**Nombre: Joshua Lattke, Kevin Andrés Banguero, Samuel Barona**

**Code: A00372620, A00369589, A00394081**

**SOFTWARE ENGINEERING PROBLEM SPECIFICATION TABLE, IDENTIFYING THE FOLLOWING ELEMENTS**

|  |  |
| --- | --- |
| CUSTOMER | Free Market |
| USER | Free market customers |
| FUNCTIONAL REQUIREMENTS | RF1: Register products.  FR2: Order registration.  RF3: Save Data Persistently  RF4: Product Disposal.  FR5: Product Finder.  RF6: Search for orders. |
| CONTEXT OF THE PROBLEM | The company Mercado Libre needs a virtual store to be able to manage the products and orders of its customers. |
| NON-FUNCTIONAL REQUIREMENTS |  |

**Functional requirements analysis table (Note: One table for each functional requirement)**

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF1: Register Products | | |
| Summary | Enter free market products | | |
| Tickets | Entry name | Data type | Selection or repetition condition |
| name | String | Must be at least 3 characters long. |
| description | String | Must be at least 5 characters long. |
| price | Double | It must have a minimum of 2 digits. |
| quantityAvailable | Int | Must initialize to 1 |
| category | TypeCategory | Must have 10 characteristics. |
|  | salesNumber | Int | It must have 1 digit minimum. |
| General activities needed to obtain the results | The program asks the user for this necessary information (name, price, etc.) in order to add the product to the system. | | |
| Resultado o postcondición | Product added | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| checkCreationProduct | Boolean | Enter all data according to variable types |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF2: Register orders | | |
| Summary | Enter orders generated by customers. | | |
| Tickets | Entry name | Data type | Selection or repetition condition |
| sellerName | String | Must be at least 3 characters long. |
| productList | Product | Must have 1 minimum product. |
| purchaseDate | Calendar |  |
|  |  |  |
|  |  |  |  |
| General activities needed to obtain the results | The user adds the products they want to their shopping list | | |
| Resultado o postcondición | The order has been registered. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| confirmationOrder Creation. | Boolean | Enter all data according to variable types |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF3: Save Data Persistently | | |
| Summary | The user chooses the product they want to delete. | | |
| Tickets | Entry name | Data type | Selection or repetition condition |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
| General activities needed to obtain the results | Each order and product record will be saved in JSON file | | |
| Resultado o postcondición | Order and product data has been written to the JSON file | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF4: Product Disposal | | |
| Summary | The user chooses product that he wants to delete. | | |
| Tickets | Entry name | Data type | Selection or repetition condition |
| productList | int | The product must be on the list |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| General activities needed to obtain the results | The user enters the number that the product list is located and must confirm whether to delete that product. | | |
| Resultado o postcondición | The product has been successfully removed. | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| checkDeleteProduct | Boolean | Enter all data according to variable types |

|  |  |  |  |
| --- | --- | --- | --- |
| Name or identifier | RF5: Search for products and orders. | | |
| Summary | The user enters the data to be able to search for a product or order. | | |
| Tickets | Entry name | Data type | Selection or repetition condition |
| category | SearchCategory | Must have 1 minimum character |
| filterData | String/int | The category can ask for number or letter |
|  |  |  |
| General activities needed to obtain the results | The user must enter the categories by which the search will be done and then enter the values of each category to filter in the search | | |
| Resultado o postcondición | Products or orders have been found successfully with the search filter | | |
| Outputs | Entry name | Data type | Selection or repetition condition |
| checkFoundProduct | Boolean | Enter all data according to variable types |

**Scenario Configuration**

|  |  |  |
| --- | --- | --- |
| **Number** | **Class** | **Scenario** |
| setupStage1, | ProductTest | A object in the class Product with name=”coffe”, description=”quallity 100% natural”, price=25, quantityAvailable=1, Category=”eats and drinks” |
| setupStage2, | ProductTest | A object in the class Order with name sellerName” carlos”, numberOrder=1 productList=”coffe” |
| setupStage3, | SaveDateTest | A object in the class Product with name=”milksnake”, description=” chocolate drink”, price=35, avaibleContent=2, Category=”eats and drinks”  A object in the class Order with name sellerName=”carlos”, productList=”milksnake, coffe” |
| setupStage4, | DeleteTest | A object in the class Product with name=”milk”, description=”100 ml”, price=23, avaibleContent=2, Category= ”eats and drinks”  A object in the class Order with name sellerName” carlos”, productList=”coffe, milksnake, milk” |
| setupStage5 | SearchTest | A object in the class Product with name=”chair”, description=” reinforced plastic”, price=575, quantityAvailable=5, Category=”clothes and accessories”  A object in the class Producto with name=”vodka”, description=”25% alcohol”, price=175” quantityAvailable=20, Category=” drinks”  A object in the class Product with name=”chair”, description=” reinforced plastic”, price=575 quantityAvailable=5, Category=” clothes and accessories”  A object in the class Product with name=”beer”, description=”7% alcohol”, price=125, quantityAvailable=12, Category=”eats and drinks”  A object in the class Product with name=”meat”, description=” Pork Meat”, price=273, quantityAvailable=8, Category=”eats and drinks”  A object in the class Order with name buyerName” daniel”, productList=”meat, coffe, beer, chair, vodka” |

**Test Case Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:**  to verify the methods of addProduct, creating the Product object in the Seller class and adding a product already created in the Order class And verify that the addOrder method of the Buyer class | | | | |
| **Class** | **Method** | **Scenario** | **Input Values** | **Expected result** |
| Seller | addProduct | SetupStage1 | name=”meat”, description=” Pork Meat”, price=273, quantityAvailable=8, Category=”eats and drinks” | The product has been added successfully. |
| Seller | addProduct | SetupStage1 | name=88, description=” Pork Meat”, price=273, quantityAvailable=28, Category=”eats and drinks” | An exception is thrown and the message is the name is invalid, no less than 3 characters are allowed. |
| Seller | addProduct | SetupStage1 | name=878, description=” Pork Meat”, price=273, quantityAvailable=28, Category=”eats and drinks” | An exception is thrown and the message is the name is invalid, no numbers allowed. |
| Seller | addProduct | SetupStage1 | name=”meat”, description=” Pork Meat”, price=273, quantityAvailable=8, Category=”eats and drinks” | An exception is thrown and the message is the price is not valid for having less than 2 digits. |
| Seller | addProduct | SetupStage1 | name=”meat”, description=” Pork Meat”, price=273, quantityAvailable=-88, Category=”eats and drinks” | An exception is thrown and the message is the price is invalid for negative values. |
| Seller | addProduct | SetupStage1 | name=”meat”, description=” Pork Meat”, price=“273”, quantityAvailable=28, Category=”hot-dog” | An exception is thrown and the message is that the category was not found |
| Seller | addProduct | SetupStage1 | name=”meat”, description=” Pork Meat”, price=273, quantityAvailable=28, Category=”eats” | The price of the product is 8. Throwing an exception shows us that a number value cannot be entered. |
| Order | addProduct | SetupStage2 | String=coffe | Product added successfully |
| Order | addProduct | SetupStage2 | String=beer | An exception is thrown and the message is the product does not exist. |
| Buyer | createOrder | SetupStage2 | SellerName=felipe, productList="coffe"  NumberOrder=1 | Order added successfully |
| Buyer | createOrder | SetupStage2 | SellerName=23, productList=”meat”  NumberOrder=1 | An exception is thrown and the message is that the product cannot be added because it does not exist |
| Order | addProduct | SetupStage1 | Product = “coffe” |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective of the Test:** to verify if the writing and reading is executed successfully. | | | | |
| **Class** | **Method** | **Scenario** | **Input Values** | **Expected result** |
| Controller | saveData | SetupStage1 | Product = coffe | Successfully saved |
| Controller | saveData | SetupStage2 | Product = meat | Throws an exception and the message is that it has not been saved because the object does not exist |
| Controller | saveData | SetupStage3 | productList = coffe, milksnake | Saved successfully |
| Controller | saveData | SetupStage3 | ProductList = coffe, milksnake, meat | An exception is thrown and the message the product "meat" cannot be saved because it does not exist |
| Controller | saveData | SetupStage3 | Order = “carlos” | Saved successfully |
| Controller | saveData | SetupStage3 | Order=1 | An exception is thrown and the message cannot be added because numbers are not allowed |
| Controller | saveData | SetupStage3 | Order=”Raul” | An exception is thrown and the message says you can't save because Carlos' order doesn't exist |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objective of the Test:** Validate if the elimination of a product has been carried out | | | | |
| **Class** | **Method** | **Scenario** | **Input Values** | **Expected result** |
| Controller | DeleteProduct | SetupStage4 | Product=”milk” | Successfully deleted |
| Controller | DeleteProduct | SetupStage4 | Product=”beer” | Throws an exception where this product could not be found to remove it |
| Controller | DeleteProduct | SetupStage4 | Product=1 | Throw an exception where you leave a message that numbers are not accepted. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Objective:** Find the product, order with the searchProduct method and searchOrder | | | | |
| **Class** | **Method** | **Scenario** | **Input Values** | **Expected result** |
| Controller | searchProduct | SetupStage5 | String = “chair” | Product found |
| Controller | searchProduct | SetupStage5 | String = “milk” | An exception is thrown and the message says the product was not found |
| Controller | searchOrder | SetupStage5 | name= “Daniel”, numberOrder = 1 | Order found |
| Controller | searchOrder | SetupStage5 | Name=”Daniel”, numberOrder=2 | An exception is thrown and the message says the order was not found |
| Controller | searchOrder | SetupStage5 | Name= “carlos”, numberOrder=1 | An exception is thrown and the message says buyer not found |