

# WEB SERVICES PROJECT REPORT



**SUPERVISED BY:**

DR. ZARGAYOUNA MAHDI

**ELABORATED BY:**

Arous Souhir

Motcheho Romuald

Bouallagui Ayoub

M2 SIA 2021-2022

## **TABLE OF CONTENT**

<b>Introduction</b>	<b>3</b>
<b>I. Problem Definition</b>	<b>4</b>
<b>II. Implementation and Architecture</b>	<b>5</b>
<b>III. Graphic Interfaces</b>	<b>13</b>
<b>IV. Scenarios</b>	<b>22</b>
<b>V. Conclusion</b>	<b>27</b>

## Introduction

Essentially, web services include any software, application, or cloud technology that provides standardized web protocols (HTTP or HTTPS) to interoperate, communicate, and exchange data messaging – usually XML (Extensible Markup Language) – throughout the internet.

In other words, They are XML-centered data exchange systems that use the internet for A2A (application-to-application) communication and interfacing. These processes involve programs, messages, documents, and/or objects.

Using them will ease the constraints of time, cost, and space for every operation needed. Therefore, applications implementing Web Services have a huge lead on those who don't in terms of organization and optimisation.

Web services offer different benefits across business operations. The technology helps IT pros and web architects streamline connectivity by minimizing development time. And with this simplified infrastructure, company executives begin to see higher ROI (return on investment).

In many B2B operations where both parties understand how the process works, web services provide efficient technology distribution throughout an entire network and companies implementing web services are showing promising results in an easier and faster way than their competitors.

## I. Problem Definition:

The company Eiffel Corp. has just acquired IfShare, another 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.

We are going to be in charge of the design and implementation of a distributed Java application to manage its service, based on Java RMI. The platform should meet the following requirements:

- 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 will 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.

## II. Implementation and Architecture:

To make this project come true, we used a lot of technologies:

- **JAVA 8:**

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.



- **JSF:**

JavaServer Faces (JSF) is a new standard Java framework for building Web applications. It simplifies development by providing a component-centric approach to developing Java Web user interfaces. JavaServer Faces also appeals to a diverse audience of Java/Web developers.



- **Tomcat Server 9.0:**

Essentially it's an open-source Java servlet and Java Server Page container that lets developers implement an array of enterprise Java applications.



- **Maven:**

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of **Maven** information.

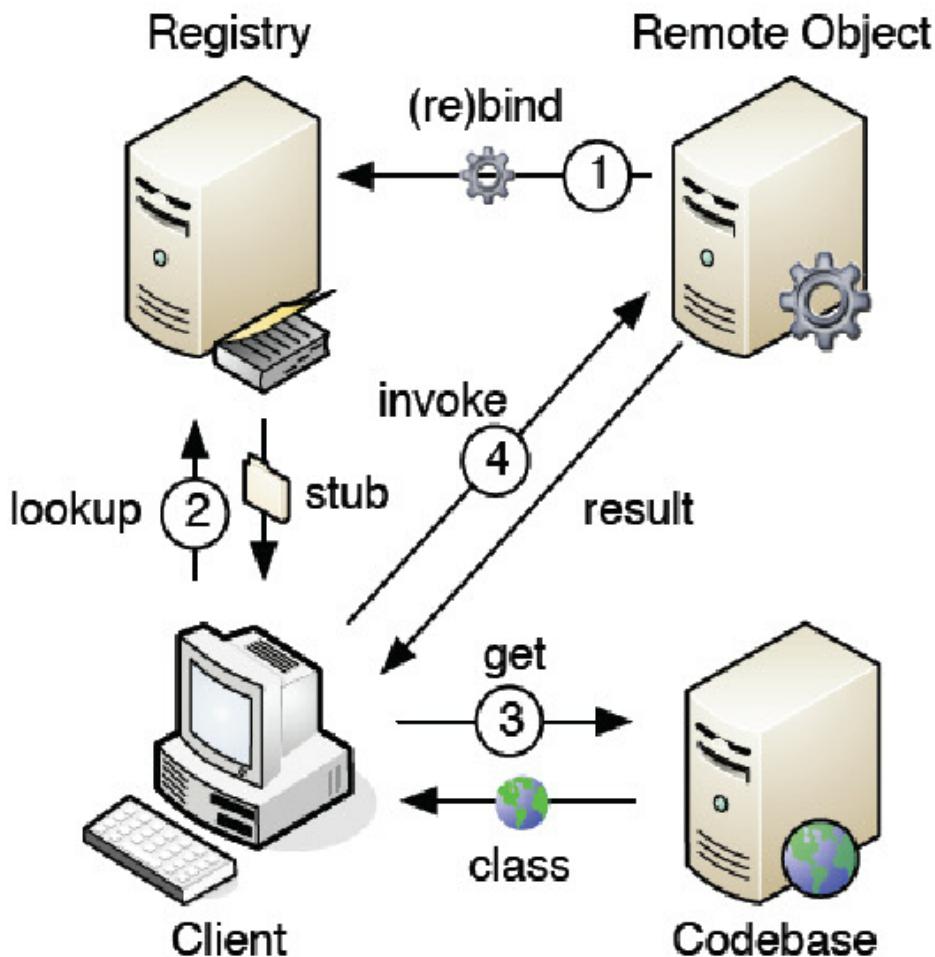


- **RMI:**

RMI stands for Remote Method Invocation. It is a mechanism that allows an object residing in one system (JVM) to access/invoke an object running on another JVM. It is used to build distributed applications; it provides remote communication between Java programs.



An RMI application consists of a server interface, a server implementation, a server skeleton and a client stub, and a client implementation. The server implementation creates remote objects that conform to the server interface. These objects are available for method invocation to clients.



- **Web Services:**

A web service is any piece of software that makes itself available over the internet and uses a standardized XML messaging system. XML is used to encode all communications to a web service. For example, a client invokes a web service by sending an XML message, then waits for a corresponding XML response.



- **WSDL (Web Service Description Language):**

The **Web Services Description Language (WSDL)** is an XML-based interface description language that is used for describing the functionality offered by a web service. The acronym is also used for any specific WSDL description of a web service (also referred to as a *WSDL file*), which provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns. Therefore, its purpose is roughly similar to that of a type signature in a programming language.

- **SOAP** (Simple Object Access Protocol):

SOAP ( Simple Object Access Protocol) is a message protocol that allows distributed elements of an application to communicate. SOAP can be carried over a variety of lower-level protocols, including the web-related Hypertext Transfer Protocol ([HTTP](#)). SOAP defines a header structure that identifies the actions that various SOAP nodes are expected to take on the message, in addition to a payload structure for carrying information. The concept of routing a message through a string of nodes that perform different functions is how SOAP supports things like addressing, security and format-independence.



- **REST** (Representational State Transfer)

REST is an acronym for REpresentational State Transfer and an architectural style for distributed hypermedia systems. Like other architectural styles, REST has its guiding principles and constraints. These principles must be satisfied if a service interface needs to be referred to as RESTful.



- **MVC** (Model - View - Controller)

Model-view-controller (MVC) is a software design pattern commonly used for developing user interfaces that divide the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.

- **Bootstrap**

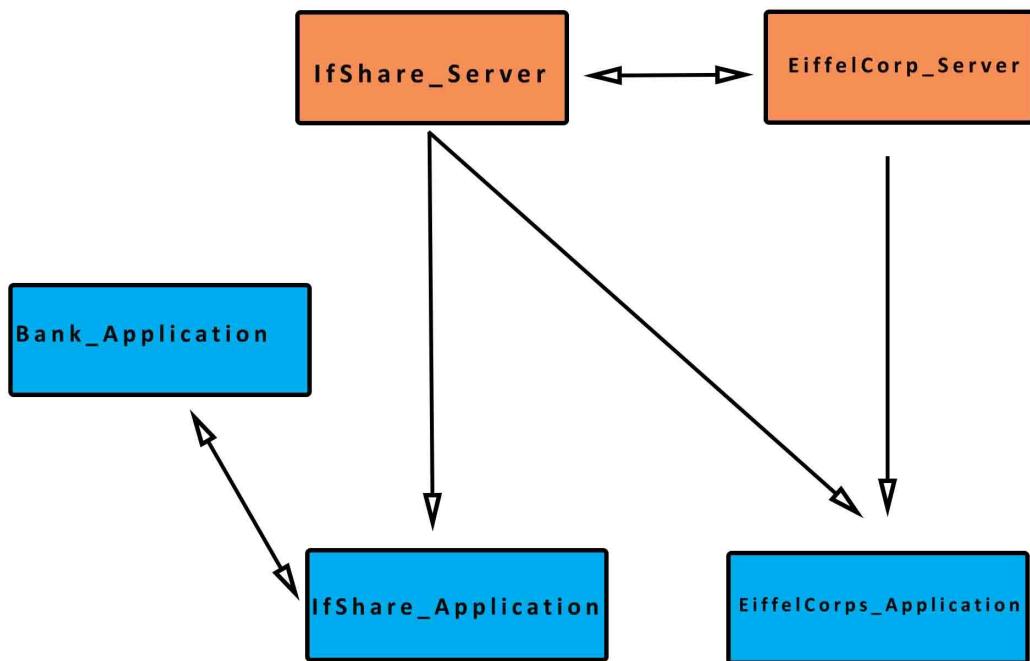
Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.



## Project conception

This project is divided into two parts linked together:

- In the first part, we focused on the implementation of RMI to ensure e-commerce operations for EiffelCorp's Employees.
- In the second part, we used Web Services in order to allow the outside world to buy products that are available and sold at least once.



## PART1 : Implementing JAVA RMI

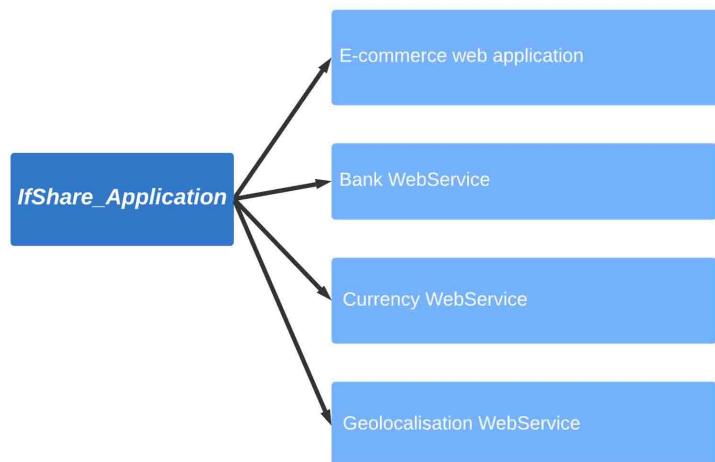
For the main part of this project, our goal is to implement a distributed Java application based on RMI in order to manage sales product operations, as well as allowing employees to leave a rating on the products they returned.

Therefore, this RMI-based application manages both the products database and the Employees database and for the products, they are bought and sold by all Eiffel Corp.

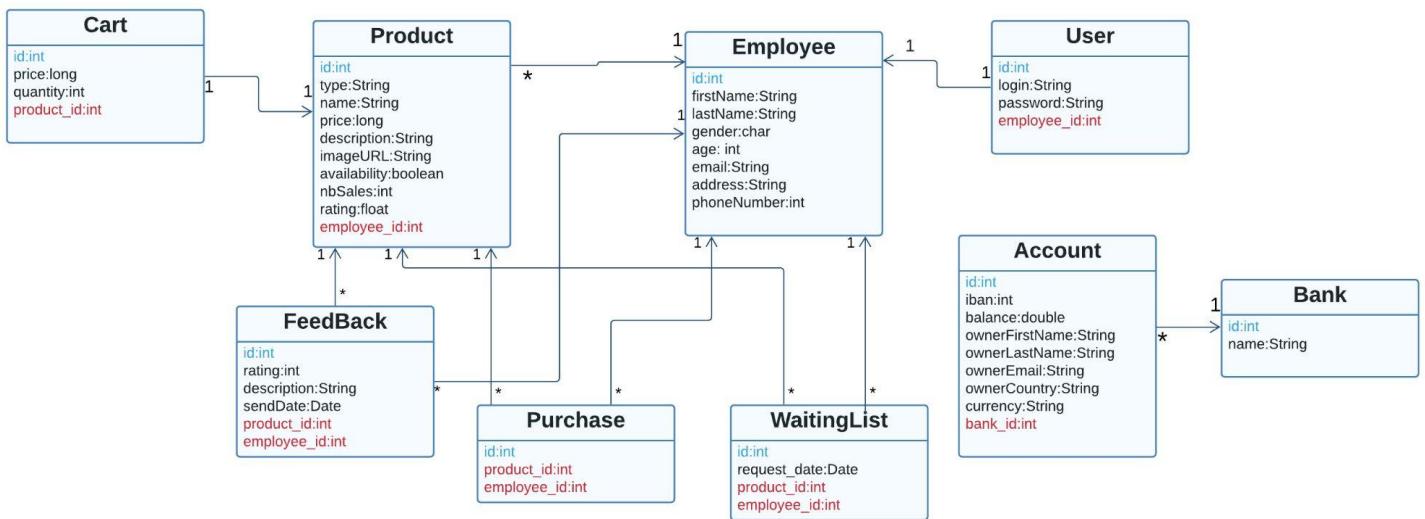
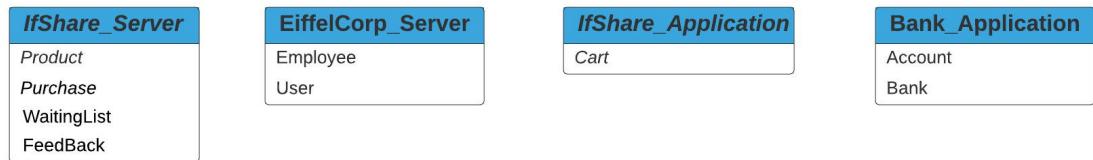
Employees can add notes about the products and their status upon resale. The application managing the product database and the one managing the employees shall run in two different JVMs.

## PART2 : Implementing Web Services

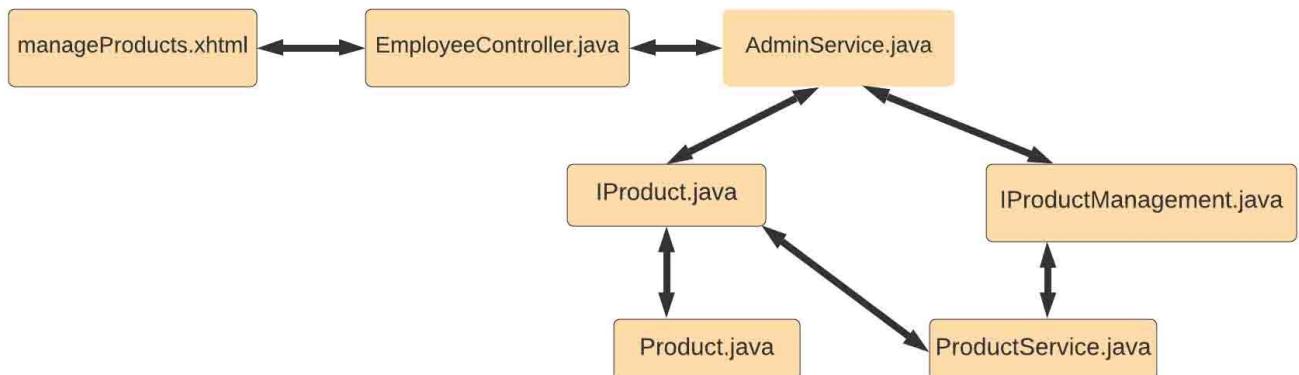
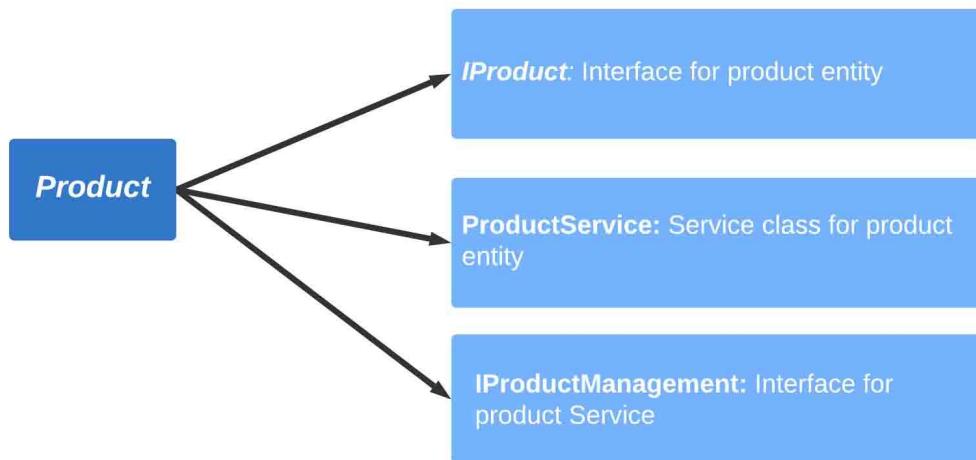
For the second part of this project, the aim was to make the application available to the outside world, and therefore, outside Customers, allowing them to buy products that have been bought previously. The use of Web Services revolves around consulting the available products, adding them into a basket and purchasing them, checking the availability of the funds of the buyer and if they're able to complete the operation no matter what currency their bank account is on, using a real-time currency converter service. For that we have a Bank Application to manage users accounts and IfShare Application that is composed of an e-commerce application to buy products, a bank webService for payment transactions related to bank application, a currency webService to convert the currency within the user's country and a geolocalisation webService to get the user's country.



Here are figures to see the entities we implemented for our projects. First, the entities in each project. Second, the class diagram using the UML Modelling, it describes all the entities, their attributes and the relations that connect them.



Here is an example to understand how the communication between classes and interfaces is done.



### PART3 : External API's

For IfShare Application and Bank Application, the prices of the products are in Euros, but the application 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.

For this purpose we used two external API's :

API for currency conversion :

<https://manage.exchangeratesapi.io/usage>

The screenshot shows the Exchangeratesapi.io control panel. At the top, there's a navigation bar with links for Pricing, Documentation, FAQ, Blog, Status, and Dashboard. The user is logged in as Marnissi Skander, with options to Sign Out. The main content area is titled "Control Panel - API Usage". It features a sidebar with links for Dashboard, Upgrade, Subscription Plan, Account, Payment, API Usage, Other APIs, and Sign Out. The main panel displays "This Month's API Usage:" with a progress bar for API Requests at 37 / 1,000 (3%). Below this is a table for Dec 2021 showing daily request volumes.

Date	Requests
2021-12-01	11
2021-12-02	6
2021-12-03	20

At the bottom, there are sections for Lookup APIs (ipstack, ipapi), Verification APIs (mailboxlayer, numverify), Data APIs (currencylayer, fixer), and Conversion APIs (screenshotlayer, pdflayer). The Exchangeratesapi.io logo is in the bottom right.

## API for geolocalisation :

<https://ipstack.com/usage>

The screenshot shows the ipstack control panel. The layout is similar to the Exchangeratesapi.io dashboard, with a navigation bar, user login information, and a main content area for API usage. The sidebar includes links for Dashboard, Upgrade, Subscription Plan, Account, Payment, API Usage, Other APIs, and Sign Out. The main panel shows "This Month's API Usage:" with a progress bar for API Requests at 21 / 100 (21%). A table for Dec 2021 provides daily breakdowns.

Date	Requests
2021-12-01	5
2021-12-02	6
2021-12-03	10

At the bottom, there are sections for Lookup APIs (userstack, languagelayer), Verification APIs (mailboxlayer, numverify), Data APIs (currencylayer, fixer), and Conversion APIs (screenshotlayer, pdflayer). The ipstack logo is in the bottom right.

### III. Graphic interfaces:

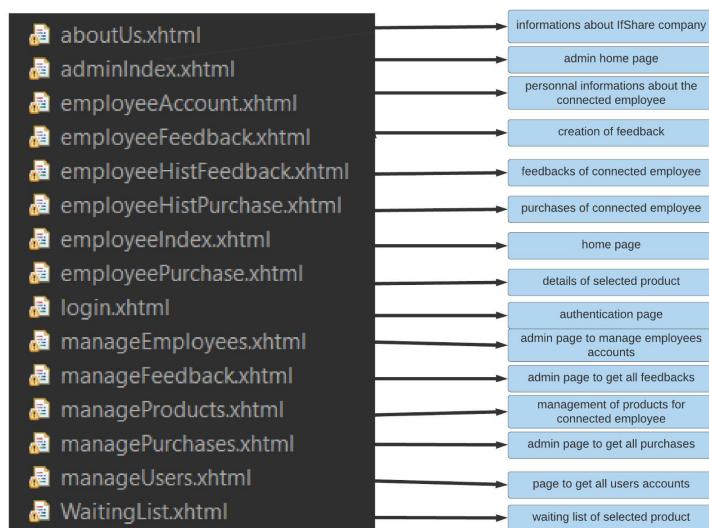
#### Design :

First thing first, we took in consideration the design part of the application since it is the first thing the user sees and we wanted our application to be user-friendly. Thus we created a logo for the IfShare company and chose attractive colors.

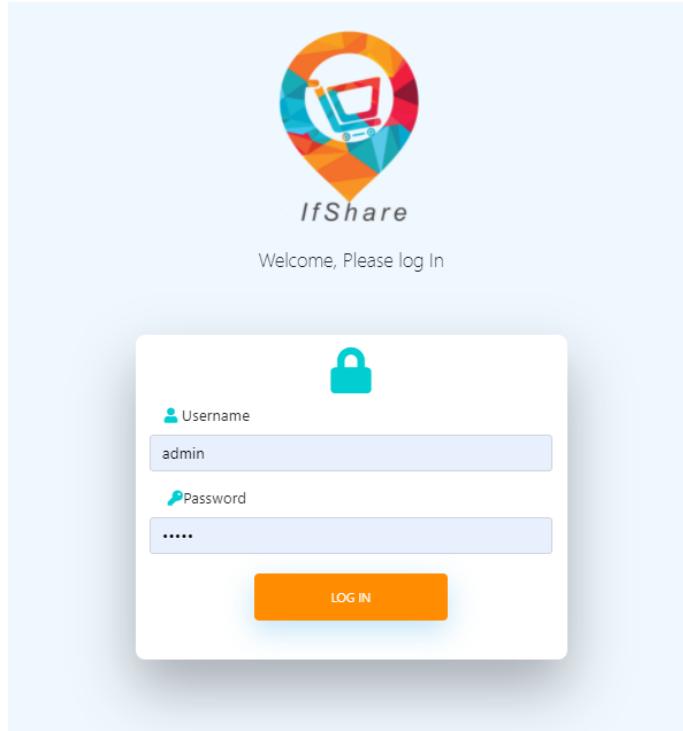


#### EiffelCorp Application :

This is the application implemented exclusively for Eiffel Corp internal employees. We implemented two parts, one for the admin, the second for the employees. These are all the interfaces implemented for EiffelCorp\_Application :



First step is the authentication :



This is the login screen, you have to enter your username and password and then you'll be redirected to a platform depending on whether you are an admin or an employee.

For the admin :

The image displays the admin platform interface. On the left side, there is a vertical sidebar with a navigation menu. The menu items include "Manage Employees" (with a user icon), "Purchases" (with a shopping cart icon), "FeedBacks" (with a star icon), "Users Accounts" (with a person icon), and "About Us" (with a question mark icon). In the center of the page, there is a large black gear icon. Below the icon, the text "Welcome to the admin platform" is centered, followed by a smaller line of text: "Check the application updates and manage EiffelCorp's employees accounts".

Here the admin can manage the employees, see all products, feedbacks and user accounts as you can see on the side bar. Here are some of the application's interfaces :



Phone Number

First Name      Last Name

Email

Address

Gender

 Male

Age

## Employees list

Id	First Name	Last Name	Gender	Age	Email	Address	Phone Number	Modify	Remove
0	souhir	arous	F	24	souhir.arous@esprit.tn	2 Allee Des Charmilles	767800359		
1	romuald	motcheho	M	25	romuald.motcheho@esprit.tn	2 Allee Des Charmilles	717770449		
2	ayoub	bouallgui	M	24	ayoub.bouallgui@esprit.tn	2 Allee Des Charmilles	742368359		

**Add Employee**

## Purchases list

Id	Product id	Employee id
0	0	1
1	2	1
2	3	1
3	1	1
4	5	1
5	4	1
6	6	1



## Users accounts list

<b>Id</b>	<b>Login</b>	<b>Password</b>	<b>Employee id</b>
0	souhir	souhir	0
1	romuald	romuald	1
2	ayoub	ayoub	2

For a simple employee :

Category	Description	Status	Rating
Dress	Type : Clothes Price : 35 €	Green	0.0★
Book	Type : Books Price : 18 €	Green	0.0★
Kids Game	Type : Kids Price : 39 €	Red	0.0★
Beauty products	Type : Beauty Price : 45 €	Green	0.0★



Name

Description

Image

Aucun fichier choisi

Price

Type

High Tech

### Products list

ID	Name	Description	Type	Price	Rating	Availability	Waiting List	Modify	Remove
0	Dress	New, size S.	Clothes	35€	★2.0	<span style="color:red">●</span>	<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value=""/>
1	Book	For kids, In very good shape.	Books	18€	★0.0	<span style="color:green">●</span>	<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value=""/>
4	Beauty products	Make up and products to take care of your skin.	Beauty	45€	★4.0	<span style="color:red">●</span>	<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value=""/>
5	Drawing material	For kids and adults, let your imagination go !	Art	70€	★0.0	<span style="color:green">●</span>	<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value=""/>



### My orders

Product	Type	Price	Give feedback
Beauty products	Beauty	45 €	<input type="button" value=""/>
Dress	Clothes	35 €	<input type="button" value=""/>

### My waiting list

Product	Date
Garden equipment	Fri Dec 03 19:24:31 CET 2021

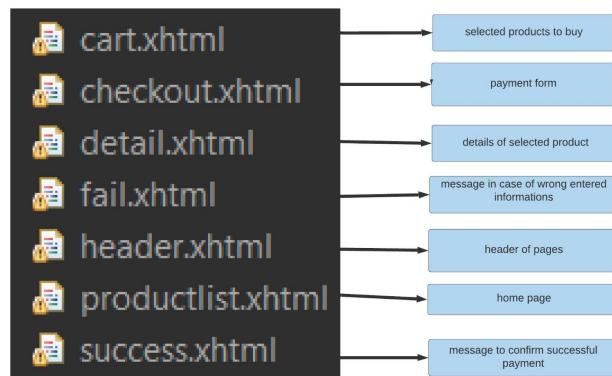


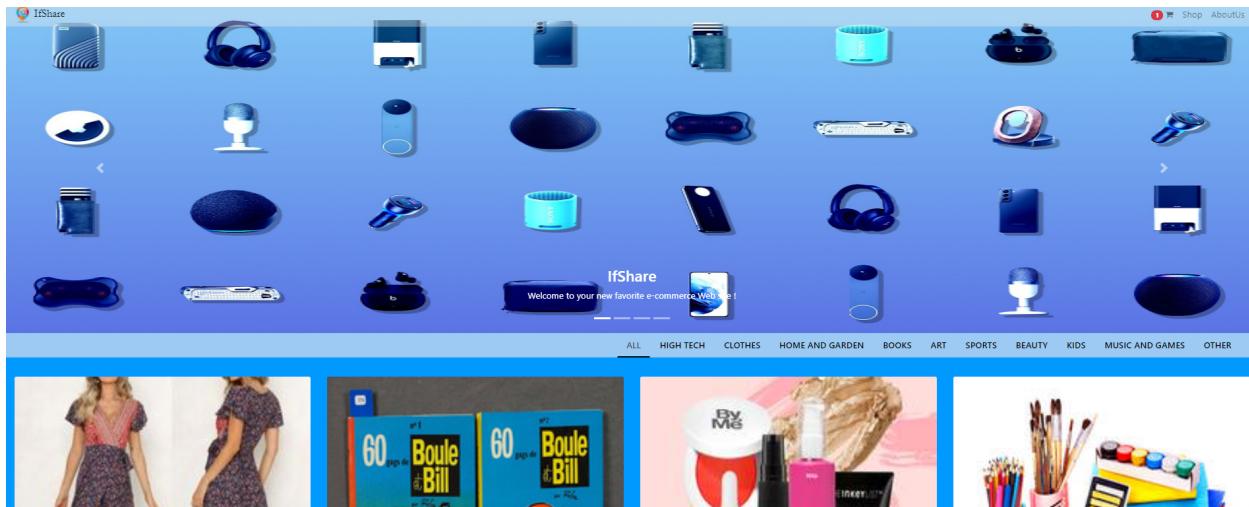
### My feedbacks

Product	Feedback	Rating
Beauty products	Good product.	4
Dress	Size too big, I'd say M not S.	2

### IfShare Application

This is the application implemented for the outside world. Since users don't have accounts, they don't have identifiers, for this reason in this application users don't have purchase history and can't sell products or give feedback. This application is made to able any user around the world to buy, at the very moment, a product that is available and has been sold at least one time in the EiffelCorp Application. These are all the interfaces implemented for EiffelCorp\_Application :





This page shows a similar layout to the first, with a blue header and a grid of product thumbnails. The thumbnails include two dresses, two books, three beauty products, and a set of drawing materials.

This page displays a detailed view of a "Beauty products" item. It includes a thumbnail image of several skincare bottles, a title, a description ("Make up and products to take care of your skin."), and a table with type and price information. Buttons for "Back to shop" and "Add to cart" are at the bottom.



## Last step !

Please fill the form below and be sure that the given informations are correct

First name

Last name

Email

Address

Country

 AFGHANISTAN

State

Zip

### Your Cart

1

Drawing material x 1	70.0 €
Total EUR	70.0 €

## Payment

IBAN

Account

owner

Lastname

Account

owner

Firstname

Checkout

## Bank Application

This is the Bank Web Service we will be using for real-time currency conversion. It revolves around a database of bank accounts from different currencies (depending on the country of the owner of the account).

Here is the one and only interface of the application, you can add an account and set its balance :



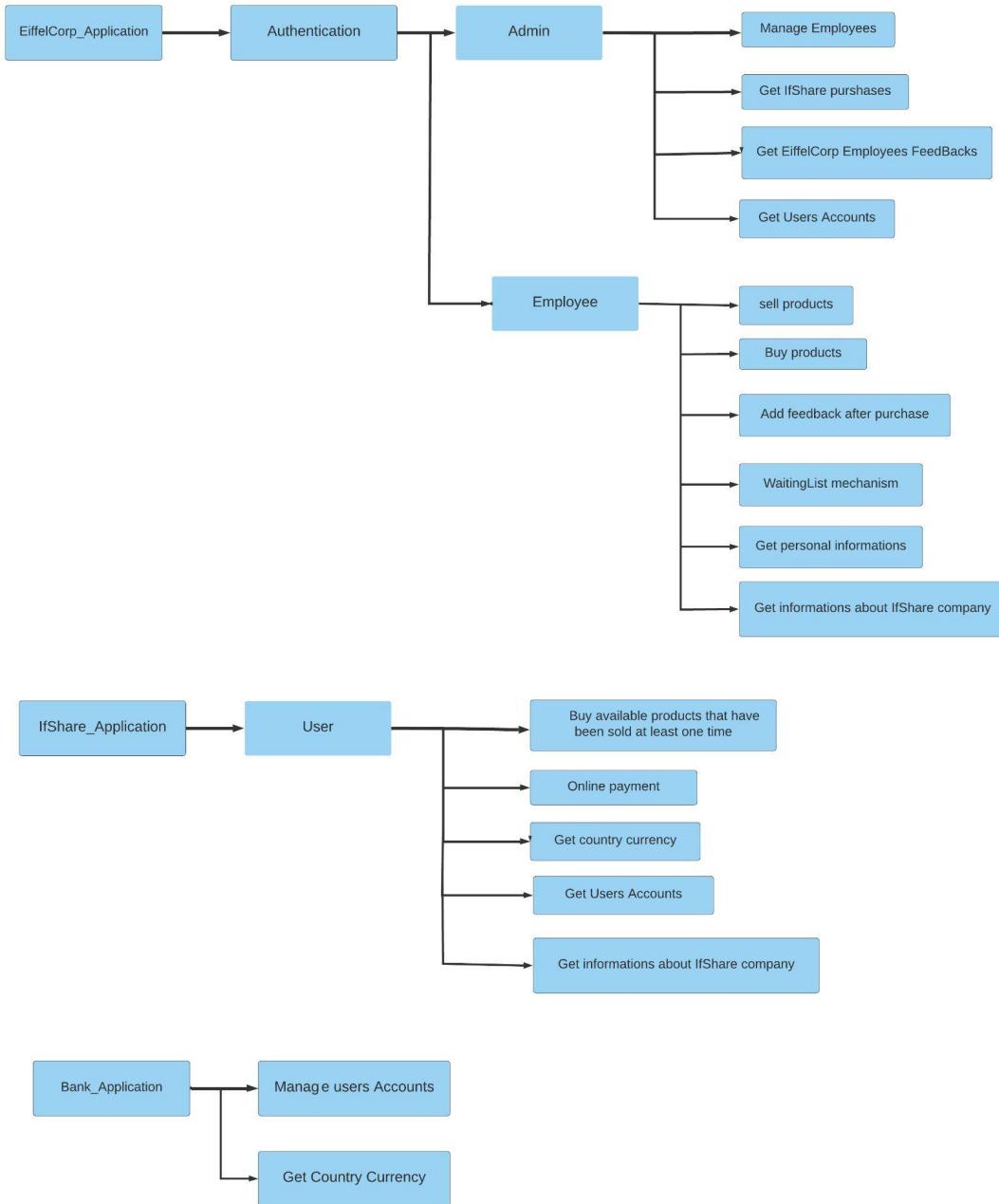
The screenshot shows the "IfShare Bank Web Service" interface. At the top, there is a blue header bar with the service name and a logo. Below the header, there is a logo for "IfShare" featuring a stylized orange and red design. The main title "Bank Web Service" is centered above a subtitle "Web page to manage bank accounts". On the left, there is a form for adding a new account, including fields for IBAN (1333), First name (Ayoub), Last name (Bouallagui), and Email (ayoub.bouallagui@gmail.com). Below this form is a table titled "Account List" showing three existing bank accounts:

IBAN	Country	First Name	Last Name	Email	Bank	Balance	Currency	Modify	Remove
1111	TUN	Souhir	Arous	aroussouhir@gmail.com	BIAT	2000.0	TND	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1222	FR	Romuald	Motcheho	romuald.motcheho@esprit.tn	S.T.B	15000.0	EUR	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1333	TUN	Ayoub	Bouallagui	ayoub.bouallagui@gmail.com	BIAT	88000.0	TND	<input checked="" type="checkbox"/>	<input type="checkbox"/>

At the bottom of the page, there is a green button labeled "Add Account".

## IV. Scenarios:

Here is a figure to sum up all the features of our three applications:



## Eiffel Corp Application

The admin adds the employees, then the employees can buy and sell products. Once An employee buys a product, he can add feedback and a rating. When a product is sold, its availability becomes false.

here is an example of an employee who want to buy a product that is unavailable :

The screenshot shows a product page for a device labeled 'high tech'. On the left, there are two images of the device: a front view and a back view. Below the images is a red circle highlighting the text 'No' under the 'Available' status in the product details on the right. The product details include:

- Type : High Tech
- Price : 99 €
- Rating : 5.0 ★

**Feedbacks**

arous souhir  
Date : Fri Dec 03 19:37:17 CET 2021  
Rating : 5  
Very good product.

**Product Details (Right Side):**

high tech	High Tech
Description :	New.
Type :	High Tech
Available :	No
Price :	99 €
Rating :	5.0

[← Back to products list](#) [Buy now](#)

Here the product 'High Tech' is unavailable, if an employee presses the button 'Buy now' he will be added to the waiting list.



### My orders

Product	Type	Price	Give feedback

### My waiting list

Product	Date
high tech	Fri Dec 03 19:39:49 CET 2021

In this situation, the employee will get the product once the product owner changes the availability state and gets to his turn in the waiting list. Here is the product owner side :

## Products list

<b>Id</b>	<b>Name</b>	<b>Description</b>	<b>Type</b>	<b>Price</b>	<b>Rating</b>	<b>Availability</b>	<b>Waiting List</b>	<b>Modify</b>	<b>Remove</b>
6	high tech	New.	High Tech	99€	★ 5.0				



## Waiting list

**Change availability**  
1 in the waiting list.



<b>Employee</b>	<b>Date</b>
motcheho romuald	Fri Dec 03 19:39:49 CET 2021



## Waiting list

**Change availability**  
0 in the waiting list.

<b>Employee</b>	<b>Date</b>

## Products list

<b>Id</b>	<b>Name</b>	<b>Description</b>	<b>Type</b>	<b>Price</b>	<b>Rating</b>	<b>Availability</b>	<b>Waiting List</b>	<b>Modify</b>	<b>Remove</b>
6	high tech	New.	High Tech	99€	★ 5.0				



Buyer side :



My orders

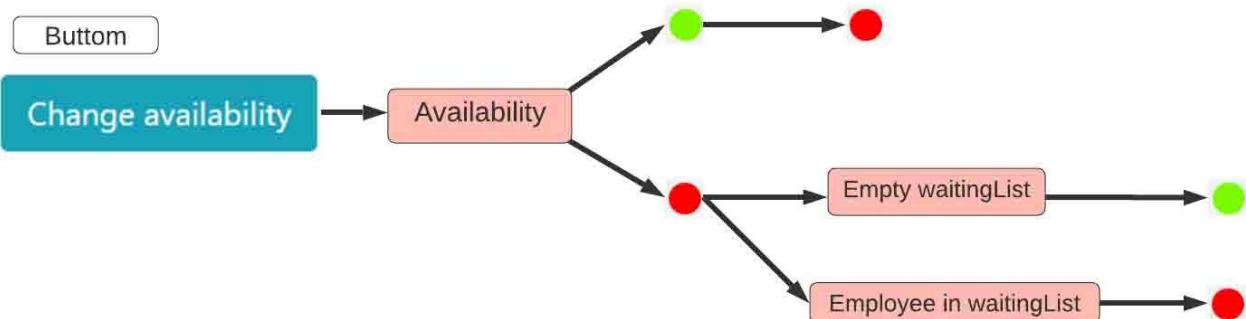
Product	Type	Price	Give feedback
high tech	High Tech	99 €	



My waiting list

Product	Date

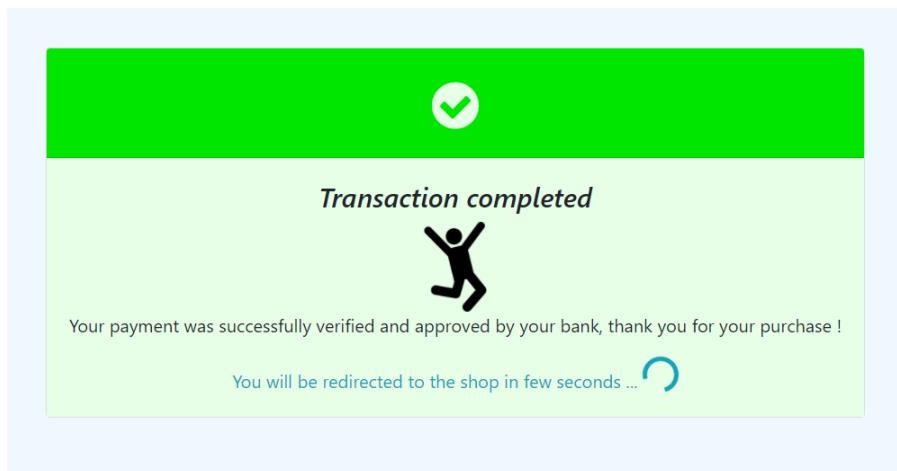
However, if there are more than one employee in the waiting list, we have a mechanism of first-in, first-out. Each employee has to wait for its turn. and as long as the waiting list is not empty, the product remains unavailable.



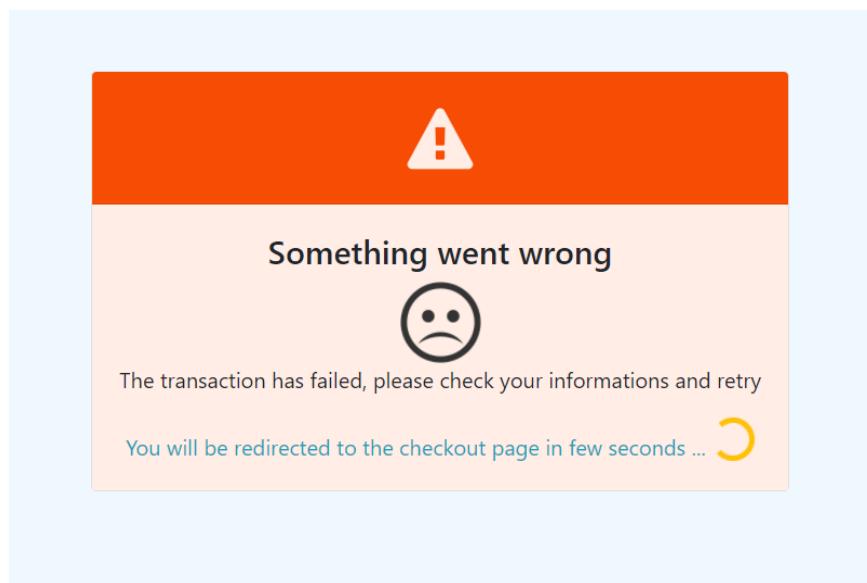
## IfShare Application and Bank Application

First of all, to be able to buy products, we need to have an account in the Bank Application. Then we can go to the platform, choose products and proceed to payment. In this step, we have to fill the checkout form with information on our bank account.

- When you submit, the currency from the bank account is converted to Euros.
  - If the converted amount is bigger than the price of the Product : the operation is a success and you get this message :



- If the converted amount less than the price of the Product or if an entered information is incorrect : the operation is failed and you get this message :



## **Conclusion**

Web Services and RMI were major players in the realization of this project, facilitating the access, use and implementation of JAVA objects and applications distributed on the Internet such as the currency conversion system that we have met. This interoperability is a huge asset that plays an effective role in different operations in different fields as they are excellent for exposing software functionality to customers. Unlike previous architectures, the adoption of web services relies on many important resources, both in development and deployment, although the speed of its interoperability is affected by bandwidth. In fact, Web Services are still stabilizing standards