|  |  |  |
| --- | --- | --- |
| 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 id to start the sale. | 2. The system asks for an item’s barcode and quantity. |
|  | 3. Cashier types in the product’s barcode, 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 type |
|  | 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 barcode doesn’t exist  3b. The product with that barcode is out of stock | |

***Operation Contracts***

Operation: startSale();

Use case: createSale;

Pre-condition: Employee is logged in;

Post condition:

* ArrayList<SaleLine> saleLines was created;

Operation: addSaleLine(quantity, barcode)

Use case: createSale;

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

Post condition:

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

Operation: insertSale(saleNr, isPacked, datePacked, isSent, dateSent, isPaid, datePaid, id\_employee, id\_customer, saleLines)

Use case: createSale;

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

Post condition:

* Employee object employee was created;
* employee.id\_employee became id\_employee;
* Customer object customer was created;
* customer.id\_customer became id\_customer;
* Sale object s was created;
* sale.saleNr became saleNr;
* sale.isPacked became isPacked;
* sale.datePacked became datePacked;
* sale.isSent became isSent;
* sale.dateSent became dateSent;
* sale.isPaid became isPaid;
* sale.datePaid became datePaid;
* sale was associated with employee;
* sale was associated with customer;
* sale was associated with an ArrayList<SaleLine> saleLines;