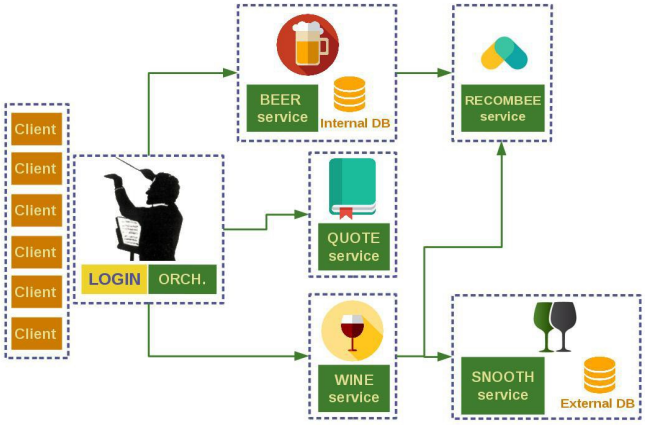
API

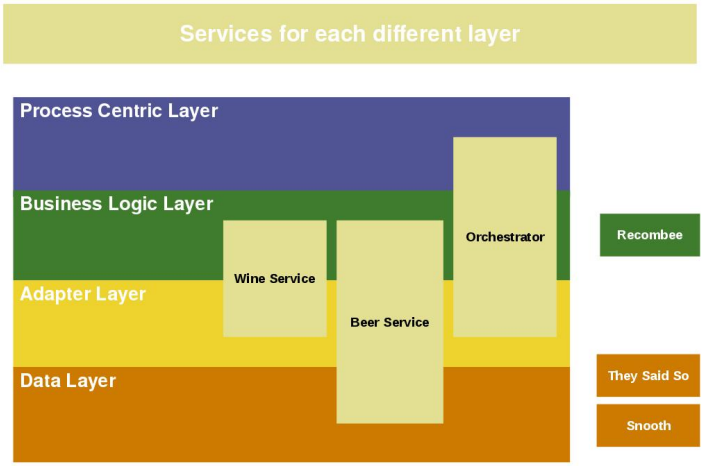
- beerwinerecom.docs.apiary.io
- beerecom.docs.apiary.io
- winerecom.docs.apiary.io

Abstract

The project collects several services, which let the user deal with beer and wine “items”. For instance, search, evaluate, obtain recommendations based on user preferences and a few more.
This project is composed by three external services, one for the recommendation, one for an external database (wines) and one for a feed of quotes. The other services, which were locally implemented, are an adapter for the external recommendation system plus a local database for the user and the beer administration, an adapter for the external database and the recommendation system and lastly an orchestrator which calls these two services (which call the external services) integrating them and exposing them as a single service to the user through a web page.

Overall Architecture





Our service is composed by six different services, which communicate with each other through the exposed API.

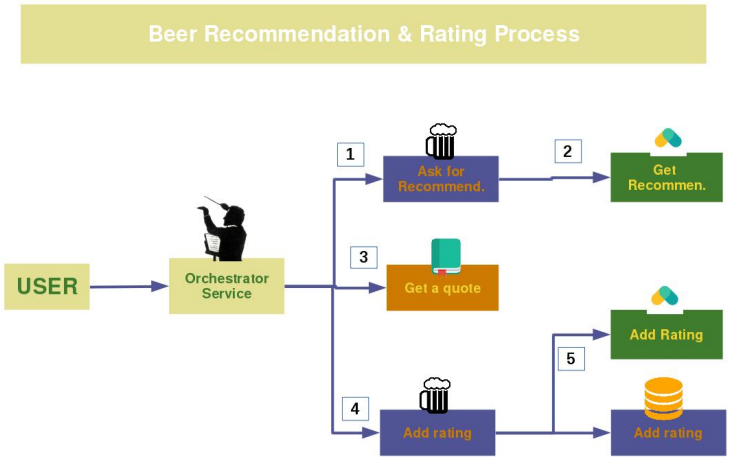
- Recombee - recommendation system.
- Snooth - external database for wines.
- They Said So - external database for quotes.
- Wine Service - adapter for Recombee and Snooth, expose the logic of the wine administration and recommendations
- Beer Service - local database for users and beers and administration (CRUD) of the entities; adapter for Recombee; expose the logic of the beer recommendations; store rating and other interactions between user and beer.
- Orchestrator - logic for login; integration of the two recommendations logic (beer and wine); exposes methods to final user, such as, search for an item, get recommendations, give/get rating for items, add/remove items to wish-list, get quotes.

Implemented Services

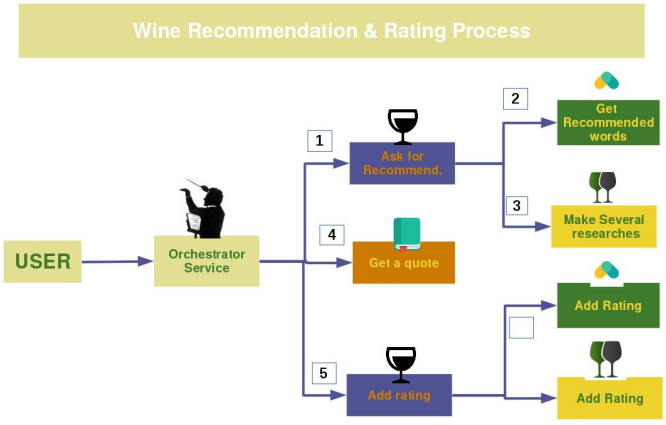
Beer - The operations implemented by this service are: user registration, which adds a new user to the local and external (Recombee) database; recommendations, which calls the external API to obtain beer recommendations; searching; which display the beers present in the db; preference (wish-list), which adds the beer to the wish-list of a user; rating; which gives a rate to a beer.
Wine - The operations implemented by this service are: create username registration, which adds a new user to the external databases (Recombee and Snooth); recommendations, which calls the external API to obtain wine recommendations; searching; which display a list of wine searched by a certain property; preference (wish-list), which adds the wine to the wish-list of a user; rating; which gives a rate to a wine.

Orchestrator - This service uses the methods from beer and wine service. Additionally, implements a login method and call the external quote API.

Main Processes



Beer Recommendation and Rating Process - Supposing the user is thirsty and has already given his preferences of has made some interactions with beers, he now wants to receive a list of beer recommendations. These are the steps that our application makes: 1. the orchestrator ask for a recom. to the beer service; 2. the beer-service ask Recombee for the recom; 3. in the meantime the orchestrator ask for the quotes and displays all (quote and recom); 4. at this point the user can evaluate the recom by giving a rate. 5. the rate is pass to the beer-service and Recombee.



Wine Recommendation and Rating Process - The process is similar to the beer one, but, since we used an external database, instead of asking for a recom. we ask for a recommended searching parameters. These parameters are used in, different and boosted, automatic search to present a list of recommended wines.