Final Project Report

DAVE3615 - Software Architecture and Frameworks

s337380

Aymeric Berthoumieu

Exchange Student

Table of contents

1. Introduction

The aim of the project is to develop an “Amazon clone”. In short, we have to implement an application which has the same main functionalities than Amazon.

The purpose of this project is to use all the knowledge we gain during this course and put it all together. In fact, it will allow us to build our first enterprise application based on micro-services in order to be reliable 24/7 with a big traffic. Finally, we practice the deployment of our application in the cloud.

2. High Level Diagram

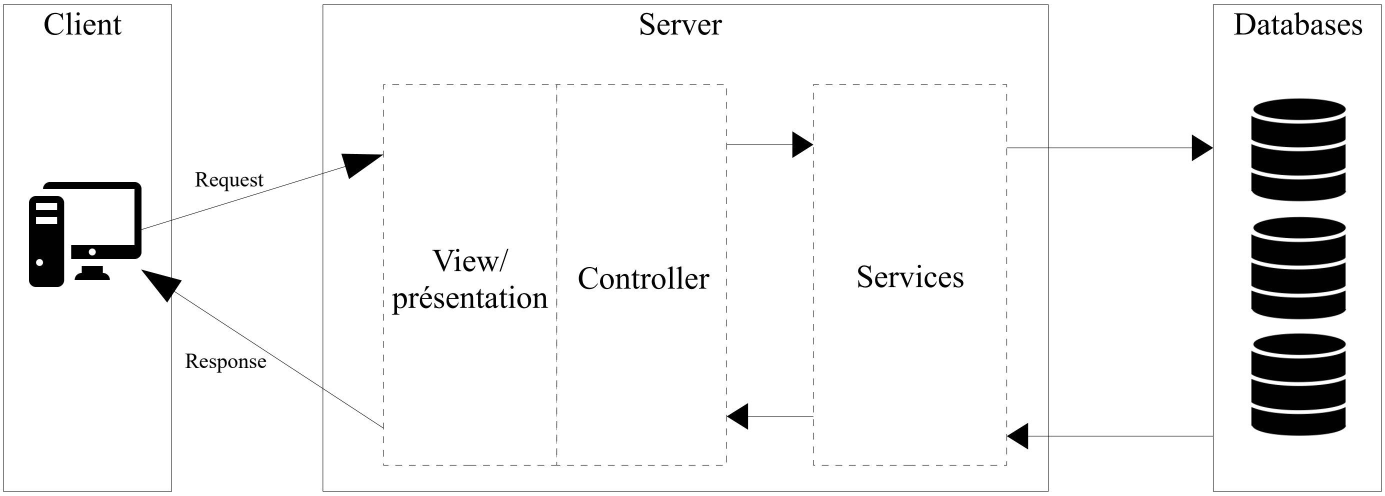


Figure 1: High level diagram

The client communicate via the presentation layer which transmits information to the controller which can then use the relevant services. Each service, using it's own database, will execute the task needed and send back to the controller the information required if it is necessary. Then the controller can respond to the client request via the view.

3. Database Model

I use different databases for the different micro-services which require it.

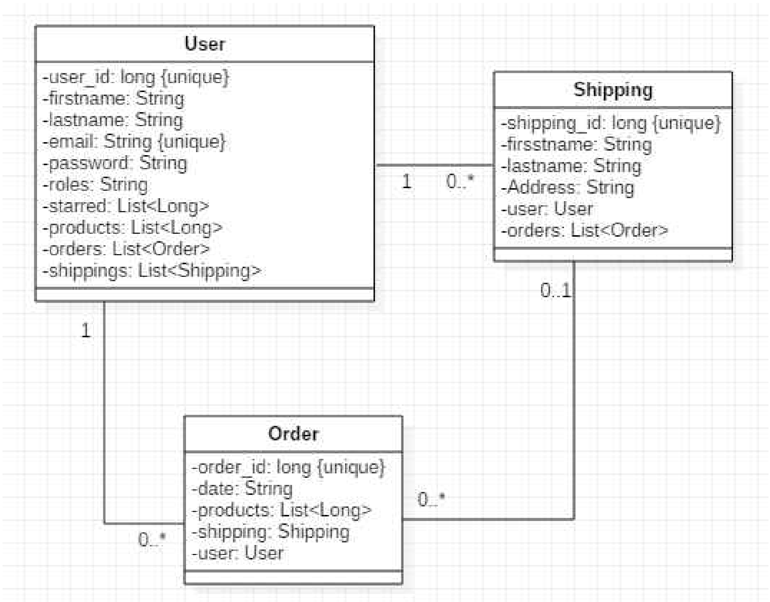


Figure 2: Database model of user micro-service

In this diagram, a User in linked to Shipping with one to many (a Shipping has always a User and a User can have 0, 1 or more Shippings).

The link between User and Order is exactly the same as between User and Shipping.

Finally, a Shipping can be mapped to 0, 1 or several Orders and an Order can have 0 or 1 Shipping.

In User and Order the *List<Long>* attributes correspond to a list of ids of products which are stored in another database.

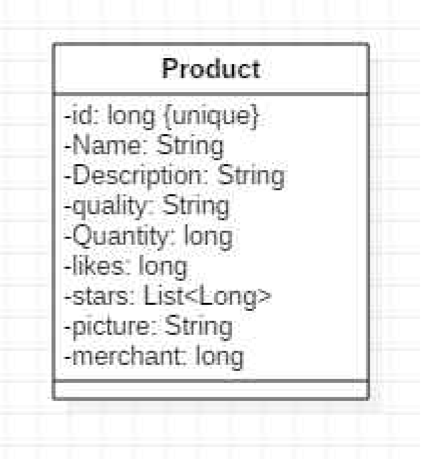


Figure 3: Database model of product micro-service

In this database I only store products.

The attribute *stars* is a list of the stars given by the users (the controller can then calculate the average, another technique would have been to store the current average and the number of user who had starred the product so far to calculate a new average when an other user stars the product)

The attribute *picture* is the path to the picture of the product and *merchant* is the id of the merchant selling this product.

4. Functionality List

The following list is the functionalities provided by the application.

Account management

- Create an account (normal user or merchant).
- Login, logout.

Profile management

- User, Merchant and Admin have a profile page.
- User and Merchant can add addresses on their profile page.
- User, Merchant and Admin can change names on their profile page.
- User, Merchant and Admin can change email on their profile page.
- User, Merchant and Admin can change password on their profile page..
- User, Merchant can delete their own account.

Products' features

- Products can be seen without login.
- Detail page for each product.
- Product have stock information. If it is out of stock it can still be seen but not ordered.
- Product can be liked, unliked.
- Product can be starred.
- Products have picture, description, quality information and quantity information.

Users' features

- User can search for a product (using id, name or description (partial words work)).
- User can order and specific a quantity of a products.
- User can ship the current order to its address.
- User can access his order history.
- User can the details of an old order.

Merchants' features

- Merchant can see the products he his selling.
- Merchant can sell new product.
- Merchant can update a product he is selling.

Admins' features

- Admin can look at all users and merchants.
- Admin can change password of a user or a merchant.
- Admin can delete a user or a merchant.

5. Micro-Services

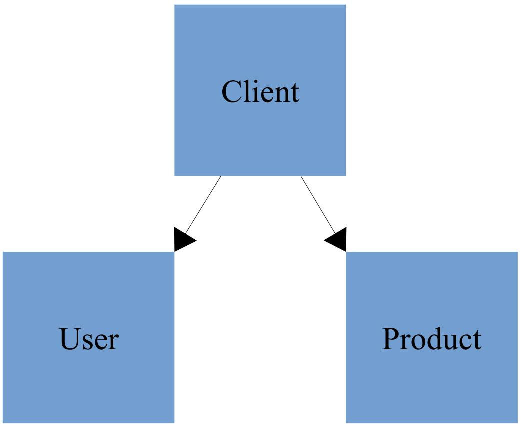
Name	Client service
Database Storage	No database storage
Description	This service implements the security and the login service. It also provides the view for the client. It takes the URL requested by the client and all path-variables required and send back the view asked by the client if he has the right rights. If note it redirect him to an error page.

Name	User service
Database Storage	Independent H2 storage
Description	This service is a Rest-API using JSON which can be used by other services. It gives access to the information about users, orders and shippings. Through this service you can create, get, modify and delete a user, shipping or order.

Name	Product service
Database Storage	Independent H2 storage
Description	This service is a Rest-API using JSON which can be used by other services. It gives access to the information about products. Through this service you can create, get, modify and delete a product.

6. Micro-Services Dependencies

Figure 4: Dependencies diagram



7. Architectural Diagram

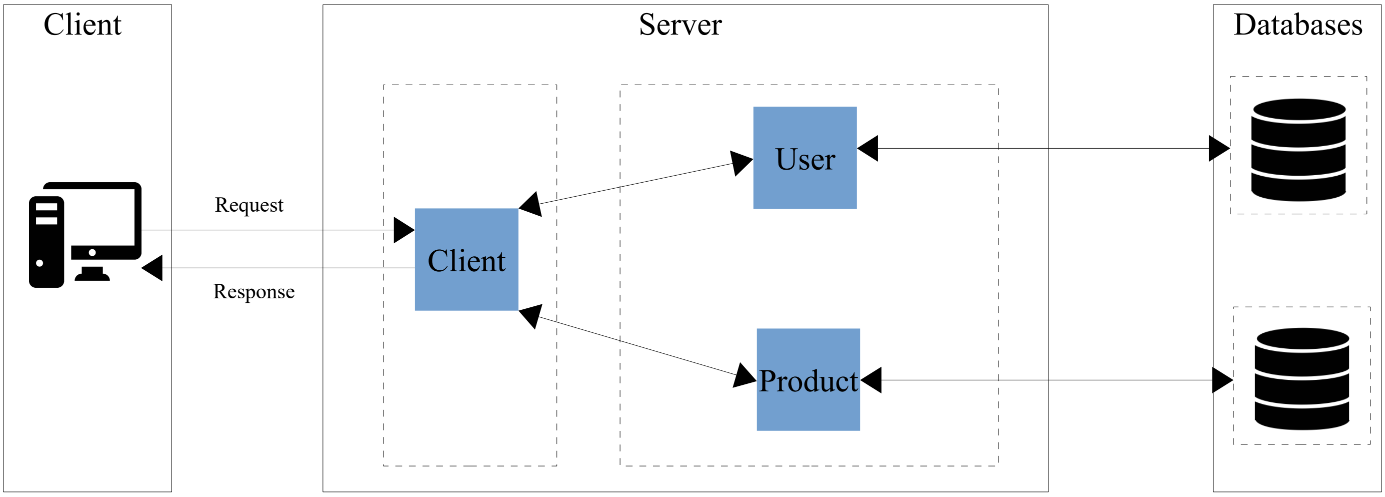


Figure 5: Architectural diagram

As mentioned in 2. High Level Diagram clients interact with presentation/view-layer in client service. This service is the main entry point. It must handle all URL path-variables, with some exceptions for API-requests.

When a request from the client need data to be answered correctly, the client service send request the relevant API (which can be both of them). This API analyzes the request and interact with its own database to update it or get the information requested by the client service. The later can then process the data and display them to the client via the views.