Your Online Store

Prepared by
Paolo Caruso
Cristian Fioravanti
Anton Volkov

Sapienza University of Rome
Laboratory of Advanced programming
Software Engineering
Prof. Mecella

April 06th 2023

1. Project Idea

1.1. Purpose

The purpose of the project is to build a website, both front-end and back-end, where to sell whatever product the admin wants to sell.

The website provides a way for customers to find, inspect and buy Products with ease, it also learns from a user and provides suggestions to which products are best for the user.

It also provides statistics and other information to the seller to see how a product is selling.

1.2. Intended Audience

The website is for two types of users:

- **Buyers** who are looking around to buy new products
- **Sellers** who wants to sell products

2. Architectural style and functionalities

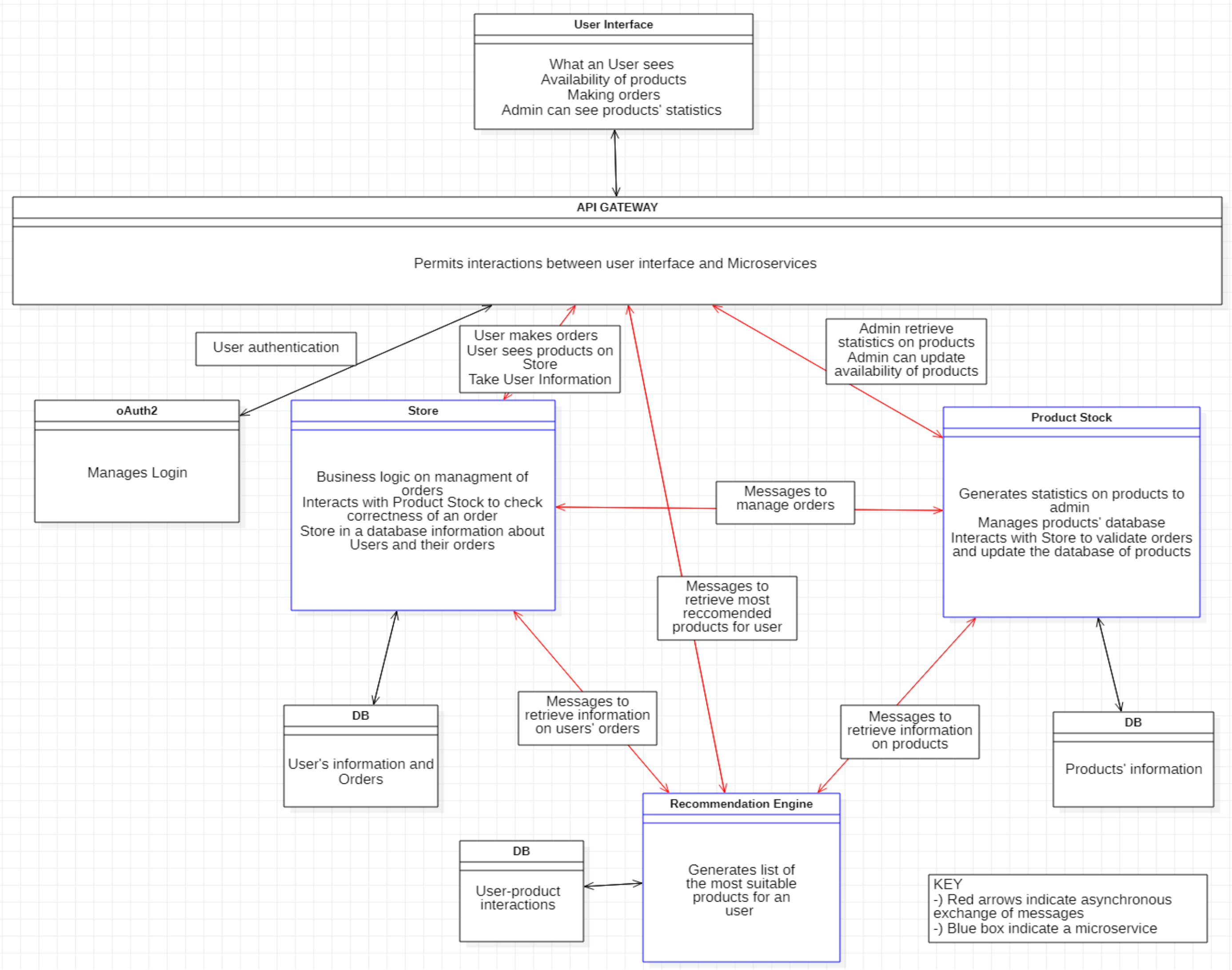
2.1. Architecture

The project has been built like a layered application based on a microservice architecture.

In particular, the service is composed of:

- A **front-end** built with HTML, JS and CSS.
- A **back-end** divided in multiple microservices in custom containers that communicate each other with **asynchronous messages**

The microservices are made with **Docker Containers** and orchestrated with **Docker Compose**.


User Interface
What an User sees
Availability of products
Making orders
Admin can see products' statistics
API GATEWAY
Permits interactions between user interface and Microservices
User authentication
User makes orders
User sees products on Store
Take User Information
Admin retrieve statistics on products
Admin can update availability of products
oAuth2
Manages Login
Store
Business logic on managment of orders
Interacts with Product Stock to check correctness of an order
Store in a database information about Users and their orders
Messages to manage orders
Product Stock
Generates statistics on products to admin
Manages products' database
Interacts with Store to validate orders and update the database of products
Messages to retrieve most reccomended products for user
DB
User's information and Orders
Messages to retrieve information on users' orders
Messages to retrieve information on products
DB
Products' information
Recommendation Engine
Generates list of the most suitable products for an user
DB
User-product interactions
KEY
-) Red arrows indicate asynchronous exchange of messages
-) Blue box indicate a microservice

2.2. Use Cases

The website has different uses:

- **Item buy**
- **Item inspection**
- **Item statistics**

2.3. Microservices and functionalities

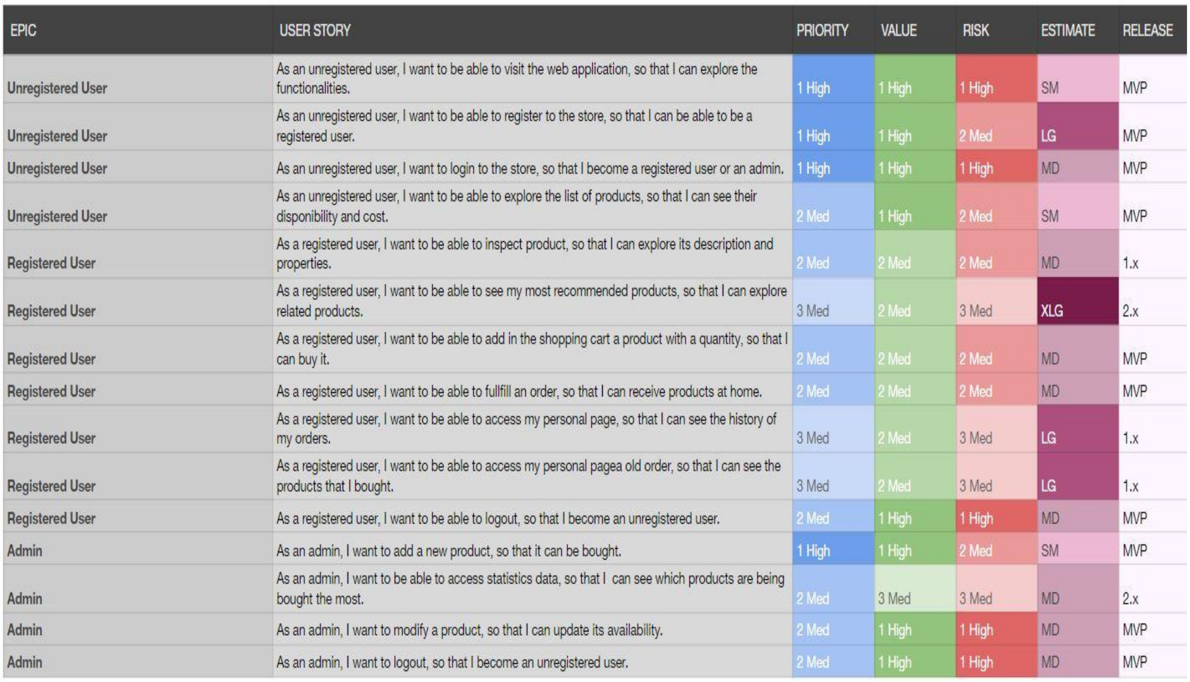
The microservices implemented are:

- **Store:** implements store logic and orders' management
- **Product stock:** implements products' management
- **Recommendation engine:** provides recommendation to application's users
- **Databases**

Chapter 3

3. Requirements

3.1 User Stories



EPIC	USER STORY	PRIORITY	VALUE	RISK	ESTIMATE	RELEASE
Unregistered User	As an unregistered user, I want to be able to visit the web application, so that I can explore the functionalities.	1 High	1 High	1 High	SM	MVP
Unregistered User	As an unregistered user, I want to be able to register to the store, so that I can be able to be a registered user.	1 High	1 High	2 Med	LG	MVP
Unregistered User	As an unregistered user, I want to login to the store, so that I become a registered user or an admin.	1 High	1 High	1 High	MD	MVP
Unregistered User	As an unregistered user, I want to be able to explore the list of products, so that I can see their disponibility and cost.	2 Med	1 High	2 Med	SM	MVP
Registered User	As a registered user, I want to be able to inspect product, so that I can explore its description and properties.	2 Med	2 Med	2 Med	MD	1.x
Registered User	As a registered user, I want to be able to see my most recommended products, so that I can explore related products.	3 Med	2 Med	3 Med	XLG	2.x
Registered User	As a registered user, I want to be able to add in the shopping cart a product with a quantity, so that I can buy it.	2 Med	2 Med	2 Med	MD	MVP
Registered User	As a registered user, I want to be able to fullfill an order, so that I can receive products at home.	2 Med	2 Med	2 Med	MD	MVP
Registered User	As a registered user, I want to be able to access my personal page, so that I can see the history of my orders.	3 Med	2 Med	3 Med	LG	1.x
Registered User	As a registered user, I want to be able to access my personal pagea old order, so that I can see the products that I bought.	3 Med	2 Med	3 Med	LG	1.x
Registered User	As a registered user, I want to be able to logout, so that I become an unregistered user.	2 Med	1 High	1 High	MD	MVP
Admin	As an admin, I want to add a new product, so that it can be bought.	1 High	1 High	2 Med	SM	MVP
Admin	As an admin, I want to be able to access statistics data, so that I can see which products are being bought the most.	2 Med	3 Med	3 Med	MD	2.x
Admin	As an admin, I want to modify a product, so that I can update its availability.	2 Med	1 High	1 High	MD	MVP
Admin	As an admin, I want to logout, so that I become an unregistered user.	2 Med	1 High	1 High	MD	MVP

Figure 2. User Stories

3.2 Lo-Fi Mockup

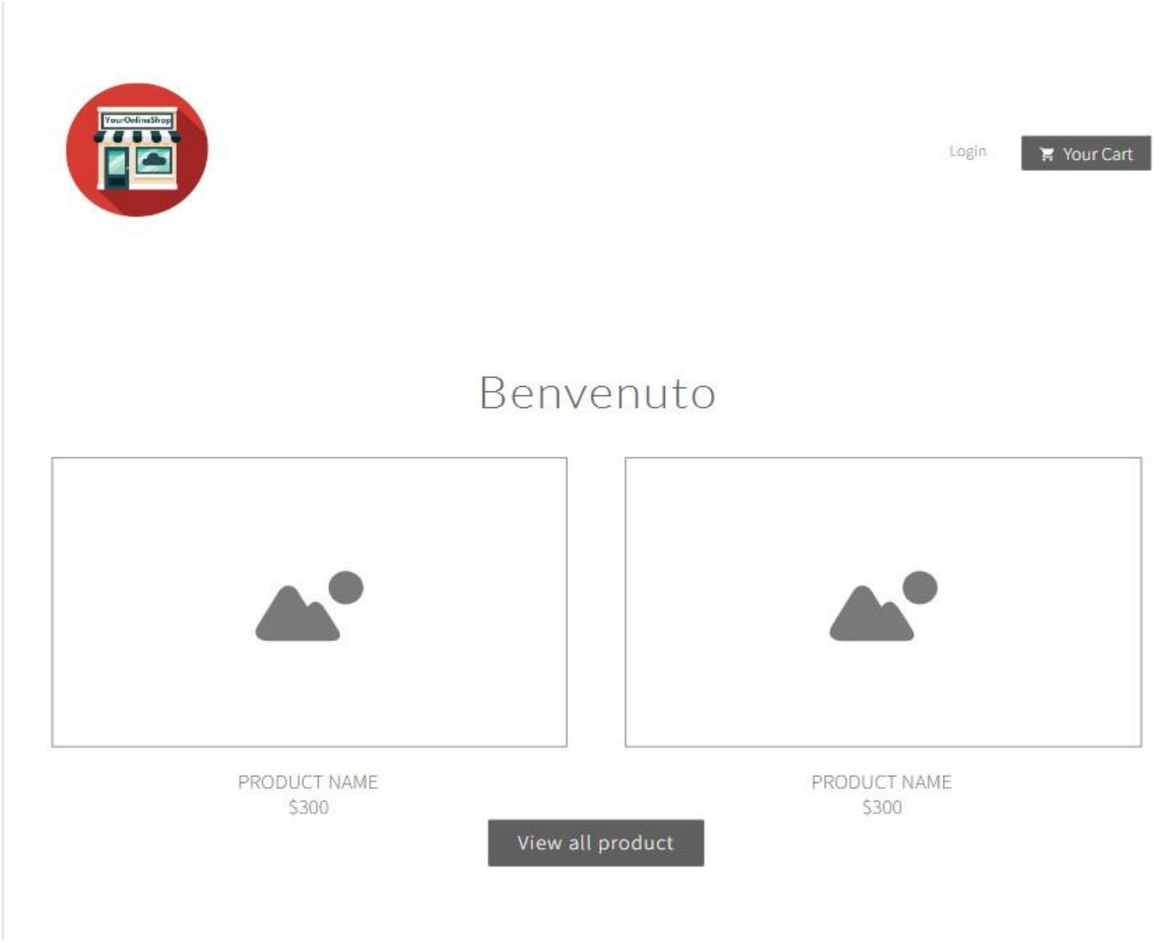
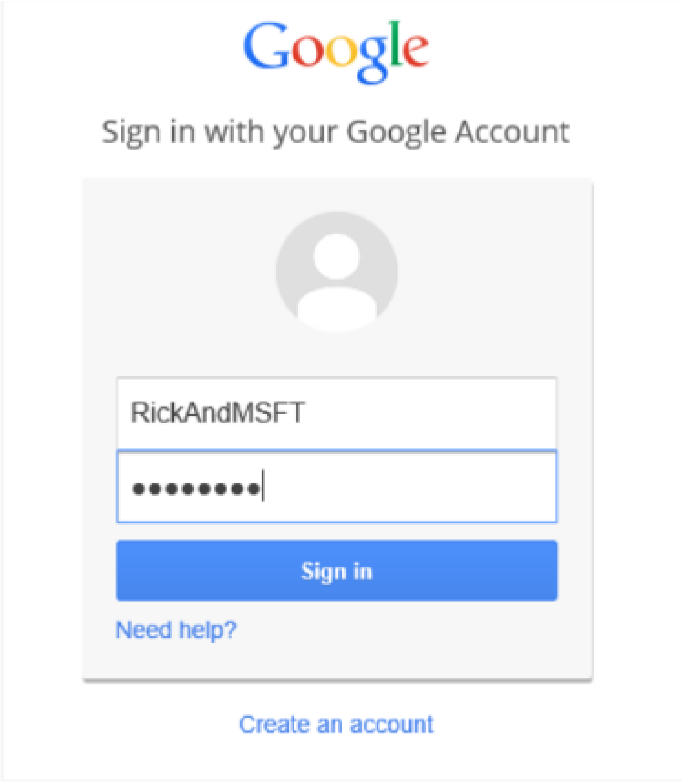


Figure 3. User Story 1



Sign in with your Google Account

RickAndMSFT

••••••••

Figure 4. User Story 2-3

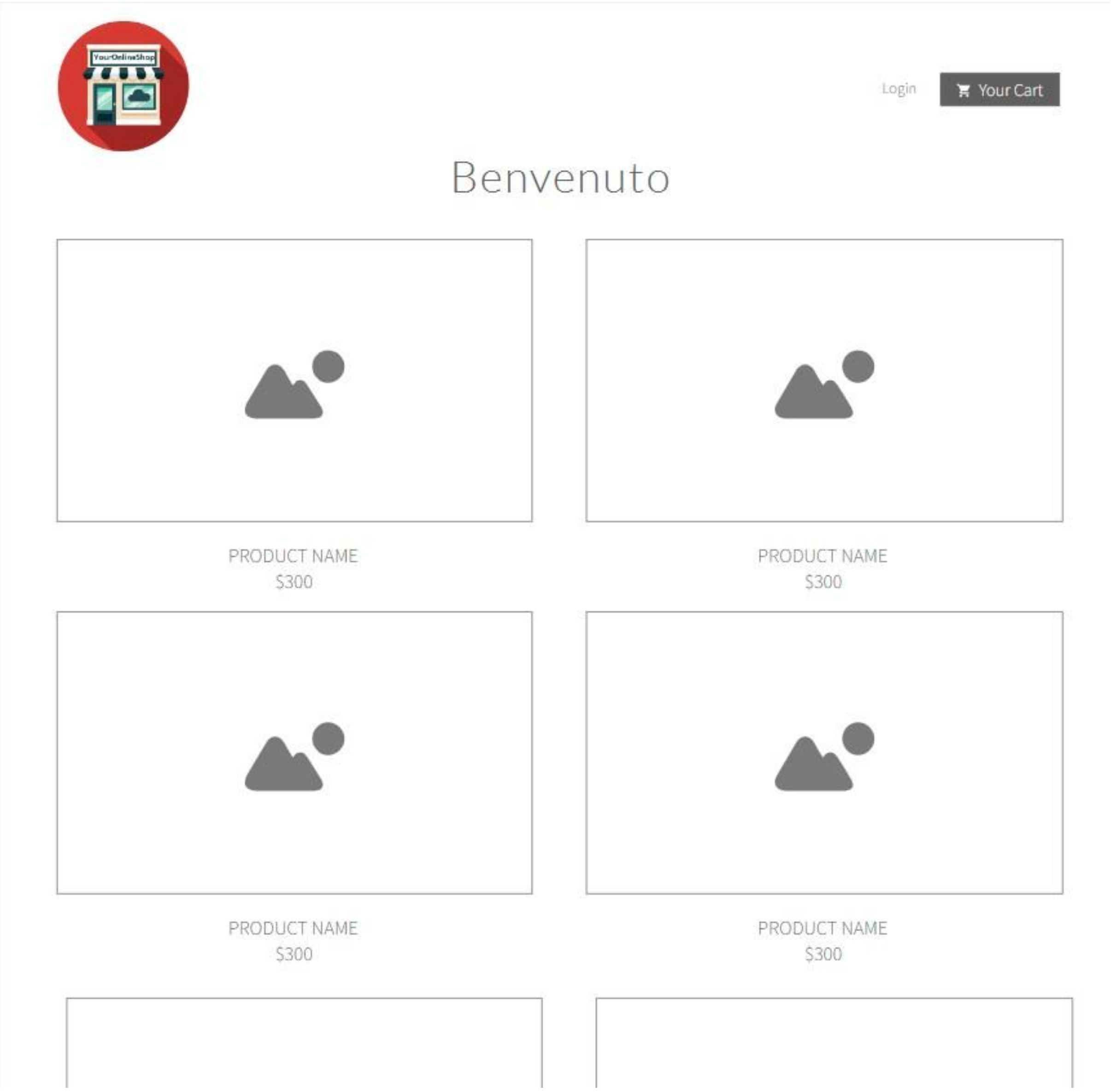
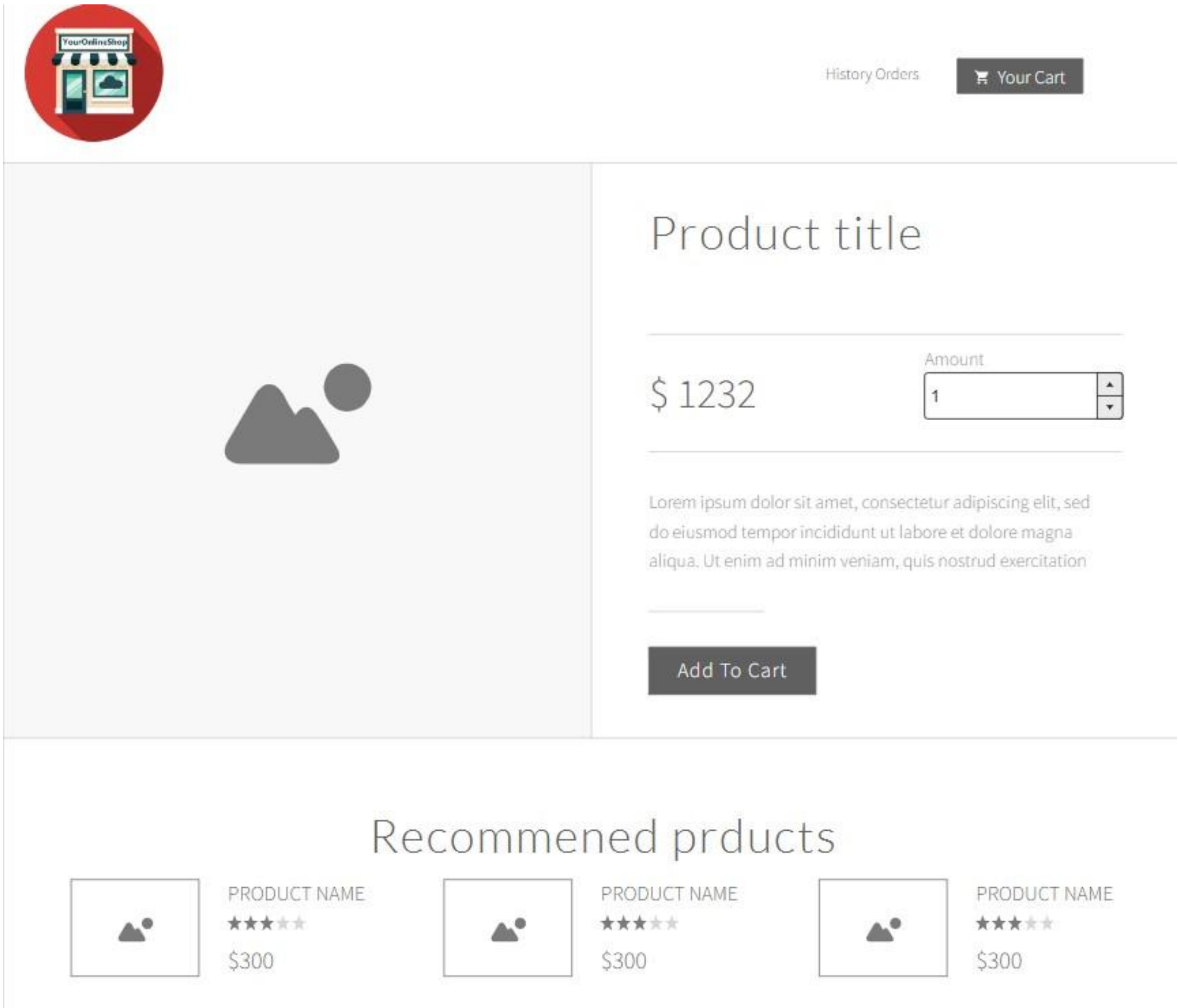


Figure 5. User Story 4



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation

Figure 6. User Story 5

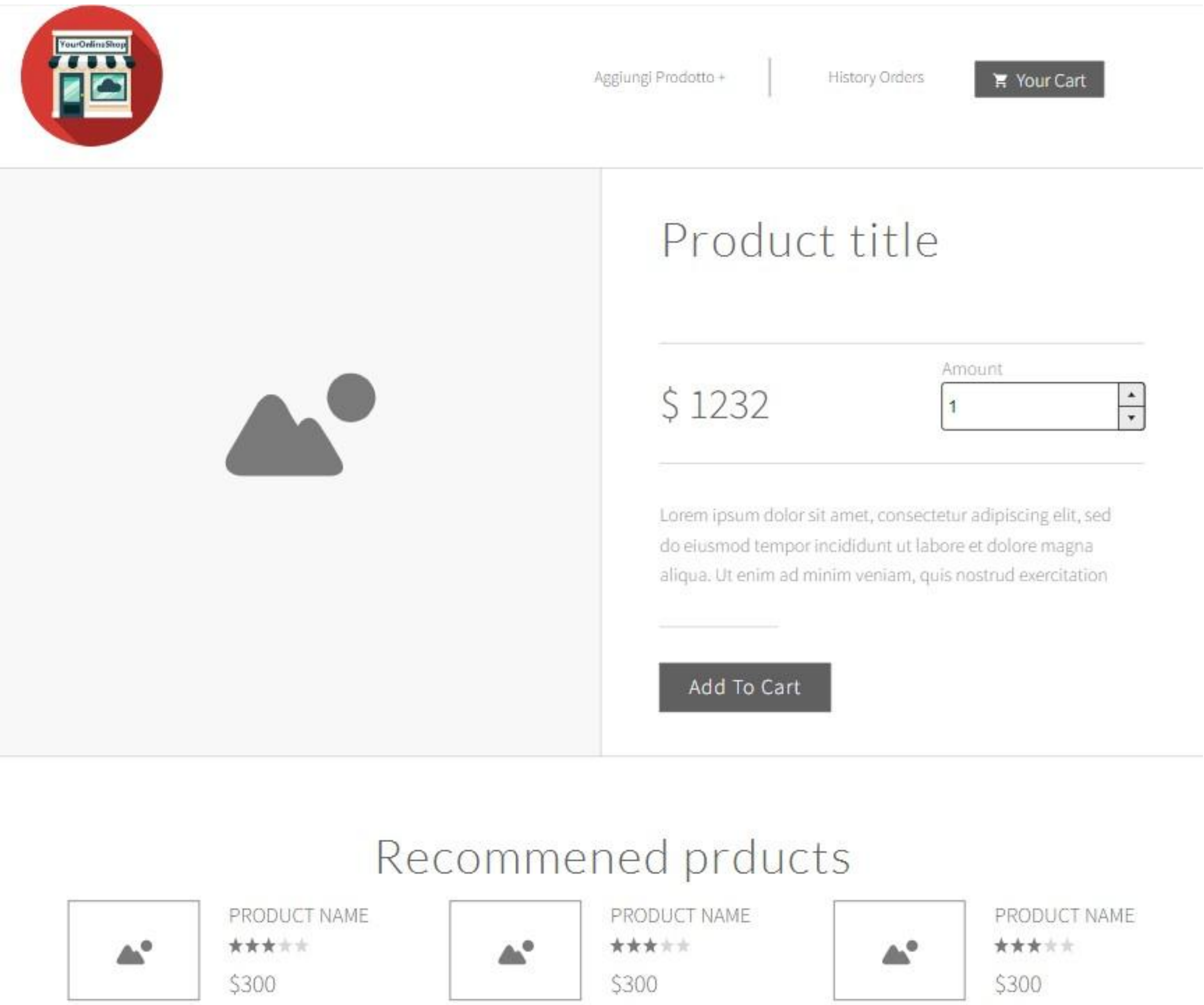
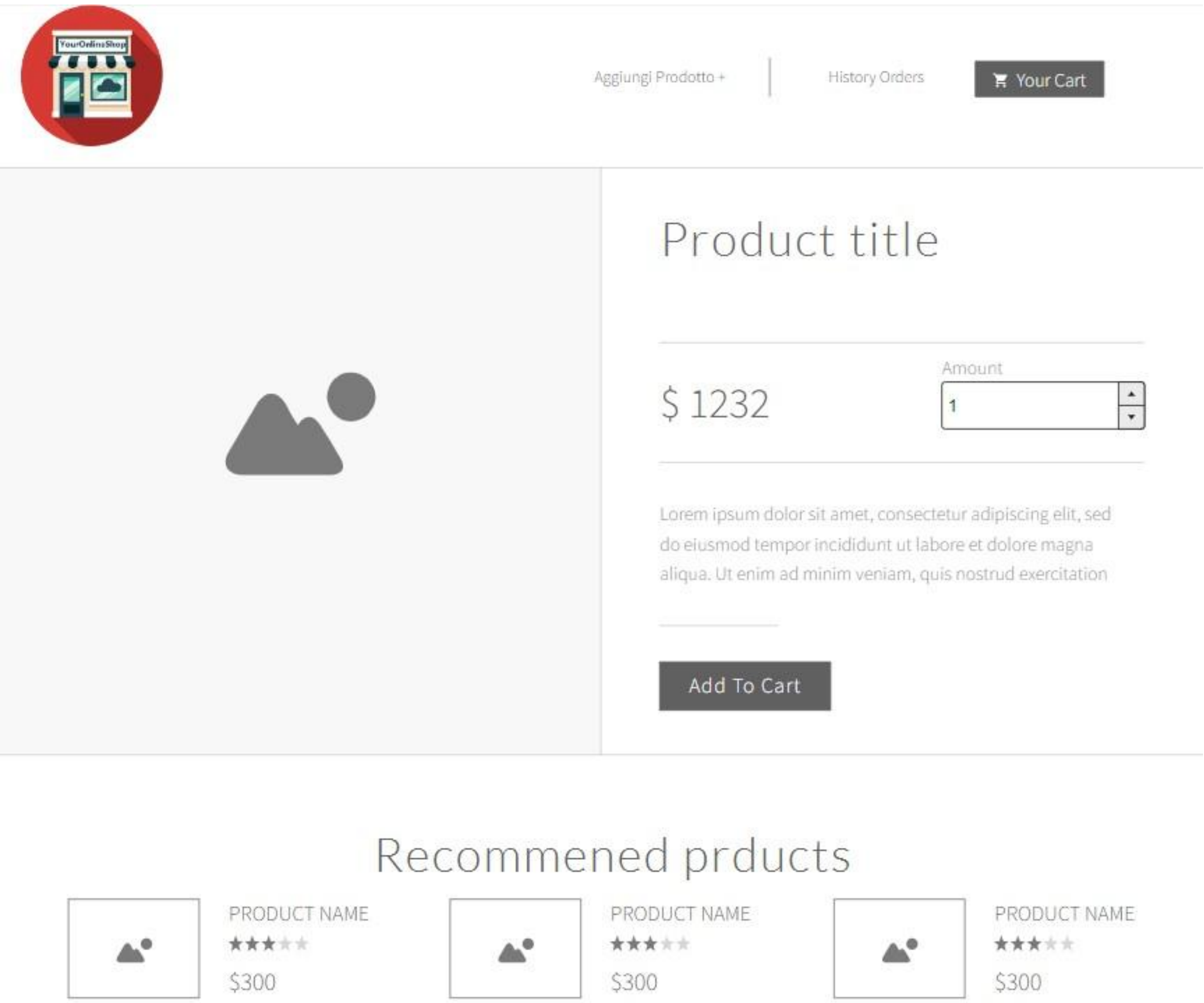

YourOnlineShop
Aggiungi Prodotto +
History Orders
Your Cart
Product title
$ 1232
Amount
1
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
Add To Cart
Recommened prducts
PRODUCT NAME
★★★★★
$300
PRODUCT NAME
★★★★★
$300
PRODUCT NAME
★★★★★
$300

Figure 7. User Story 6



Product title

$ 1232

Amount

1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation

Recommened prducts

PRODUCT NAME

★★★☆☆

$300

PRODUCT NAME

★★★☆☆

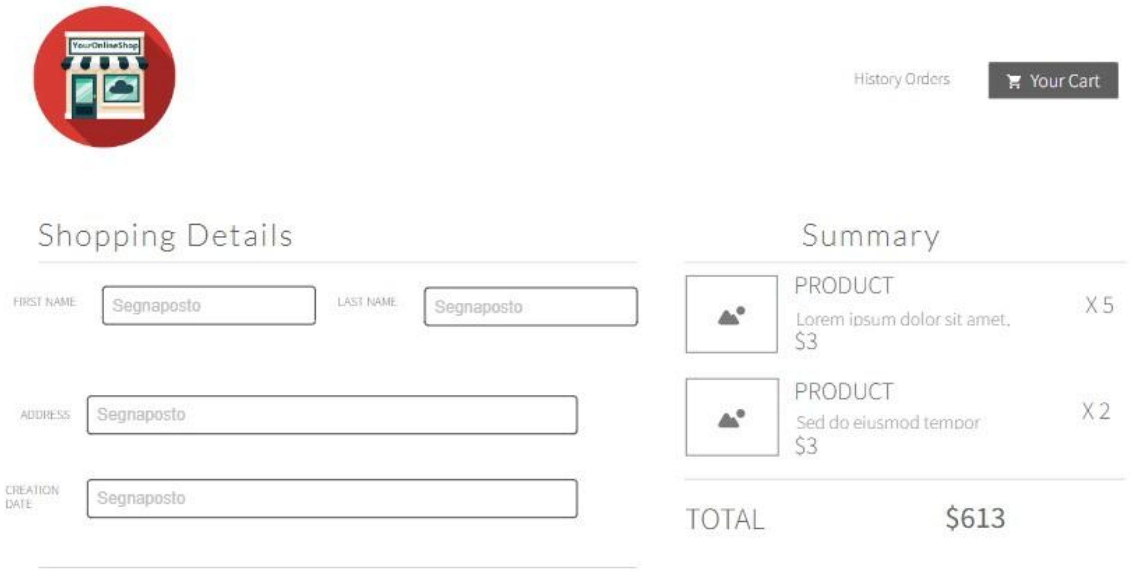
$300

PRODUCT NAME

★★★☆☆

$300

Figure 8. User Story 7



Shopping Details

FIRST NAME Segnaposto LAST NAME Segnaposto

ADDRESS Segnaposto

CREATION DATE Segnaposto

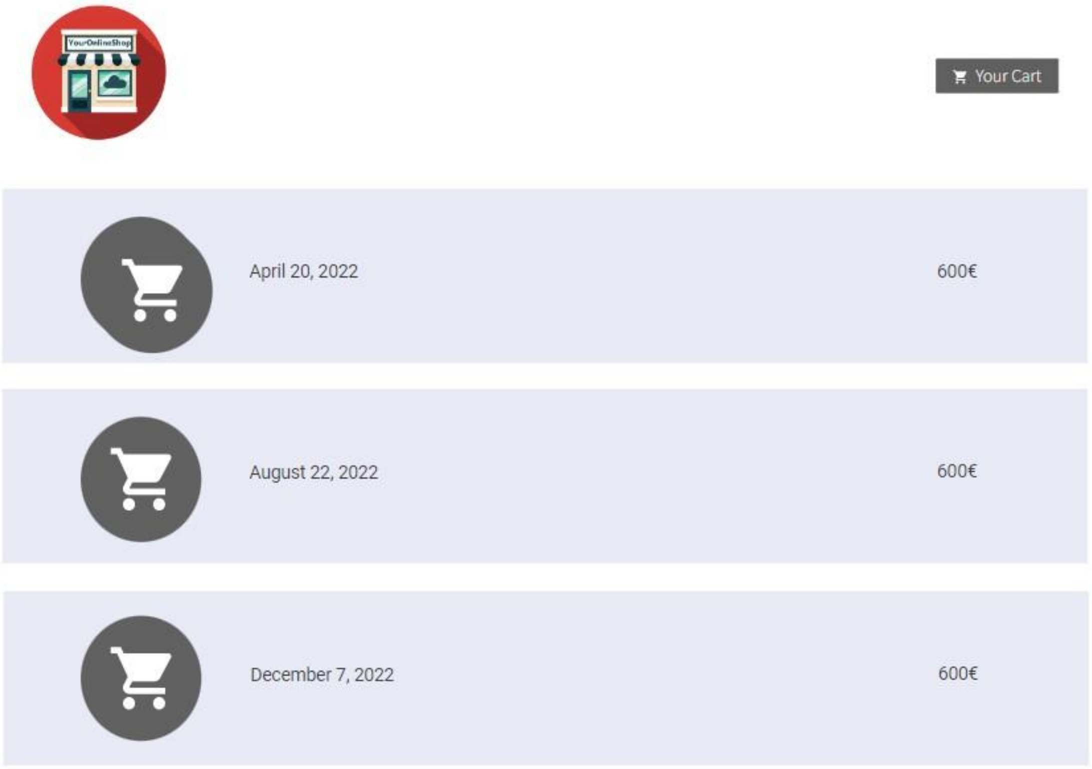
Summary

PRODUCT
Lorem ipsum dolor sit amet,
$3
X 5

PRODUCT
Sed do eiusmod tempor
$3
X 2

TOTAL $613

Figure 9. User Story 8

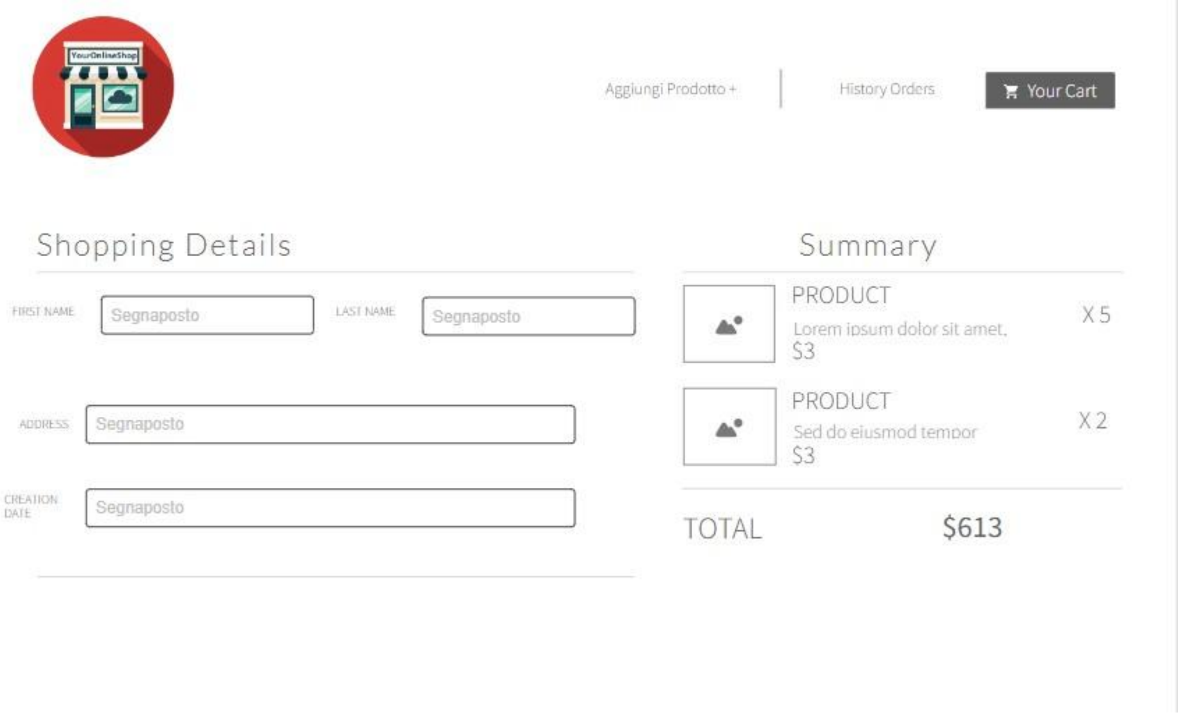


April 20, 2022 600€

August 22, 2022 600€

December 7, 2022 600€

Figure 10. User Story 9



Shopping Details

FIRST NAME Segnaposto LAST NAME Segnaposto

ADDRESS Segnaposto

CREATION DATE Segnaposto

Summary

PRODUCT
Lorem ipsum dolor sit amet,
$3
X 5

PRODUCT
Sed do eiusmod tempor
$3
X 2

TOTAL $613

Figure 11. User Story 10

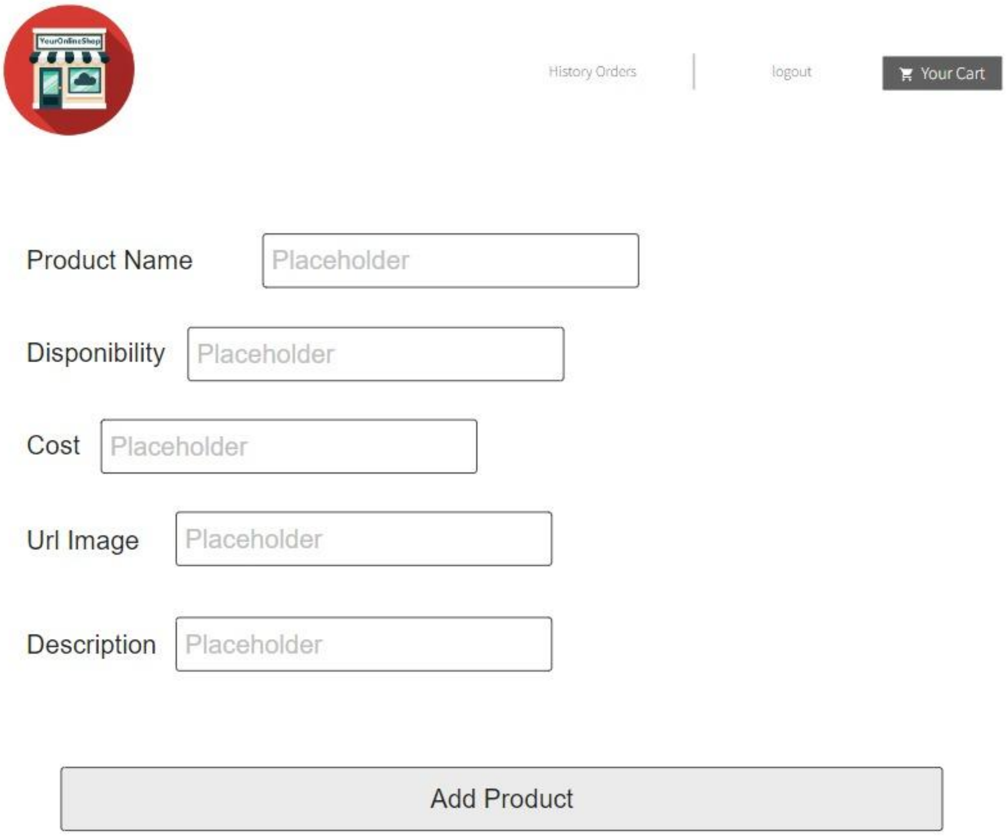


Figure 12. User Story 12-13

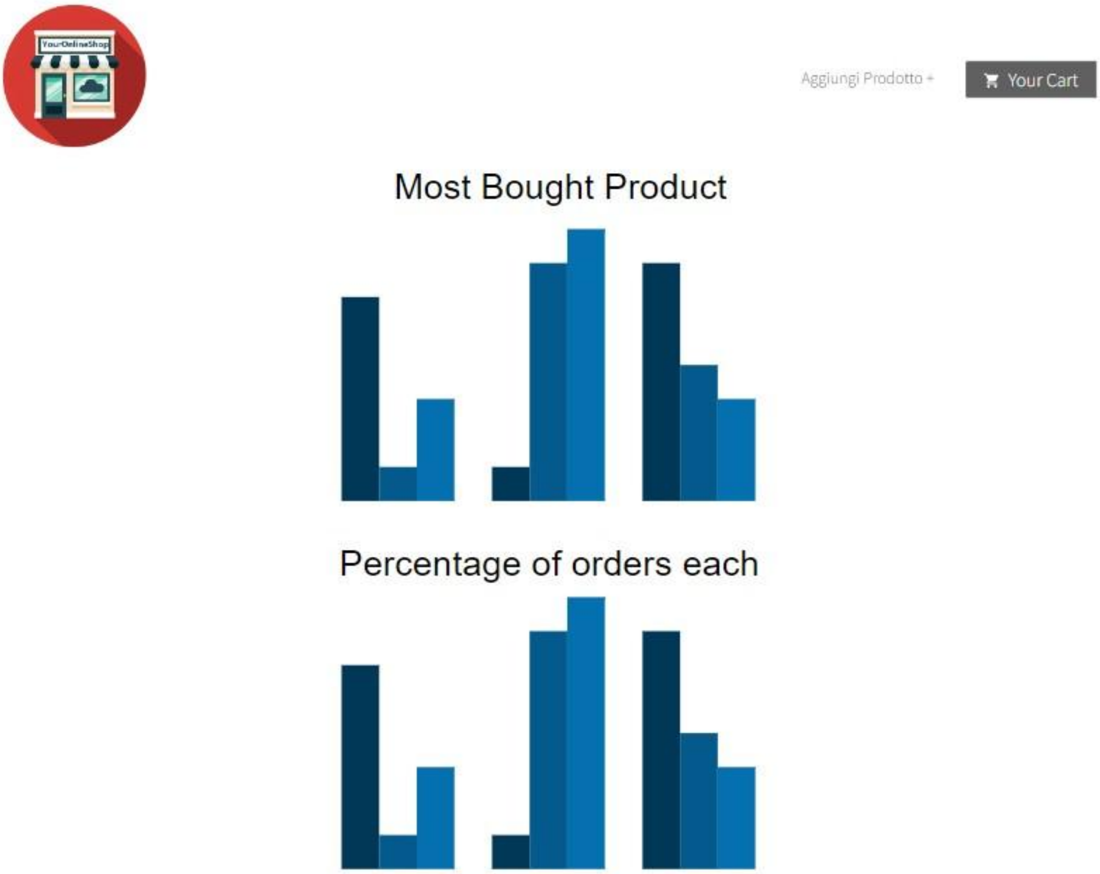
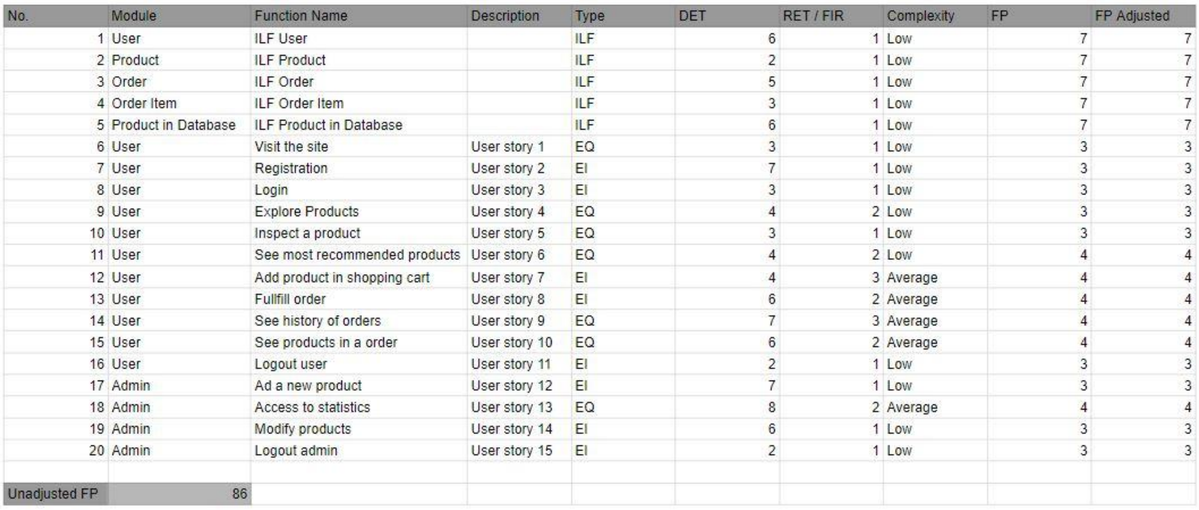


Figure 13. User Story 13

4. Complexity of Software Development

4.1. Function Points Calculation



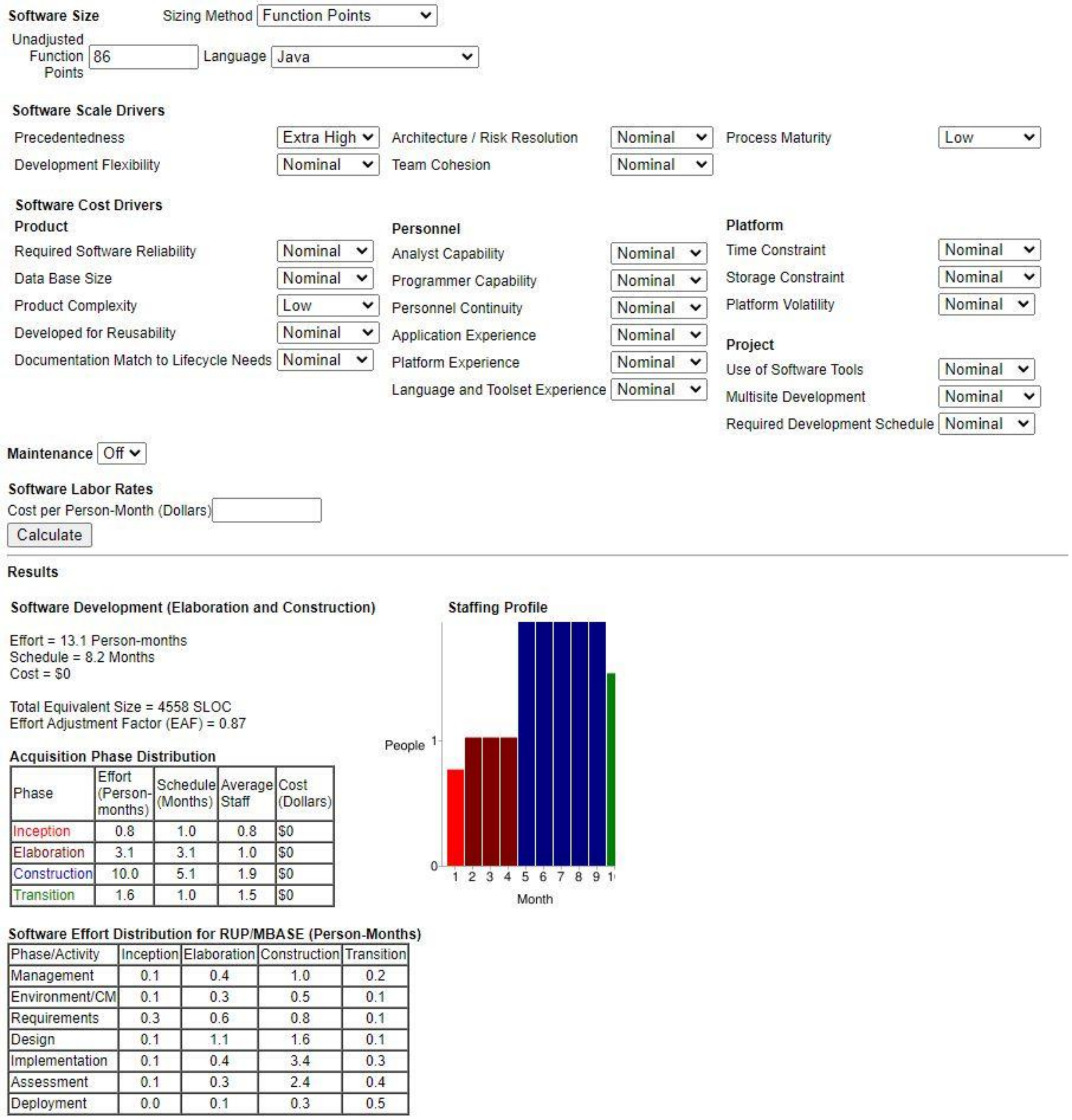
No.	Module	Function Name	Description	Type	DET	RET / FIR	Complexity	FP	FP Adjusted
1	User	ILF User		ILF	6	1	Low	7	7
2	Product	ILF Product		ILF	2	1	Low	7	7
3	Order	ILF Order		ILF	5	1	Low	7	7
4	Order Item	ILF Order Item		ILF	3	1	Low	7	7
5	Product in Database	ILF Product in Database		ILF	6	1	Low	7	7
6	User	Visit the site	User story 1	EQ	3	1	Low	3	3
7	User	Registration	User story 2	EI	7	1	Low	3	3
8	User	Login	User story 3	EI	3	1	Low	3	3
9	User	Explore Products	User story 4	EQ	4	2	Low	3	3
10	User	Inspect a product	User story 5	EQ	3	1	Low	3	3
11	User	See most recommended products	User story 6	EQ	4	2	Low	4	4
12	User	Add product in shopping cart	User story 7	EI	4	3	Average	4	4
13	User	Fullfill order	User story 8	EI	6	2	Average	4	4
14	User	See history of orders	User story 9	EQ	7	3	Average	4	4
15	User	See products in a order	User story 10	EQ	6	2	Average	4	4
16	User	Logout user	User story 11	EI	2	1	Low	3	3
17	Admin	Ad a new product	User story 12	EI	7	1	Low	3	3
18	Admin	Access to statistics	User story 13	EQ	8	2	Average	4	4
19	Admin	Modify products	User story 14	EI	6	1	Low	3	3
20	Admin	Logout admin	User story 15	EI	2	1	Low	3	3
Unadjusted FP	86								

Figure 14. Function Points calculation

The estimation in terms of FP and COCOMO II is presented in the spreadsheet above and in the one below.

It is possible to estimate the LOCs around 4,6KLOCs considering an average of 53 LOCs/FP using modern programming languages like Java. These lines are obviously including even the source code of the libraries that are going to be used in the project.

4.2. COCOMO II Calculation



Software Size Sizing Method: Function Points

Unadjusted Function Points: 86 Language: Java

Software Scale Drivers

Precedentedness	Extra High	Architecture / Risk Resolution	Nominal	Process Maturity	Low
Development Flexibility	Nominal	Team Cohesion	Nominal		

Software Cost Drivers

Product

Required Software Reliability	Nominal
Data Base Size	Nominal
Product Complexity	Low
Developed for Reusability	Nominal
Documentation Match to Lifecycle Needs	Nominal

Personnel

Analyst Capability	Nominal
Programmer Capability	Nominal
Personnel Continuity	Nominal
Application Experience	Nominal
Platform Experience	Nominal
Language and Toolset Experience	Nominal

Platform

Time Constraint	Nominal
Storage Constraint	Nominal
Platform Volatility	Nominal

Project

Use of Software Tools	Nominal
Multisite Development	Nominal
Required Development Schedule	Nominal

Maintenance Off

Software Labor Rates

Cost per Person-Month (Dollars)

Calculate

Results

Software Development (Elaboration and Construction)

Effort = 13.1 Person-months
Schedule = 8.2 Months
Cost = $0

Total Equivalent Size = 4558 SLOC
Effort Adjustment Factor (EAF) = 0.87

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	0.8	1.0	0.8	$0
Elaboration	3.1	3.1	1.0	$0
Construction	10.0	5.1	1.9	$0
Transition	1.6	1.0	1.5	$0

Staffing Profile

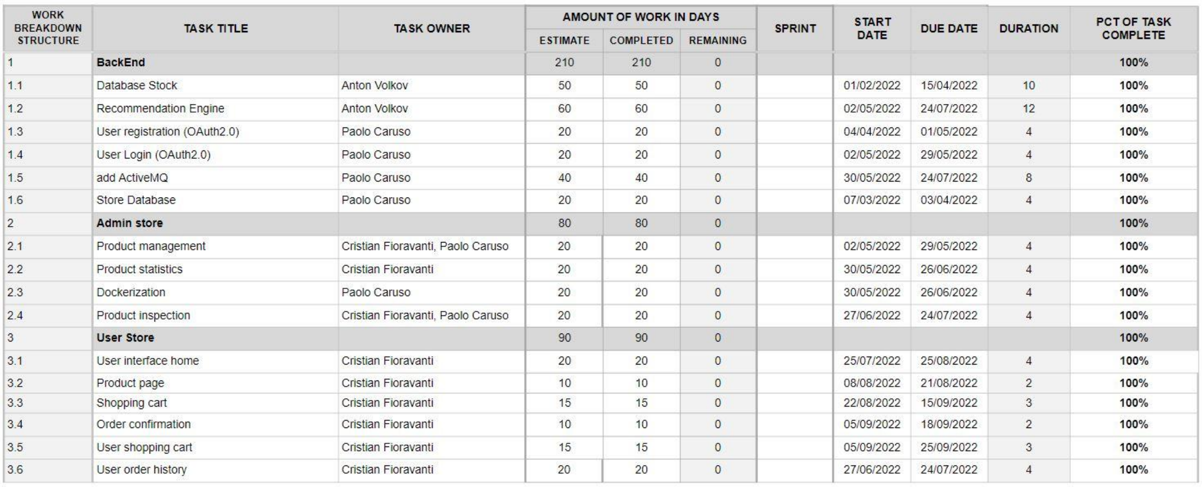
Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.1	0.4	1.0	0.2
Environment/CM	0.1	0.3	0.5	0.1
Requirements	0.3	0.6	0.8	0.1
Design	0.1	1.1	1.6	0.1
Implementation	0.1	0.4	3.4	0.3
Assessment	0.1	0.3	2.4	0.4
Deployment	0.0	0.1	0.3	0.5

Figure 15. COCOMO II Calculation

5. Development process based on SCRUM

These are the sprints and the product backlog of the project documented in a SCRUM spreadsheet:



WORK BREAKDOWN STRUCTURE	TASK TITLE	TASK OWNER	AMOUNT OF WORK IN DAYS			SPRINT	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
			ESTIMATE	COMPLETED	REMAINING					
1	**BackEnd**		210	210	0					100%
1.1	Database Stock	Anton Volkov	50	50	0		01/02/2022	15/04/2022	10	100%
1.2	Recommendation Engine	Anton Volkov	60	60	0		02/05/2022	24/07/2022	12	100%
1.3	User registration (OAuth2.0)	Paolo Caruso	20	20	0		04/04/2022	01/05/2022	4	100%
1.4	User Login (OAuth2.0)	Paolo Caruso	20	20	0		02/05/2022	29/05/2022	4	100%
1.5	add ActiveMQ	Paolo Caruso	40	40	0		30/05/2022	24/07/2022	8	100%
1.6	Store Database	Paolo Caruso	20	20	0		07/03/2022	03/04/2022	4	100%
2	**Admin store**		80	80	0					100%
2.1	Product management	Cristian Fioravanti, Paolo Caruso	20	20	0		02/05/2022	29/05/2022	4	100%
2.2	Product statistics	Cristian Fioravanti	20	20	0		30/05/2022	26/06/2022	4	100%
2.3	Dockerization	Paolo Caruso	20	20	0		30/05/2022	26/06/2022	4	100%
2.4	Product inspection	Cristian Fioravanti, Paolo Caruso	20	20	0		27/06/2022	24/07/2022	4	100%
3	**User Store**		90	90	0					100%
3.1	User interface home	Cristian Fioravanti	20	20	0		25/07/2022	25/08/2022	4	100%
3.2	Product page	Cristian Fioravanti	10	10	0		08/08/2022	21/08/2022	2	100%
3.3	Shopping cart	Cristian Fioravanti	15	15	0		22/08/2022	15/09/2022	3	100%
3.4	Order confirmation	Cristian Fioravanti	10	10	0		05/09/2022	18/09/2022	2	100%
3.5	User shopping cart	Cristian Fioravanti	15	15	0		05/09/2022	25/09/2022	3	100%
3.6	User order history	Cristian Fioravanti	20	20	0		27/06/2022	24/07/2022	4	100%

Figure 16. Product backlog and Sprints in SCRUM

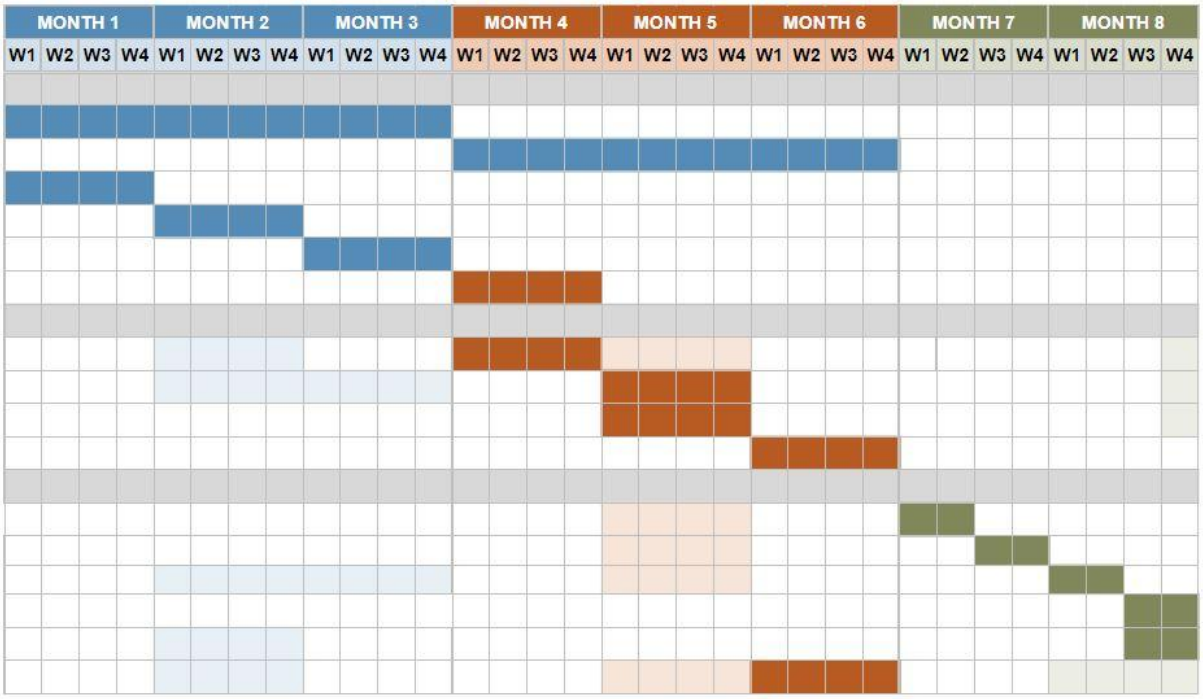


Figure 17. Task Gantt chart, each row corresponds to the same row in Figure 18.

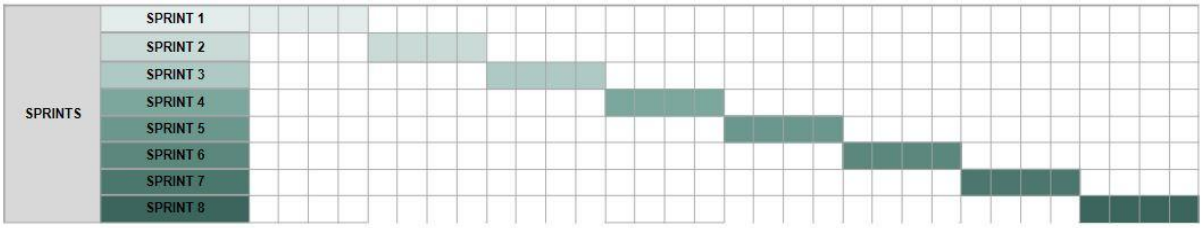


Figure 18. Sprint Gantt chart

6. Technologies

6.1 MySQL: Product database & Store database

The product and store databases are developed in Java using Spring Boot, in particular with the use of the framework Hibernate, used to simplify database access by mapping Java objects to relational database tables. The code was then built and dockerized, with the resulting docker used in a Docker Compose which connects it to a mySQL docker container where the data is stored.

6.2 Recommendation Engine

The Recommendation Engine is built with Python starting from the recommendation algorithm library LightFM, and provides communication for other services with a Flask-RESTful API. It is dockerized so it can be used easily with minimum setup.

6.3 ActiveMQ

ActiveMQ is an open-source, JMS (Java Message Service) compliant messaging system that provides reliable, asynchronous messaging for Java-based applications. It supports a variety of messaging styles including point-to-point, publish-subscribe, and message groups. By using ActiveMQ, the project is leveraging a Java-based messaging system for asynchronous communication.

6.4 OAuth 2.0

It is a framework for open-standard authorisation that enables secure access to resources like APIs and Web Services. By getting an access token that may be used to access the resource, it lets clients to access resources owned by resource owners (such as users). The project is utilizing a secure authorization framework by utilizing OAuth 2.0 to control user authentication and authorization and access resources.

7. Distribution

The steps to follow to build and deploy the system on whatever platform (either on-premise or on cloud) using a IAC approach are the following ones:

- Install Docker Desktop on your device.
- Download the zip of the project from GitHub at .
- Unzip the project and open a terminal inside it.
- Execute maven clean install
- Execute docker-compose up –build -d.
- From a Browser digit localhost:8080/home to access the website.

The GitHub link contains all the source code, configuration files, docker compose files and dockerfiles.