STANFORD UNIVERSITY

ADMINISTRATIVE SYSTEMS

FUNCTIONAL SPECIFICATION DOCUMENT

MEGAPOP

DOCUMENT VERSION <1.0>

21/11/2019

AUTHORS

Name	Role	Department
José Luis Magdaleno	Front-end	Java Application Developer
Vieux Djigo	Back-end	
Marc Vilaró	Database	
Antoni Espadas	Manager	

Table of Contents

1. Introduction

Recently, the idea MegaPop is born in the need to investigate an alternative of e-commerce website for the sale of C2C products. The reason were motivated by the irruption ideas to influence in market electronic share in first and second hand sales. The main idea to build up this page turns around on surface user motivated to search original products unknown and feel secure to purchase without contact with the vendor. MegaPop is the intermediate to make the sales process. Java Spring is the main tool implemented in this system program.

1.1 Purpose of the document

The Functional Specification Document is a document that provides detailed information on *how* the system solution will function and the requested behavior. This document is created based on the high-level requirements identified in the Business Requirements Document and provides traceability on the functional specifications back to the business requirements. Included in this document will be the detailed functional requirements including use cases, system inputs and outputs, process flows, diagrams, and mock ups.

1.2 Project Scope

This project will consist of creating a web site based upon the first and second hand sales. The project will be completed by 21 November, 2019 in LA CIFO Violeta (Barcelona). Modules of the page will include a free sign up, a way for MegaPop to grow surface users, and a way to motivate sells products.

1.3 Related documents

Component	Name (with link to the document)	Description
Balsamiq	https://balsamiq.com/wireframes/	Wireframing
Lucidchart	https://www.lucidchart.com/pages/es	Diagram UML
Trello	https://trello.com/	A list of lists by your team
GitHub	https://github.com/	Development platform to code
H2	https://www.h2database.com/html/main.html	Java SQL database
YouTube	https://www.youtube.com/	Videos platform

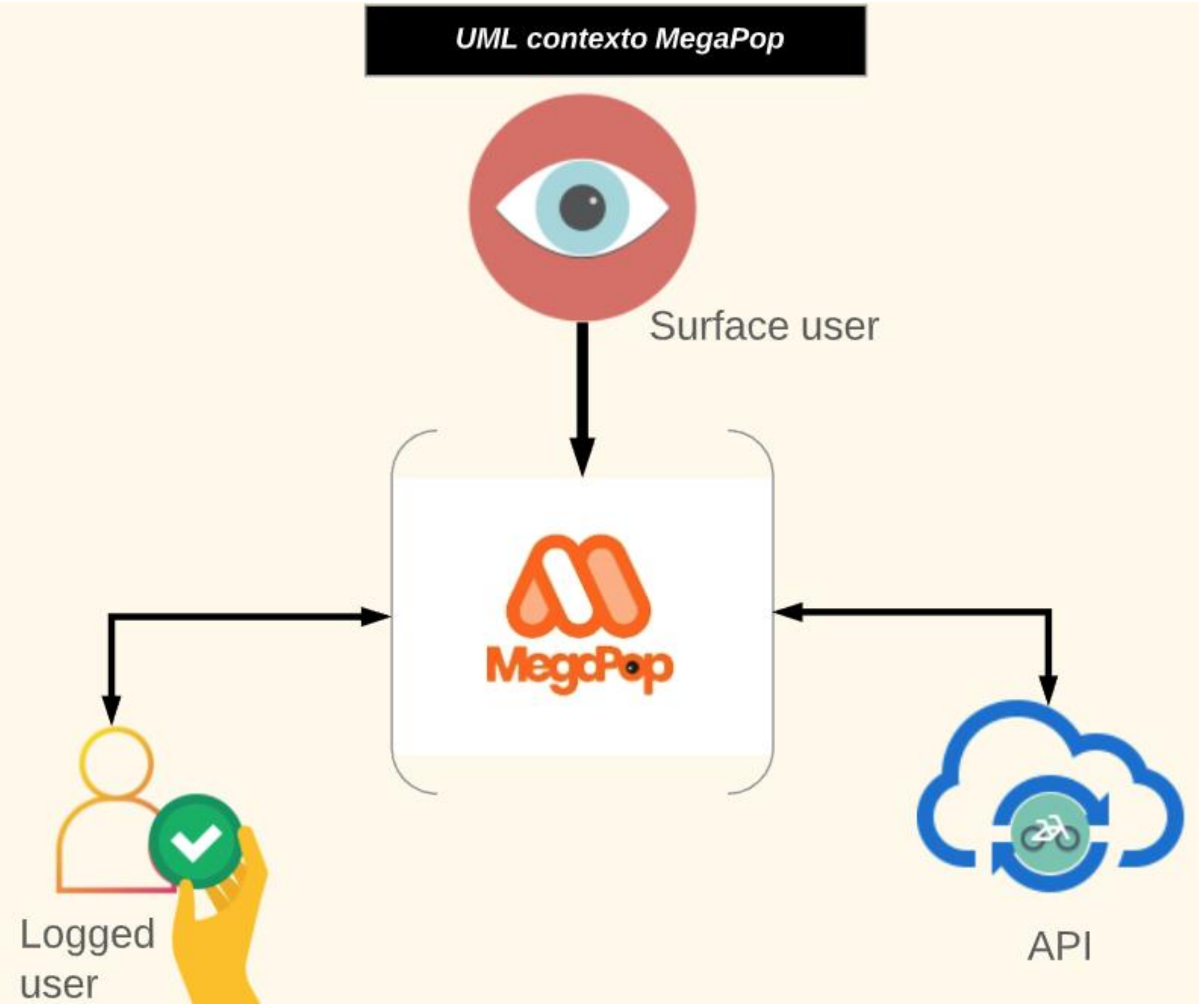
1.4 Terms/Acronyms and Definitions

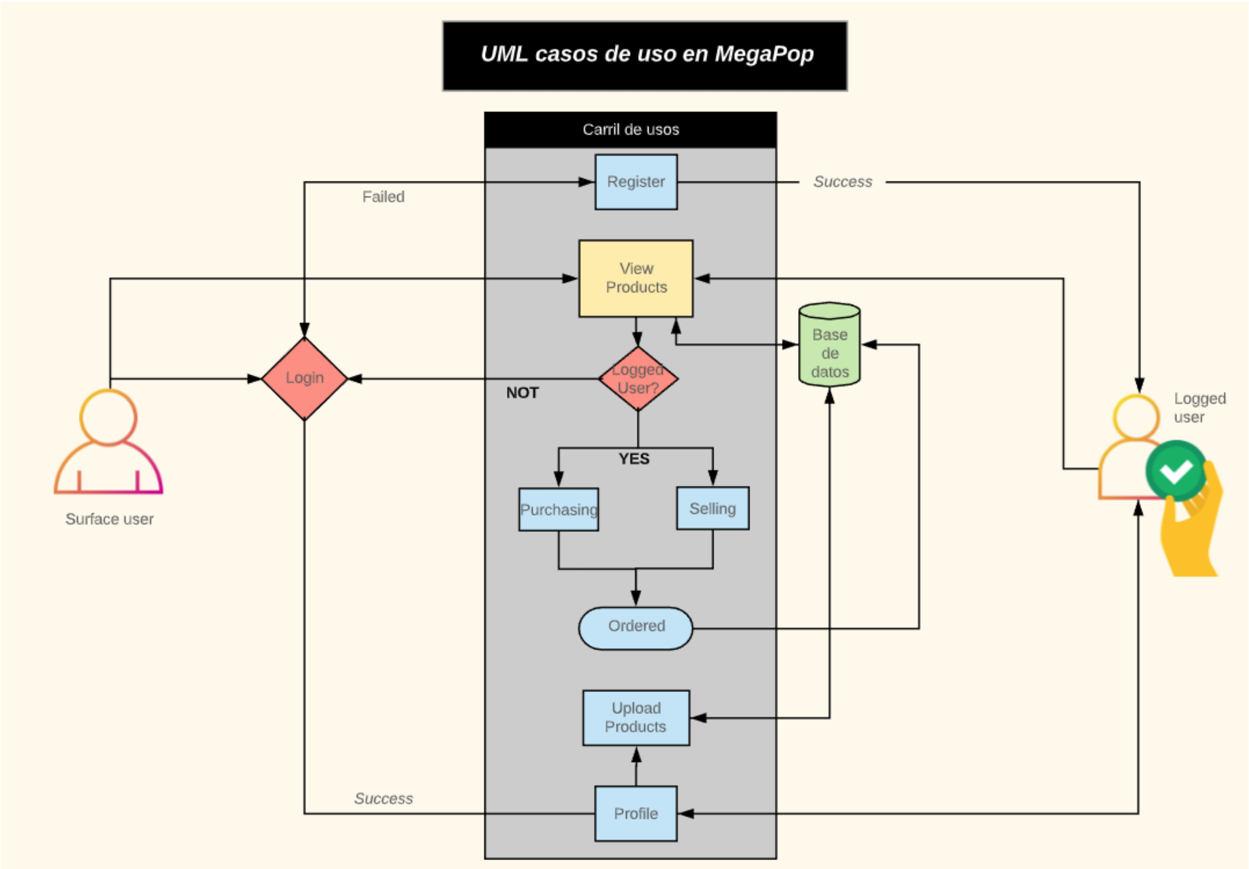
Term/Acronym	Definition	Description
JPA	Java Persistence API	Java application programming
MVC	Model–View–Controller	Software design pattern
C2C	consumer-to-consumer	Relating to buying and selling
DB	database	Dollection of data software
H2 DB	database's name	Java SQL database used

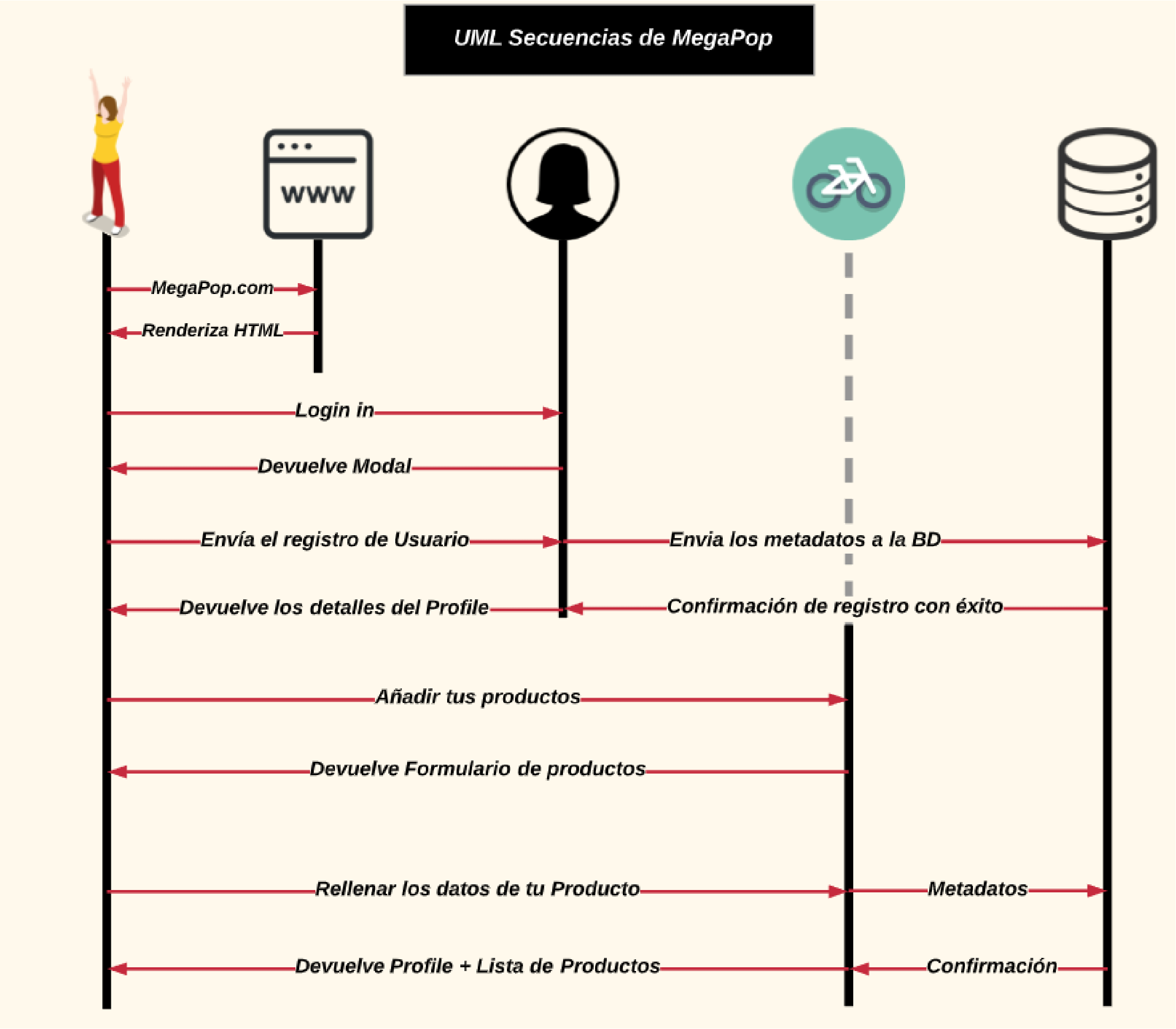
2. System/ Solution Overview

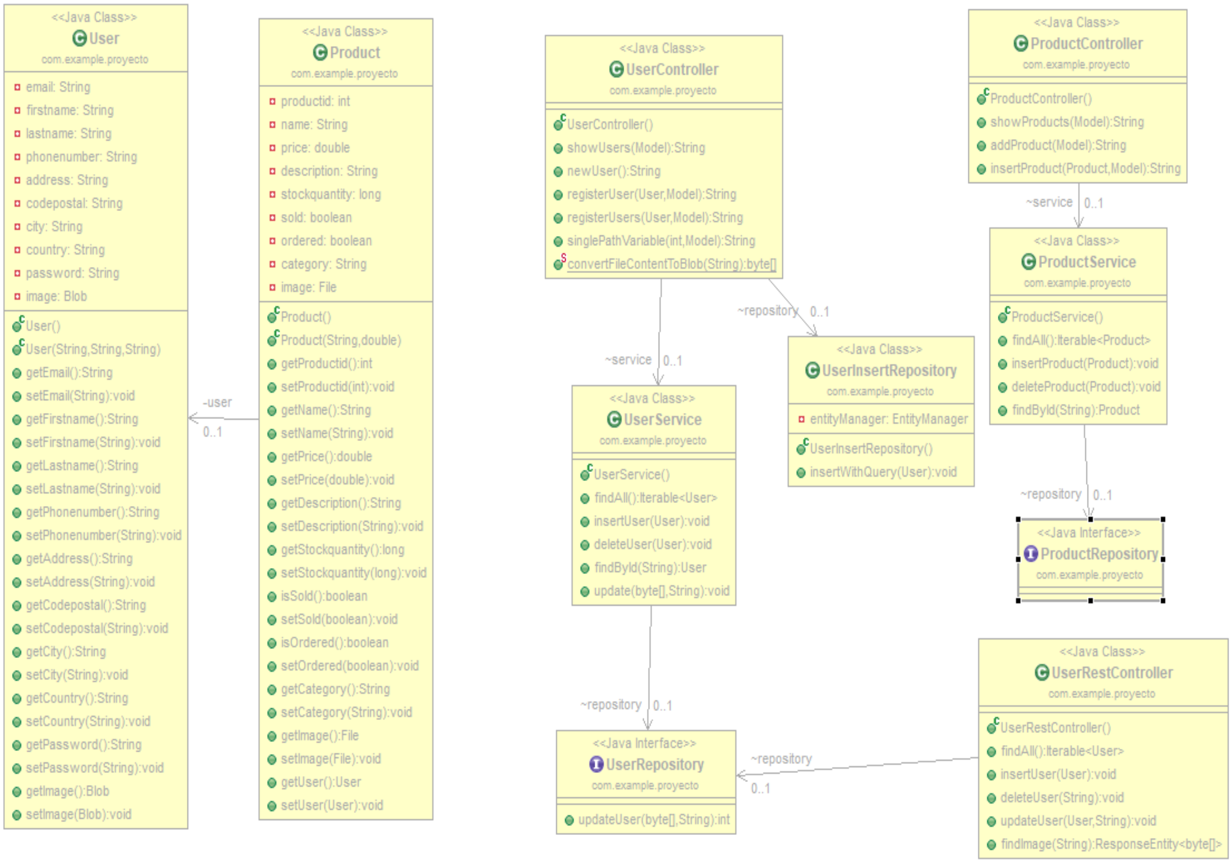
MegaPop works with Java Spring Boot application for programming the whole system. We combine different tools (look up chapter 1.3) to make the decision made by the team. We put focus on a simple website with a estable database what allows to applied knowledge and learning curve short term. Then, we identify two main role: surface user, who can enter the website only for view products, and logged user, who is allowed to do different tasks like purchase or sell. This make a interchange flow information and different condition to code in Java Spring. JPA, H2 database, Thymleaf are needed to keep this mess up on.

2.1 Context Diagram/ Interface Diagram/ Data Flow Diagram, Application Screen Flow, Sitemap, Process Flow




UML casos de uso en MegaPop
Carril de usos
Register
Failed
Success
View Products
Base de datos
Login
Logged User?
NOT
Logged user
YES
Purchasing
Selling
Surface user
Ordered
Upload Products
Success
Profile


UML Secuencias de MegaPop
www
MegaPop.com
Renderiza HTML
Login in
Devuelve Modal
Envía el registro de Usuario
Envia los metadatos a la BD
Devuelve los detalles del Profile
Confirmación de registro con éxito
Añadir tus productos
Devuelve Formulario de productos
Rellenar los datos de tu Producto
Metadatos
Devuelve Profile + Lista de Productos
Confirmación



2.2 System Actors Java

2.2.1 Java Classes

User/Role	Relationship	Interface/ Public Class
Product	*User*	Public
User	*User*	Public
UserController	*Controller*	Public
UserService	*Repository*	Public
UserRepository	*Repository*	Interface
UserInsertRepo.	*Repository*	Public

ProductController	*Service*	Public
ProductService	*Repostery*	Interface
ProductRep	*Repostery*	Interface
UserRestController	*Repostery*	Public

3. Functional Specifications

There're a public actors that we have to remain and respect each other. It's the most human tool. We divide the whole work in different areas, but running in the same direction

Class	Frequency of Use	Security/Access, Features Used	Additional Notes
Back-End	*Frequent*	responsive Spring Controller	
Data Base	*Frequent*	responsive Spring Model and Database	
Manager	*Frequent*	responsive to collaborate in the three areas	
Front-End	*Frequent*	responsive web design Spring vista	

The main function works around the project is the action for purchase and sell in order C2C.

3.1 <Purchasing and Selling>

3.1.1 Purpose/ Description

Next information explain about a user logged allows to order/purchase or selling a desirable product applied Spring Boot.

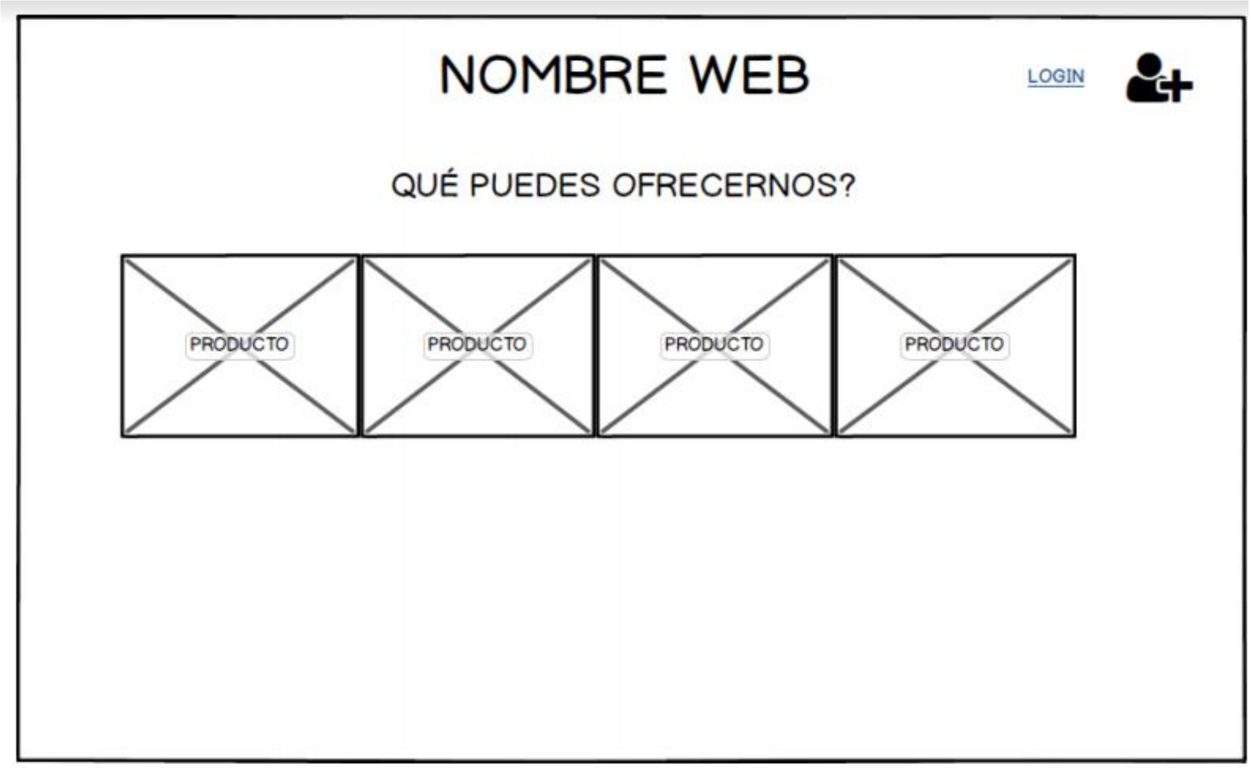
3.1.2 Use case

UC-1	Purchasing
Primary Actor(s)	*Front-end - Controller - DB*
Stakeholders and Interest	*Surface User*
Trigger	*need to be registered to purchase*
Pre-conditions	*Not need*
Post-conditions	*be registered with success*
Main Success Scenario	*1. At Home Page select one product desirable* *2. Need to be logged* *3. In case not, need to fill the register's formulary and success* 4. Click button Ordered
Extensions	In case not be Loggin, need to be registered and continue with the purchasing process
Priority	*High*
Special Requirements	*Be logged*

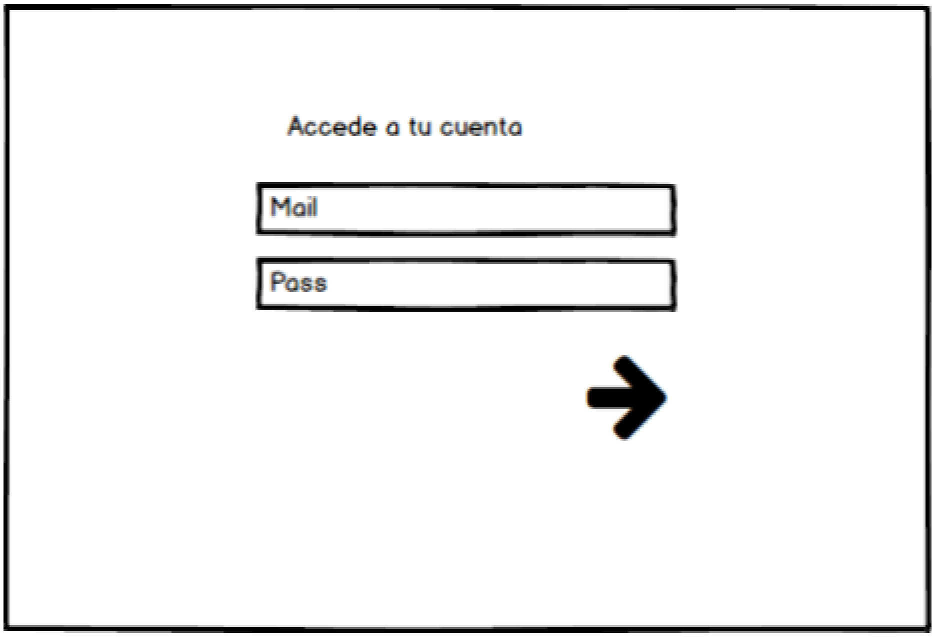
UC-1	Selling
Primary Actor(s)	*Front-end - Controller - DB*
Stakeholders and Interest	*Logged user*
Trigger	*need to be registered to sell*
Pre-conditions	*Be logged and be purchased by some logged user*
Post-conditions	*Upload a product with its full information filled*
Main Success Scenario	*1. Stay in profile* *2. Add new product to sell* *3. In case sold, the information will appear on profile home*
Extensions	In case not be Loggin, need to be registered and continue with the selling process
Priority	*High*
Special Requirements	*Be logged*

3.1.3 Mock-up

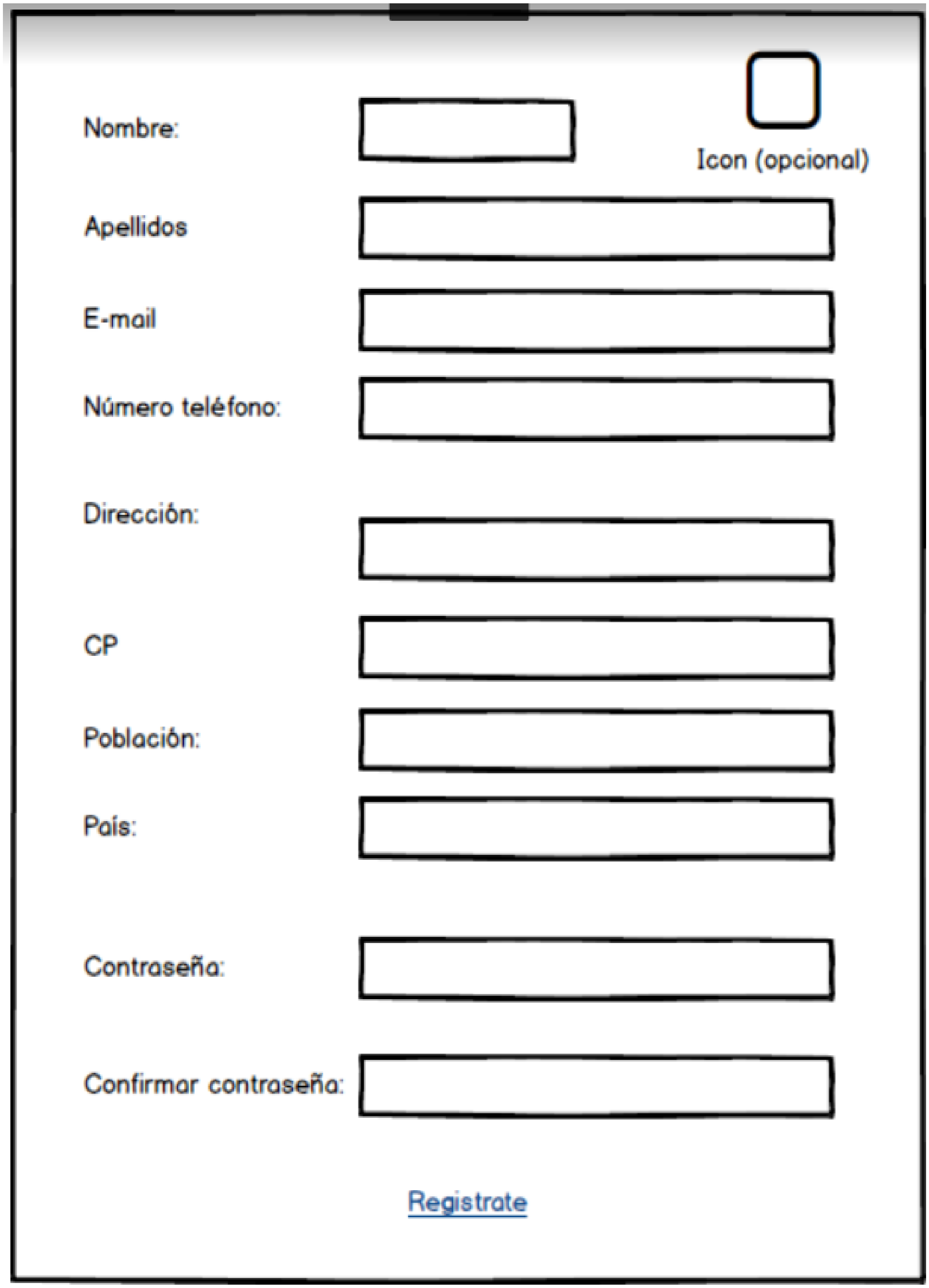
Mock UP - Surface User



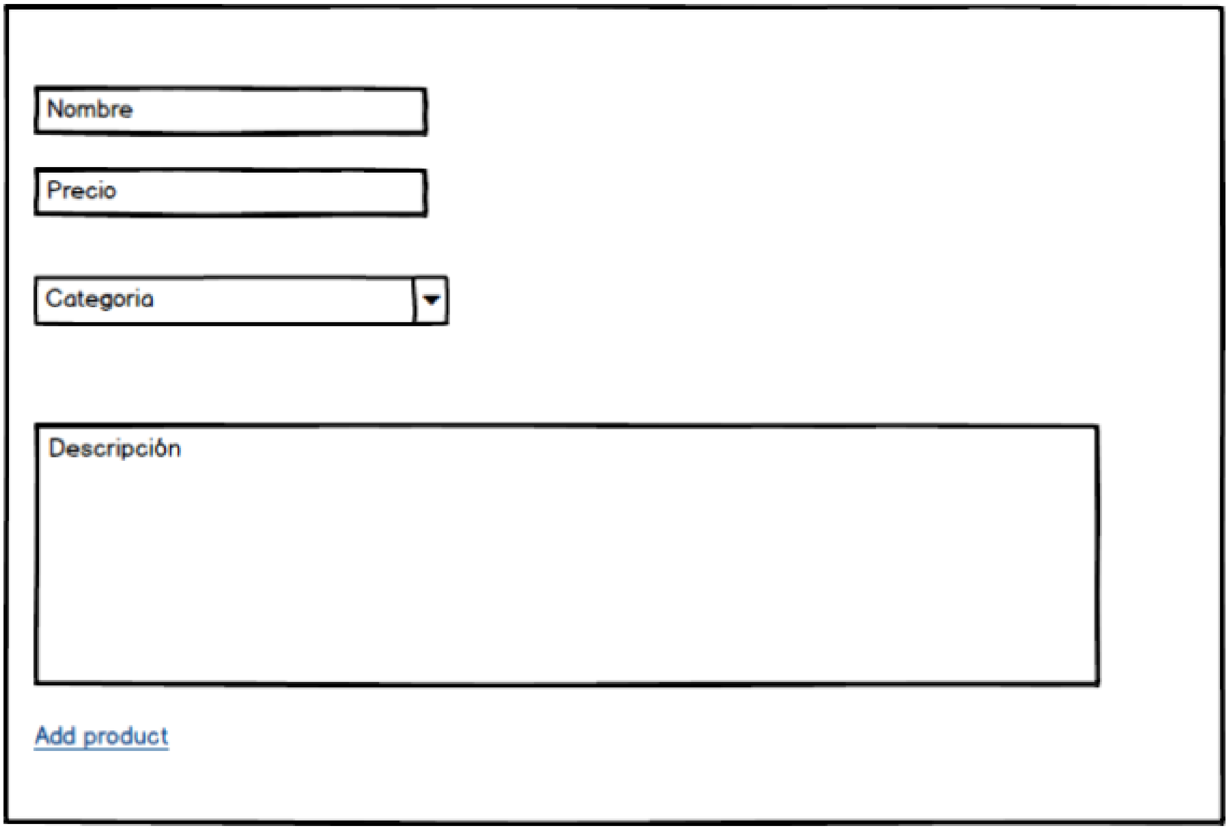
Mock-Up Loggin



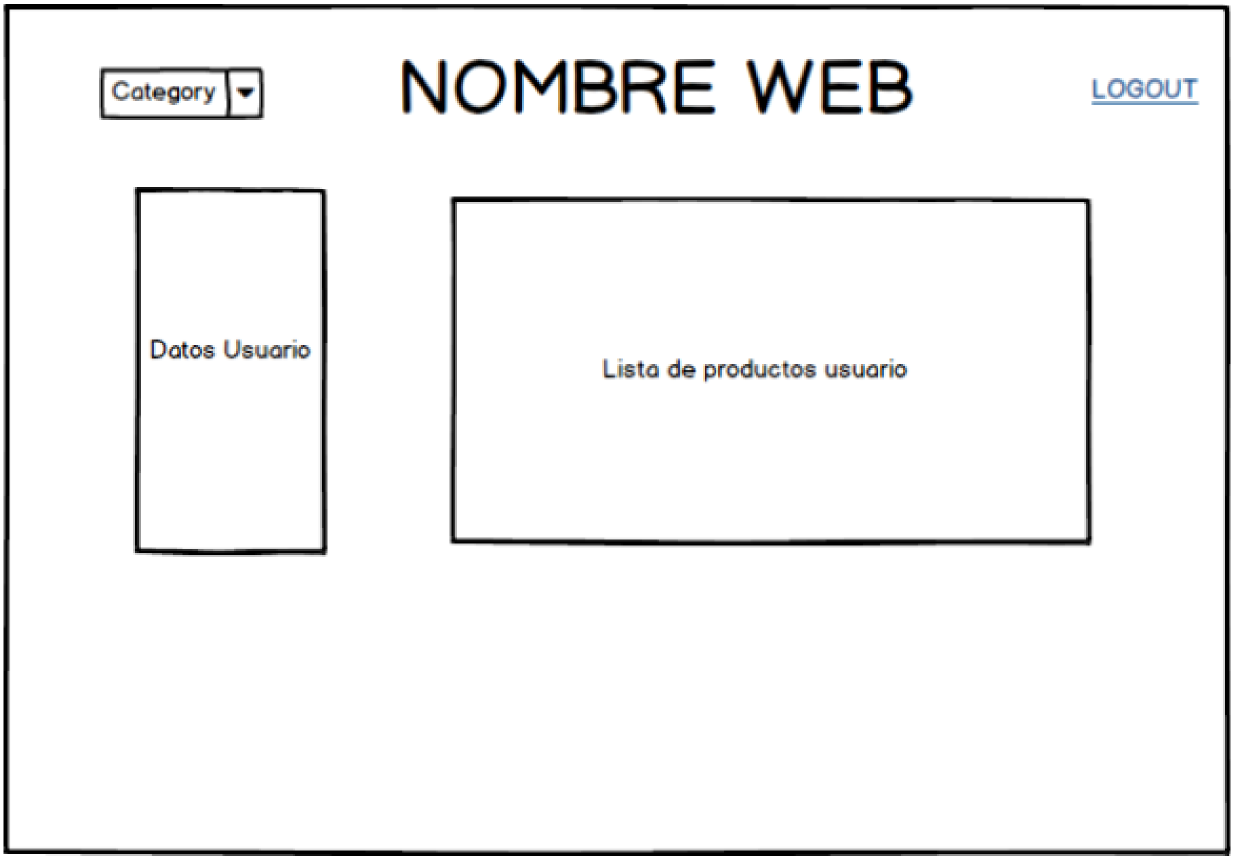
Mock-Up Formulary



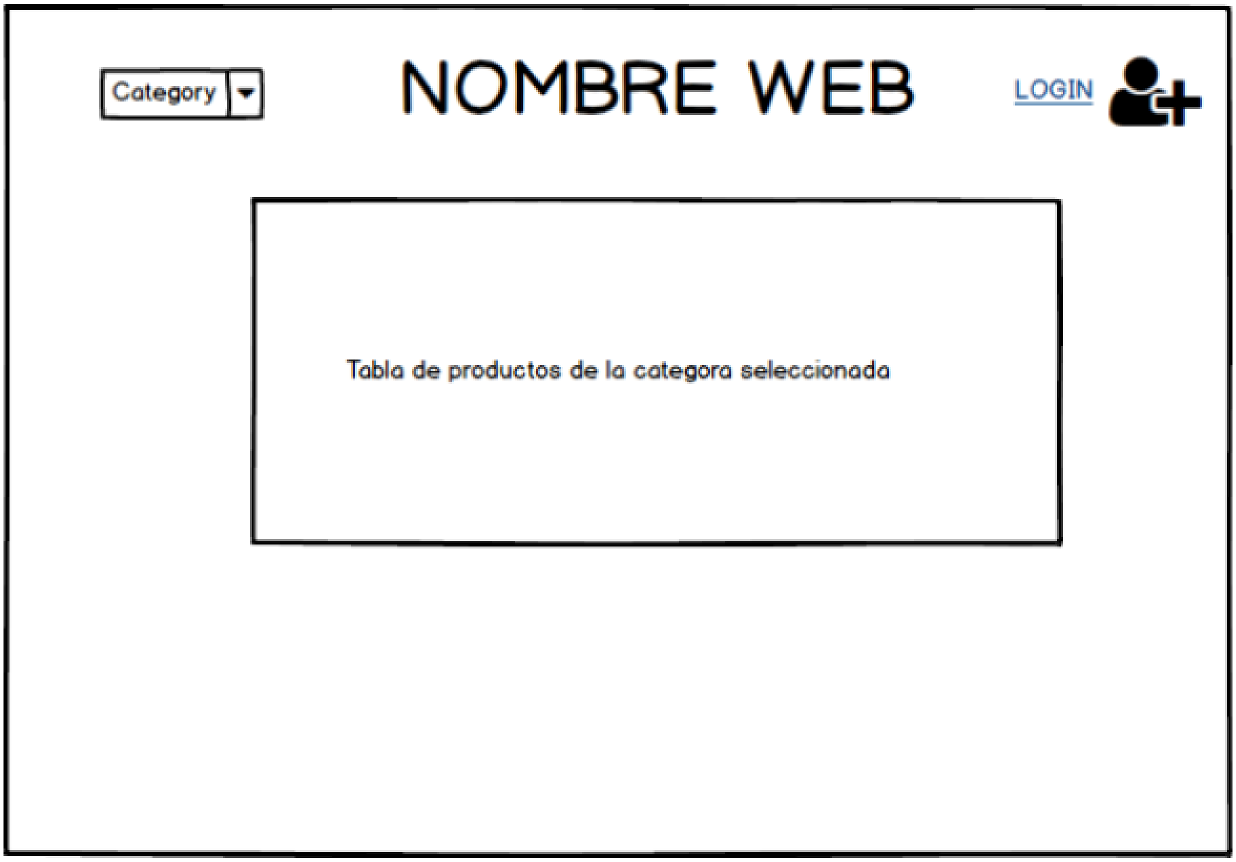
Mock Up Product Register



Mock-Up Profile



Mock -up Category



3.1.4 Functional Requirements

Spec ID	Specification Description
Mock UP - Surface User	*Show the main home page for surface user. You could view all the products and loggin if you desire.*
Mock-Up Formulary	*In case not be registered, you receive a formulary that you must complete every field*
Mock -up Category	*Show all the products registered by differents users logged*
Mock-Up Profile	*Show the whole information about user logged*
Mock-up	*<Short explanation of the specification>*

3.1.5 Field level specifications

Form Elements:

Call-out	Field Label	UI Control	Mand?	Editable	Data Type	Value Set	Data Example
Mock-Up Loggin	Login	textbox	Yes	Yes	String	Alphanumeric	Hello123
Mock-up Category	Search	textbox	Not	Yes	String	Alphanumeric	Samsung
Mock-Up Formulary	Register	Button	Yes	Yes	String	NA	Formulary
Mock-Up Profile	Add new product	Button	Yes	Yes	String	NA	Formulary

Form Business Rules and Dependencies:

Field Label	Validation / Business Rules	Error Messages
Example: User name	*User name shall be a valid Stanford Sunet ID*	*For incorrect user name display on setFocusOff: “Please provide a valid user name”*
User Logged	*User logged should be keep up himself information*	*In case not be filled personal information a screen show: “There’s field on blanks to be filled.”*
List of Products	*indicates if it’s sold: yes or not*	*If you click a product but it’s already purchased, the product list should remove the product*
Product and user images	*User and products could upload images*	*Indicate there’s upload images options when you fill your personal detailes*

Register user	*Surface user need to fill all the fields to complete the register*	*In case not be filled personal information a screen show: “There’s field on blanks to be filled.”*

Buttons, Links and Icons:

Button, Link, Icon Label	OnClick Event	Other Event	Visible	Enabled Vs Disabled	Navigate To	Validation	Dependencies
Register	*Send to Controller the response and return the Formulary to be filled as new user*	*Display a Formulary page to fill every field*	*Yes, as long as the user is not logged in.*	*Enable, by default. Disable, after first logged in*	*User Dashboard page*	*Verify if Username is a valid sunset ID and user name and password collects correctly in Data Store.*	*Disable the New user functionality on subsequent pages if user logs in through this button.*
Login	*Verify if user name and password are correct. If yes, log the user into the system.*	*OnMouseHover display the following message: “User or Password incorrect.”*	*Yes, always*	*Enable, by default. Disable, after first key entered in either username or password field.*	*User Dashboard page*	*Verify if Username is a valid sunet ID and user name and password match with registry data.*	*Disable the New user functionality on subsequent pages if user logs in through this button.*
Filter	*Display a list of products specifically a category selected*	*Display a blank list without products to search*	*Yes, always*	*Enable, by default.*	*User Dashboard page at the top*	*Show the whole list from de Database about the category selected*	*Not click that button the function search not run.*
Add new Product	*Send to Controller the response and return the*	*Display a Formulary page to fill every field*	*Yes, when you’re Profile site.*	*Disable, by default. Enable when you’re Profile site.*	*User logged Dashboard page*	*Add Product User inside Database Store to*	*Need a Profile User logged and active profile.*

	Formulary to be filled as new Product user					*sell your products in the specify category*	
Delete	*Delete a product profile*	*after clicking, immediately disappear*	*Yes, when you're Profile site.*	*Disable, by default. Enable when you're Profile site*	*User logged Dashboard page*	*Delete Product User inside Database Store*	*Need a Profile User logged and active profile.*
Sold?	*Make the condition Sold of your Product in TRUE or FALSE*	*FALSE or TRUE, after clicking. If it's True, the product can't be Ordered*	*Yes, when you're Profile site.*	*Disable, by default. Enable when you're Profile site*	*User logged Dashboard page*	*Indicate to us if the product is available or not.*	*Need a Profile User logged and active profile.*
Ordered	*Action to purchase a product selected*	*OnMouseHover display the following message: "This product is not available."*	*Yes, when you're Category List filtered*	*Disable, by default. Enable when you're Category filtered*	*User logged Category LIst page*	*Action to purchase*	*Need to be logged in and product NOT sold.*

4. Exception Handling/ Error Reporting

Exception/ Error ID	Error	Cause	Solution Strategy
Upload Images	Data Base not return images to Vista	H2 not store images to Spring	Implement new DB

6. References

Please, for further information consult the code fluent about the entire project on Git-HUB website:

https://github.com/MegaPopPQTM-2019/Proyecto_MegaPop

Otherwise, if you need to look up the website, we attached you a MegaPop demo video on YouTube:

https://www.youtube.com/watch?v=fVmBoiFoi_c