Université Gustave Eiffel

REPORT

Web services project

Implementation of Store to buy and Sell products with RMI and WSDL between EiffelCorp and IfShare

Supervised by:

DR. ZARGAYOUNA MAHDI

Elaborated by:

Aissaoui Ilhem
Ksiaa Donia
Diab Billel
Ben Said Ghada

Master 2 SIA
2021/2022

TABLE OF CONTENT

Introduction

We are witnessing an era where e-commerce is increasingly taking hold in the digital sphere, and in a context of pandemic and curfews, this type of company, notably Amazon, has exploded its sales figures. Indeed, this type of application is almost exclusively distributed and therefore is based on technologies such as RMI and web services.

RMI (Remote Method Invocation) is a java API that perform Remote Method Invocation.
RMI offers multiple advantages:
Possibility to share a service between entities furthermore remote invocation is possible:

If a client wants to use a method M of object O included in the server, RMI will allow a remote invocation, that means the client will only have access to the service of M remotely.

Web services is a technology which allows communication of connected applications in a very cost-effective manner utilizing the Web, they ensure the interoperability of distributed applications on the Internet.

The most multinational actors of e-commerce such as Amazon, Alibaba, Wish or AliExpress exploit the power of Web services, using them will reduce the execution time, and improve temporal and spatial complexity.

The project aims to allow accessibility for sale and purchasing services offered by IfShare for EiffelCorp’s employees and external clients using RMI technology and SOAP technology.

I. Description of the problem

Eiffel Corp. has just acquired IfShare, a company specialized in the sale of products (books, clothes, equipment, etc.) from individuals to individuals. It wishes to make its employees benefit from this service.
In order to manage this service, we have to implement the design and implementation of a distributed Java application , based on Java RMI. Products sold on IfShare can be bought and sold by all Eiffel Corp. employees. Employees can add notes about the products and their status upon resale. The application managing the product database and the one managing the employees run in two different JVMs.

When a person requests to purchase a type of product and it is not available, he or she is placed on a waiting list; as soon as the requested product becomes available, the person is notified and purchases the product. If there are several people on the waiting list, the "first come, first served" principle is applied.

In a second step, Eiffel Corp. wants to open its products to the outside world, enriched by the notes and comments of its employees, and make it accessible to the outside world via a Web service called IfService. It offers for sale products that have been sold at least once within Eiffel Corp. The web service allows users to view product prices, check availability, add products to a shopping cart and purchase them. To make a purchase on the web service, another web service Bank is contacted by IfService to check the availability of funds for the purchase and make the payment. The prices of the products are in Euros, but the university allows sales in any currency of the world, and must provide prices in the currency requested by the buyer. The exchange rates used must be found in real time.

1. Functional specification

It will consist on a website based on Java RMI and the use of Web Services that would:

for employees, system:

- System allows employees to view products list, description and price.
- System allows employees to sell a product.
- System allows employees to check the availability of products.
- System allows employees to purchase a product.
- System allows employees to resale a product.
- System allows employees to add note and rate and their status upon resale.
- System allows employees to request a product not available.

- System allows employees to receive a notification of a requested product when it is available.

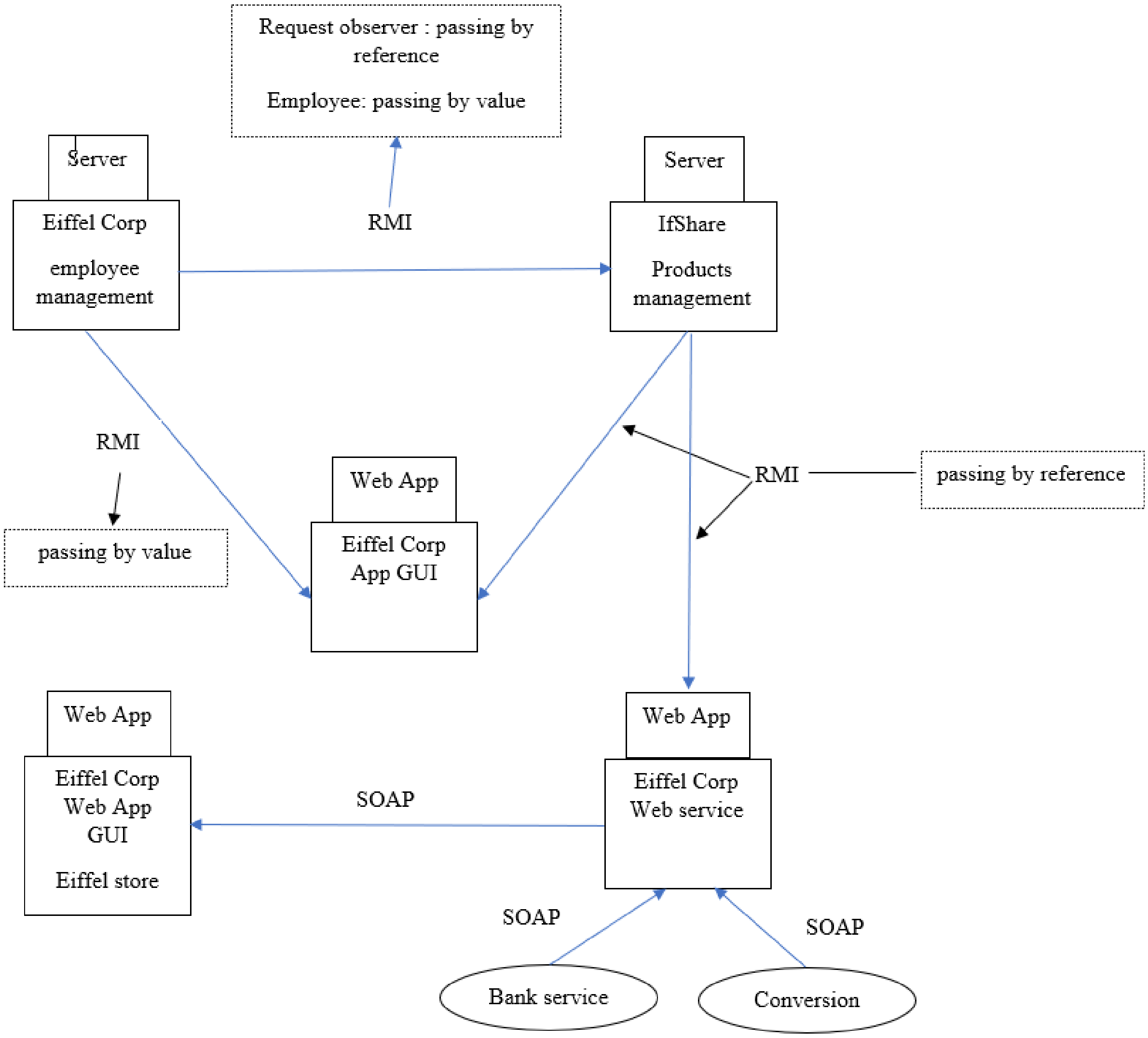
For customers:

- System Allow customers to view products list, description , price, rate and comments.
- System allows customers to check the availability of products.
- System allows customers to add products to their shopping cart.
- System allows customers to view their shopping cart.
- System allows customers to change the currency from euro to any other currency.
- System allows customers to create a bank account.
- System allows customers to deposit a sum of money in a bank account.
- System allows customers to purchase products.

2. Technical specification

- The EiffelCorp server and IfShare server run in two different JVMs
- The communication between EiffelCorp server and IfShare server must be via RMI.
- The IfShare cannot modify or manage the employees of EiffelCorp.

II. Global architecture



1. Main Structure

Our project is structured in sub-projects:

1. Serveur_IfShare: the RMI Server that offers to the Eiffel Corp employees the possibility to sale and purchase products.
2. Serveur_EiffelCorp: RMI server of EiffelCorp including all the methods to manage Employees
3. WebApp_EiffelCorp: a web application providing GUI for the Two RMI servers, where the employees can connect to the platform, sale, purchase and resale products.
4. ClientTestRMI: a RMI client for the two servers, just for test the services in console.
5. EiffelCorp_BuyService: the Dynamic Web Project that offers to the world the possibility to buy products sold at least once, via web service. This project is also an RMI Client for IfShare, to get the information about the products and a Web Service

Client for BankWebService to perform the payment, and a web service client for a conversion service.

6. BankWebService : is a Dynamic Web Project that checks the availability of funds for the purchase, making the payment in more currencies.
7. Client_WebApp: Web application, is the client of the web service that allows customers to see and buy products.

2. Entity involved

Employee: id, first name, last name, address, login, password, inbox
Customer: id, first name, last name, address, login, password, productsale, carte
Product: id, name, category, availability, price, stat, image
Feedback: id, ProductID, EmployeeID, rating, comment, dateComment
Purchase: id, idSale, EmployeeId, CustumerId, purchaseDate, Stat
Sale: id, ProductId, EmployeeID, SaleDate, Stat

III. Part 1: RMI

1. RMI services

EmployeeService:

- Add, remove and search an employee.
- Login to an employee account.

ProductService:

- Add and remove a product.
- Search a product by name, id, type and catégor.

FeedBackService:

- Add and remove and search feedback.
- Search for feedback from an employee.
- Get the feedback of a product.

PurshaseService:

- Add and remove a purchase.
- Search purchases by products, employee and id.

SaleService:

- Add modify and remove a sale
- Search purchases by products, employee and id.

2. Passing by value and passing by reference

There are two different approaches to pass parameters between remote methods invocation:

- By value: make a copy in memory of the actual parameter's value that is passed in;
- By reference: pass the real object. Then the object could be modified by client-side.

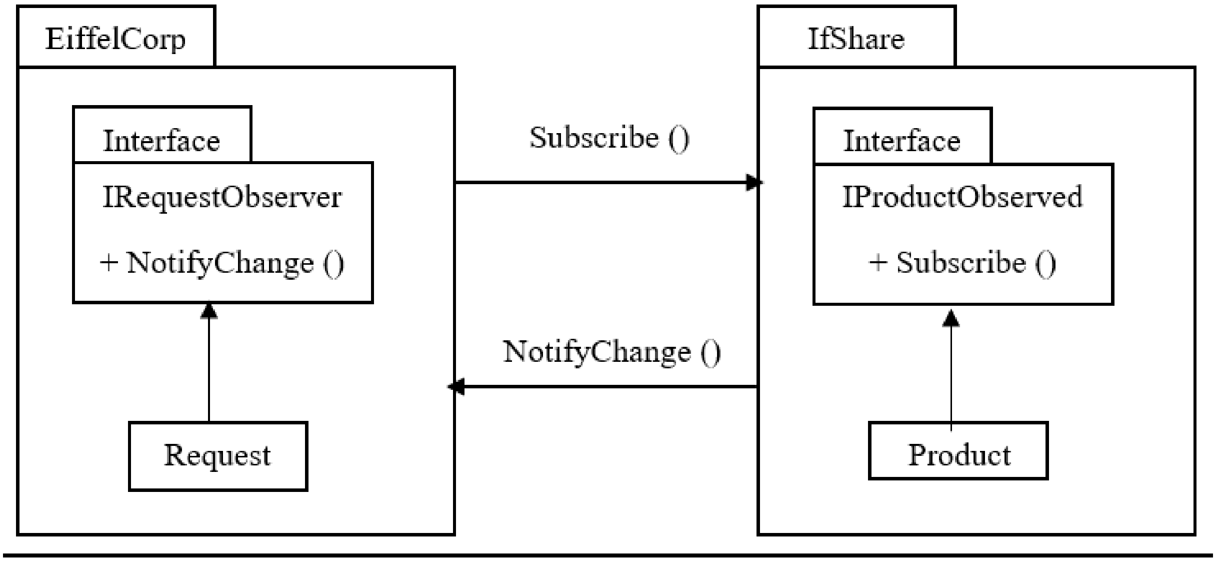
For our case, we choose the passing by value for:

the **employee** of Eiffel Corp to not allow the users of IfShare or employee of Eiffel Corp managing the employee's database.

For **products**, the employees can sale and purchase the products, so the products service will change in client side, we decide to pass the product and other entity by reference.

3. Observer implementation:

To implement an observer pattern, we have the product as observed and the employee as observer, and we have the employee is passed by value, we have created a Request object (passing by reference) in Eiffel Corp server, that contains the id of employee and the id of product, so when an employee requests a product that is not available, we subscribe the request to the product in waiting list(LinkedList in java that manage the list in FIFO), when the products is available we notify the request and the request notify the employee.



IV. Part 2: Web Service

1. Web service:

BuySerivce:

- Convert money to any currency in the world.
- Get all products list and its information.
- Create a customer account and log in.

- Buy a product.
- Create a bank account, deposit and withdraw into it.
- Get the name of employees by id.
- Add and remove products from the shopping card.

V. Encountered problems:

1. The first problem was the design of the project, we didn’t understand well the context.
2. Difficulty in choosing the passing of data: by value or by reference. For the first time we make all passing by reference, after that we understood that the employees cannot have a direct access into the objects.
3. Problem with data, we didn’t use a database so we managed the data with map and list, but the data was not saved, so we define all services like singleton.
4. Some troubles in the setup of web services with complex types like feedback or product and the types of java. The problem was related to the JAX-RPC specification: we were using java.util.List as a data structure to exchange product lists between the web service and its client. Unfortunately, List is not supported by the JAX-RPC 1.1 standard. To solve this problem, we change the List by simple array, and we create complete types for the product, customer and feedback.

VI. Implementation

1. Languages and tools:

In this project we used the following tools:

Java:

Java is an object-oriented programming language created by James Gosling and Patrick Naughton. A peculiarity of Java is that software written in this language is compiled to an intermediate binary representation which can be executed in a Java Virtual Machine (JVM) without regard to the operating system.

RMI Remote Method Invocation:

RMI is a mechanism for calling methods between Java objects running on different virtual machines (separate address spaces), on the same computer or on remote computers connected by a network.

This method makes transparent the access to objects distributed on a network, facilitate the implementation and use of remote Java objects and preserve security (inherent to the Java environment).

The remote objects can be transmitted by value or by reference.

Web Services:

Web services are based on distributed computing technology and provide a standard means of interoperating between different software applications, they comply with several WWW standards, such as Web Services definition language (WSDL) and Simple Object Access Protocol (SOAP).

1.1 SOAP (Simple Object Access Protocol)

SOAP is a protocol for the exchange of structured information in the implementation of web services built on XML. It allows the transmission of messages between remote objects, which means that it allows an object to invoke methods of objects physically located on another server.

WSDL:

WSDL (Web Services Description Language) is a file that specifies what a request message should contain and the appearance of the response message in unambiguous notation. The notation used by a WSDL file to describe message formats is based on the XML Schema standard, which means that WSDL is both programming language and platform neutral.

Web Server : Tomcat:

Apache Tomcat is an open-source Web server tool that allows the implementation of Java Servlets and Java Server Pages (JSP) to promote an effective Java server environment.

Maven:

Maven is a build automation tool used primarily for Java projects. It can also be used to build and manage projects written in other languages. The Maven project is hosted by the Apache Software Foundation.

JSF:

Java Server Faces is a framework that allows the creation of web applications with Java, it’s an Html page with authoring component-based user interfaces on the Java EE platform.

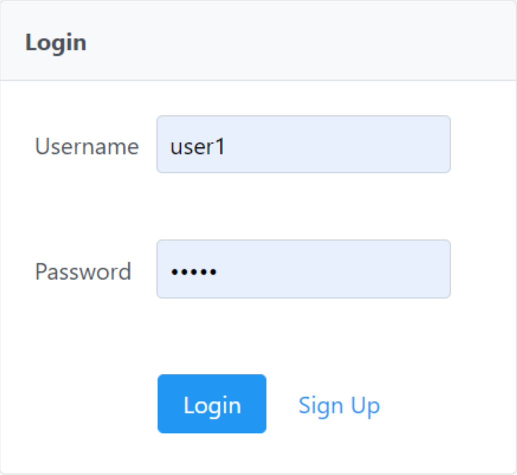
PrimeFaces:

PrimeFaces is an open-source framework for jsf, featuring a lot of components, and created by Turkish company PrimeTek Informatics.

2. Interfaces

Here some interfaces of Eiffel Corp Web application for customers.

Interface Login

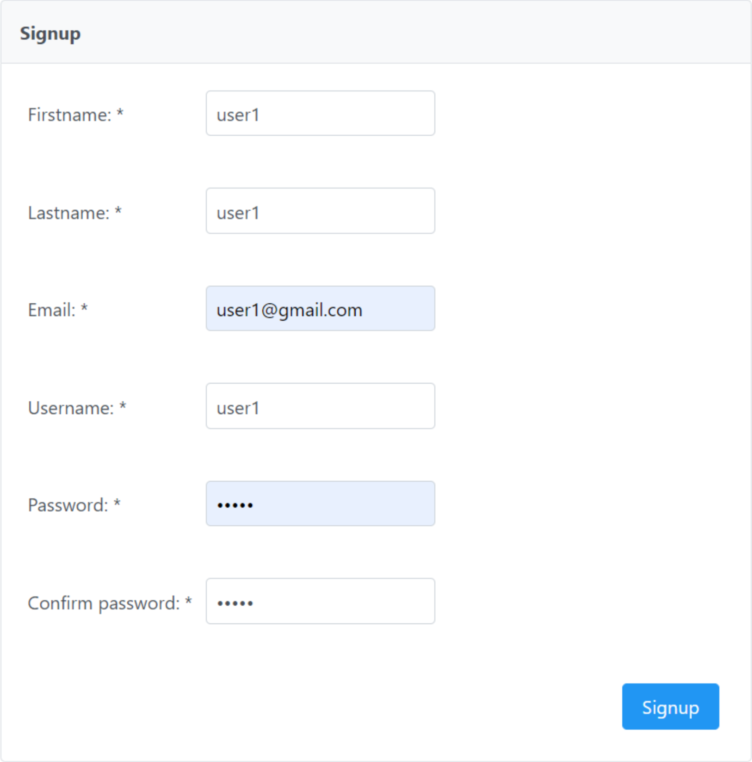


Login

Username: user1

Password: •••••

Interface Signup: When the customer signup, we create automatically a bank account.



Signup

Firstname: * user1

Lastname: * user1

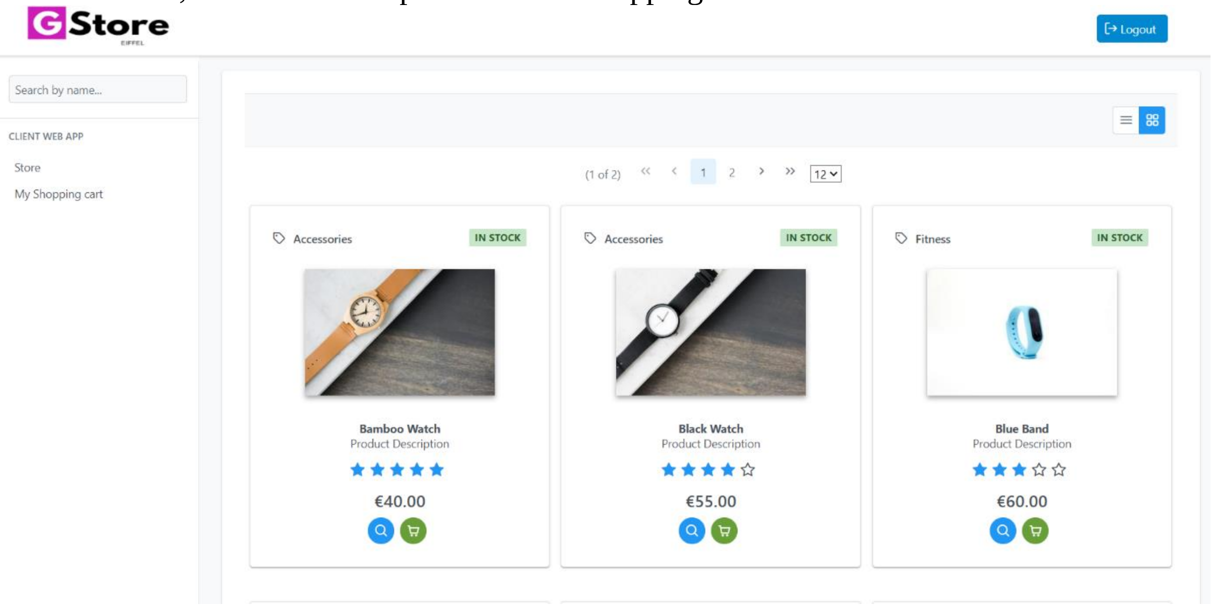
Email: * user1@gmail.com

Username: * user1

Password: * •••••

Confirm password: * •••••

Interface Store: to show the list of products in list or view: the customer can see the availability of product, the category, the total rate, the price, and can click to the loop to see the details and the comments, also he can add product to the shopping cart.

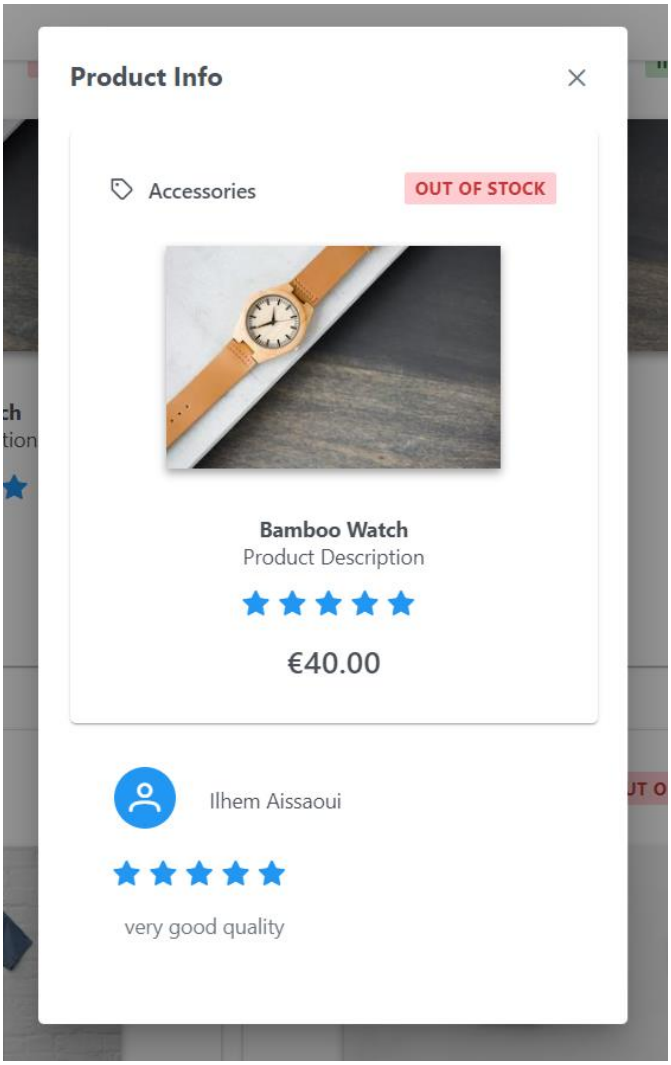


Accessories IN STOCK
Bamboo Watch
Product Description
★★★★★
€40.00

Accessories IN STOCK
Black Watch
Product Description
★★★★☆
€55.00

Fitness IN STOCK
Blue Band
Product Description
★★★☆☆
€60.00

Dialog to show the product details and comment:

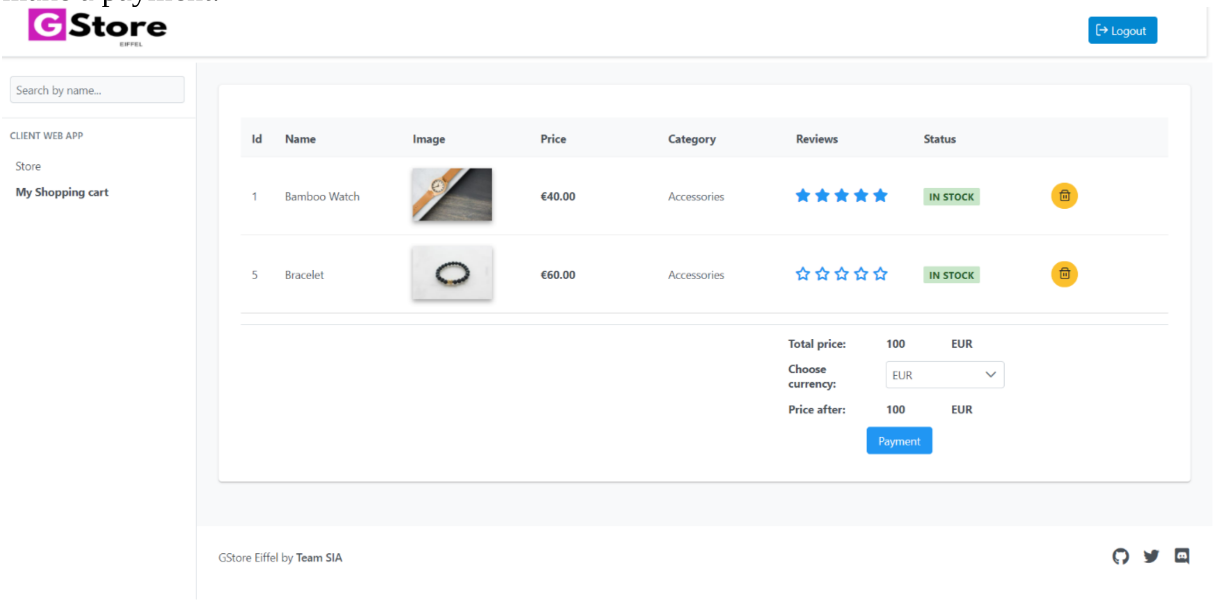


Accessories OUT OF STOCK

Bamboo Watch
Product Description
★★★★★
€40.00

Ilhem Aissaoui
★★★★★
very good quality

Interface of shopping cart: customer can delete product, convert price in his currency and make a payment.



Id	Name	Image	Price	Category	Reviews	Status	
1	Bamboo Watch		€40.00	Accessories	★★★★★	IN STOCK	🗑
5	Bracelet		€60.00	Accessories	☆☆☆☆☆	IN STOCK	🗑

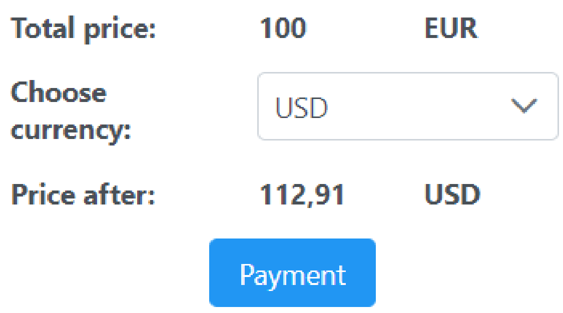
Total price: 100 EUR

Choose currency: EUR

Price after: 100 EUR

GStore Eiffel by Team SIA

The conversion panel; customer can choose the currency:

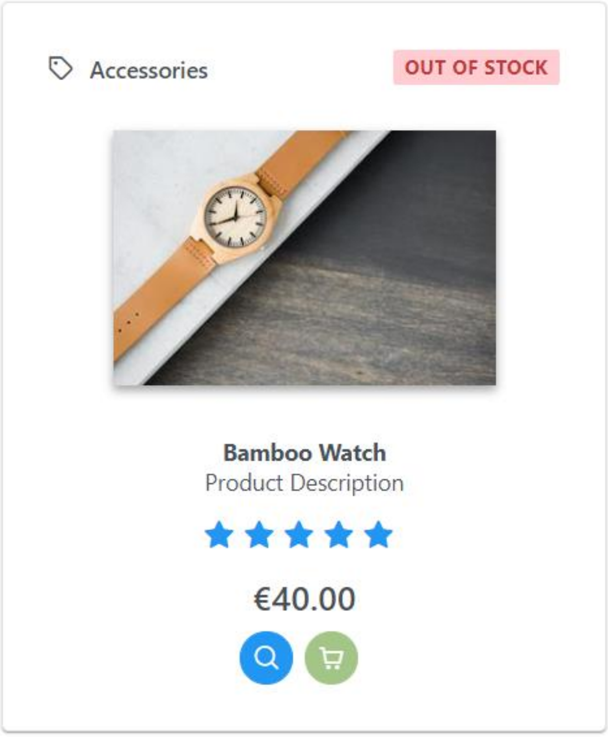


Total price: 100 EUR

Choose currency: USD

Price after: 112,91 USD

After the payment the product will be updated in store and can not be added to shopping cart or purchase it.



Conclusion

In this project, we implemented and developed a java distributed application which allow to Store to buy and Sell products with RMI and web services

The first step was to use RMI to share a service or objects also we needed the Observer in order to get notification externally whenever an action occurs,
RMI was used for internal communication between Eiffel Corp and IfShare, we used both value and reference passage depending on the case. The employees have access to all the services of IfShare remotely.

The second step was to set up a web service for external communication between clients and services of IfShare to support interoperable interaction across a network.