**REQUIREMENTS TABLE INTEGRATIVE TASK 2**

**Crew:**

Felipe Rojas Prado A00393918

Juan Sebastián González A00371810

Juan Felipe Jojoa Crespo A00382042

| Client | Mercado Libre |
| --- | --- |
| User | Administrator of the platform |
| Functional requirements | R\_1: The system should be able to add products to the store.  R\_2: The system should constantly update the quantity of each product.  R\_3: The system should have a product search function.  R\_4: The system should have an order search function.  R\_5: The System should allow the creation of a order  R\_6: The system will allow to create .JSON file after closing the program.  R\_7: The system should allow the creation of a product list. |
| Problem context | * MercadoLibre wants to develop an application to allow the online sale of its virtual store. The user of the application would be the administrators of the MercadoLibre virtual store, who would be responsible for entering information about the products, managing inventory, registering orders, and searching for products and orders on the platform. * The objective of the application is to improve the online shopping experience for MercadoLibre customers, allowing efficient inventory management and order registration. In addition, the application must have features such as data persistence and exception handling to ensure the quality and reliability of the platform. |
| Non-functional requirements | * Guarantee the integrity of the data. * Handle exceptions to avoid unexpected states. |

| Name or identifier | R1 | | |
| --- | --- | --- | --- |
| Abstract | The system should be able to add products to the store. | | |
| Inputs | Input name | Data type | Condition of select or repetition |
| productName | String | Can’t be null |
| productDescription | String | Can’t be null |
| productPrice | Float | Can’t be null or less than 0 |
| productQuantity | Integer | Can’t be null or less than 0 |
| productCategory | Category |  |
| General activities needed to obtain the results | 1. User selects the "Add Product" option. 2. User enters the name, price, and quantity of the product. 3. System saves the product information to the database. | | |
| Result or postcondition | The product is successfully added to the store. | | |
| Outputs | Output name | Data type | Condition of select or repetition |
| MsgConfirmation | String |  |

| Name or identifier | R2 | | |
| --- | --- | --- | --- |
| Abstract | The system should constantly update the quantity of each product. | | |
| Inputs | Input name | Data type | Condition of select or repetition |
| productID | String |  |
| quantity | Integer |  |
| General activities needed to obtain the results | 1. After order is taken, search for product ID 2. Decrease the quantity by the amount sold in order | | |
| Result or postcondition | The quantity of the product in the inventory is decreased after the sale. | | |
| Outputs | Output name | Data type | Condition of select or repetition |
| MsgSold | String |  |

| Name or identifier | R3 | | |
| --- | --- | --- | --- |
| Abstract | The system should have a product search function. | | |
| Inputs | Input name | Data type | Condition of select or repetition |
| productName | String |  |
| productID | Integer |  |
| productCategory | String |  |
| productPrice | int | Range, that cant be less than 0. |
| productSales | int | Cant be less than 0. |
| General activities needed to obtain the results | 1. User selects the search option. 2. User enters the search criteria. 3. System searches for products based on the criteria. 4. System displays a list of matching products. | | |
| Result or postcondition | A list of matching products is displayed on the screen. | | |
| Outputs | Output name | Data type | Condition of select or repetition |
| listOfProducts | String |  |

| Name or identifier | R4 | | |
| --- | --- | --- | --- |
| Abstract | The system should have an order search function. | | |
| Inputs | Input name | Data type | Condition of select or repetition |
| orderID | Integer | Can´t be null or less than 0 |
| clientName | String | Can´t be null |
| totalPrice | int | Range, can´t be null or less than 0 |
| orderDate | Date | Can´t be null |
| General activities needed to obtain the results | 1. Display a search bar for the user to enter the OrderID. 2. Retrieve the OrderID entered by the user. 3. Search for the order with the corresponding OrderID in the database. 4. Display the order details to the user. | | |
| Result or postcondition | The system displays the details of the order with the corresponding OrderID entered by the user. | | |
| Outputs | Output name | Data type | Condition of select or repetition |
| orderID | Integer | One row per product in the order. |
| customerName | String |
| customerEmail | String |
| orderDate | Date |
| productID | Integer |
| productName | String |
| quantity | Integer |
| price | Float |
| total | Float |

| Name or identifier | R5 | | |
| --- | --- | --- | --- |
| Abstract | The System should allow the creation of a order | | |
| Inputs | Input name | Data type | Condition of select or repetition |
| clientName | String |  |
| productList | Json/ArrayList | Adds, or creates product list |
| General activities needed to obtain the results | 1. User selects the option to enter an order of purchase. 2. Inputs client name 3. Inputs already created productList or creates new one | | |
| Result or postcondition | An order of purchase is created in the system autofilling the total price and date. | | |
| Outputs | Output name | Data type | Condition of select or repetition |
| orderID | String | Order format |
| clientName | String |
| productList | Json/ArrayList |
| totalPrice | int |
| orderDate | Date |

| Name or identifier | R6 | | |
| --- | --- | --- | --- |
| Abstract | The system should allow creating a .JSON file after closing the program. | | |
| Inputs | Input name | Data type | Condition of select or repetition |
| None. | | |
|
|
| General activities needed to obtain the results | 1. Wait for the user to close the program. 2. Generate the .JSON file using the data stored in the program. | | |
| Result or postcondition | The system generates a .JSON file containing the data of the store and the transactions made. | | |
| Outputs | The .JSON file containing the data of the store and the transactions made. | | |
|

| Name or identifier | R7 | | |
| --- | --- | --- | --- |
| Abstract | The system should allow the creation of a product list. | | |
| Inputs | Input name | Data type | Condition of select or repetition |
| productID | String |  |
| quantity | Integer |  |
| General activities needed to obtain the results | 1. User selects the option to enter an order of purchase. 2. User creates productList 3. User inputs the product ID and the quantity of the product to be purchased. 4. System checks that the product ID exists and the quantity is greater than zero. 5. System updates the inventory of the product by subtracting the purchased quantity. 6. System creates an order with a unique order ID and saves it to the system. 7. System returns the order ID, product ID, and quantity of the purchased product to the user as confirmation. | | |
| Result or postcondition | An order of purchase is created in the system. | | |
| Outputs | Output name | Data type | Condition of select or repetition |
| productListID | String |  |
| productID | String | One per item |
| quantity | Integer |