|  |  |  |
| --- | --- | --- |
| Use case name | createSale | |
| Actors | Cashier | |
| Pre-conditions | Customer and Item(s) has to exist in the system, Employee is logged in | |
| Post-conditions | Sale is registered in the system | |
| Frequency | As many times as needed | |
| Main Success Scenario | 1. Cashier types in the customer’s name to start the sale. | 2. The system asks for an item’s barcode and quantity. |
|  | 3. Cashier types in the product’s name, quantity and the same attributes of all the products the customer wants to buy. | 4. The system shows the price of each product purchased and updates total amount. |
|  | 5. Cashier clicks that all items were added | 6. The system asks for the payment |
|  | 7. Cashier chooses the payment type | 8 The system shows that the sale has ended |
| Alternate Flows | 1a. The customer doesn’t exist in the system  3a. The product with that name doesn’t exist  3b. The product with that name is out of stock | |

***Operation Contracts***

Operation: startSale(name);

Use case: createSale;

Pre-condition: Employee is logged in, Customer exists in the system;

Post condition:

* Employee object e was associated with employee
* Customer object c was created;
* c.name became name;

Operation: addSaleLine(name, quantity)

Use case: createSale;

Pre-condition: Item(s )has to exist in the system;

Post condition:

* SaleLine object sl was created;
* sl.name became name
* sl.quantity became quantity
* sl was added to an ArrayList<SaleLine>

Operation: finishSale( e, isPaid)

Use case: createSale;

Pre-condition: ArrayList<SaleLine> sl has to have been created, Employee e and Customer c hast to exist in the system.

Post condition:

* A Sale object s was created;
* s.paid became paid
* s was associated with employee;
* s was associated with item;
* s was associated with customer;