

# Project Description

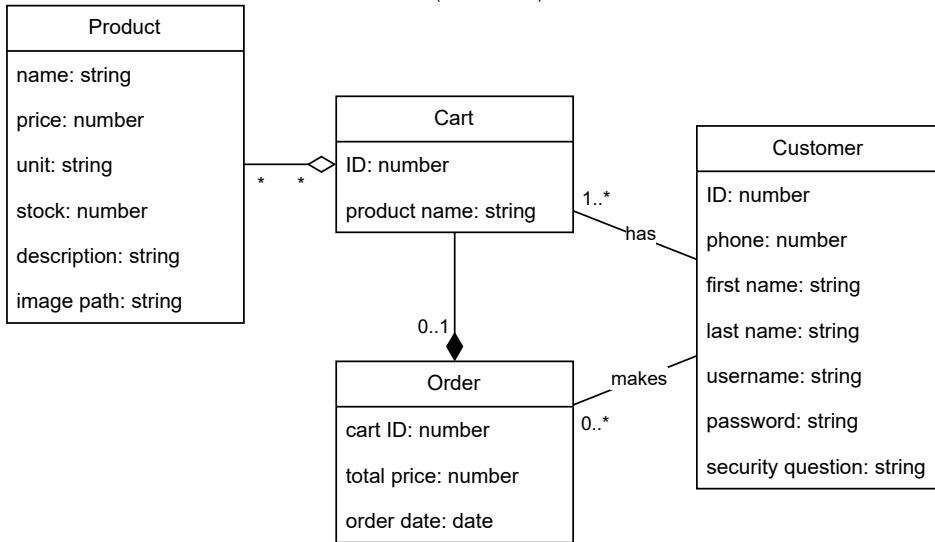
VESO

**Title:** VESO

**Description:** Veso is an application that aims to facilitate grocery shopping for customers. It eliminates the disadvantage of crowded supermarkets — A customer can place an order from the comfort of his/her home.

## Domain Model

VESO (Online Store)



**Notes:** 1.) \* cardinality means 0..\*  
assignment's instruction also has it.

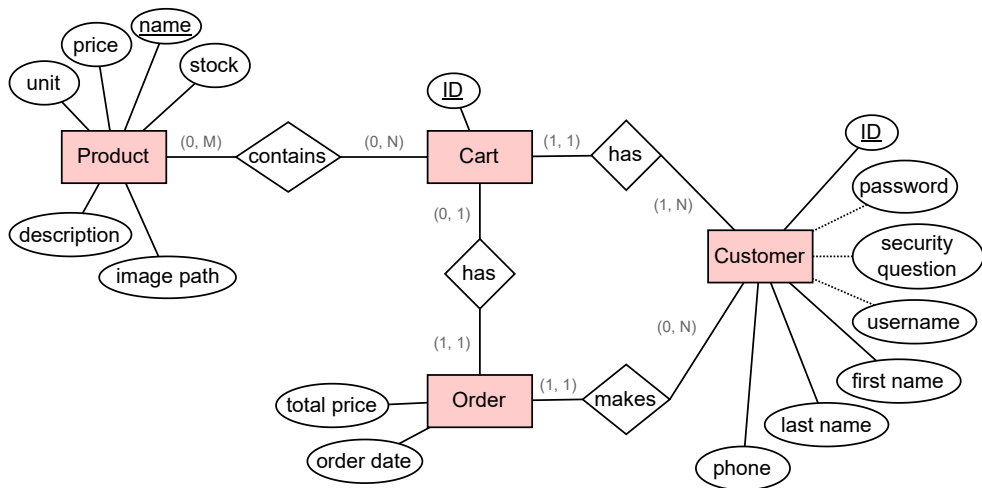
2.) No cardinality means 1..1

3.) Data types are added, because the example in

**Reading example:** A customer makes 0 or many orders; An order is made by minimum 1 and maximum 1 customer.

# Conceptual Model

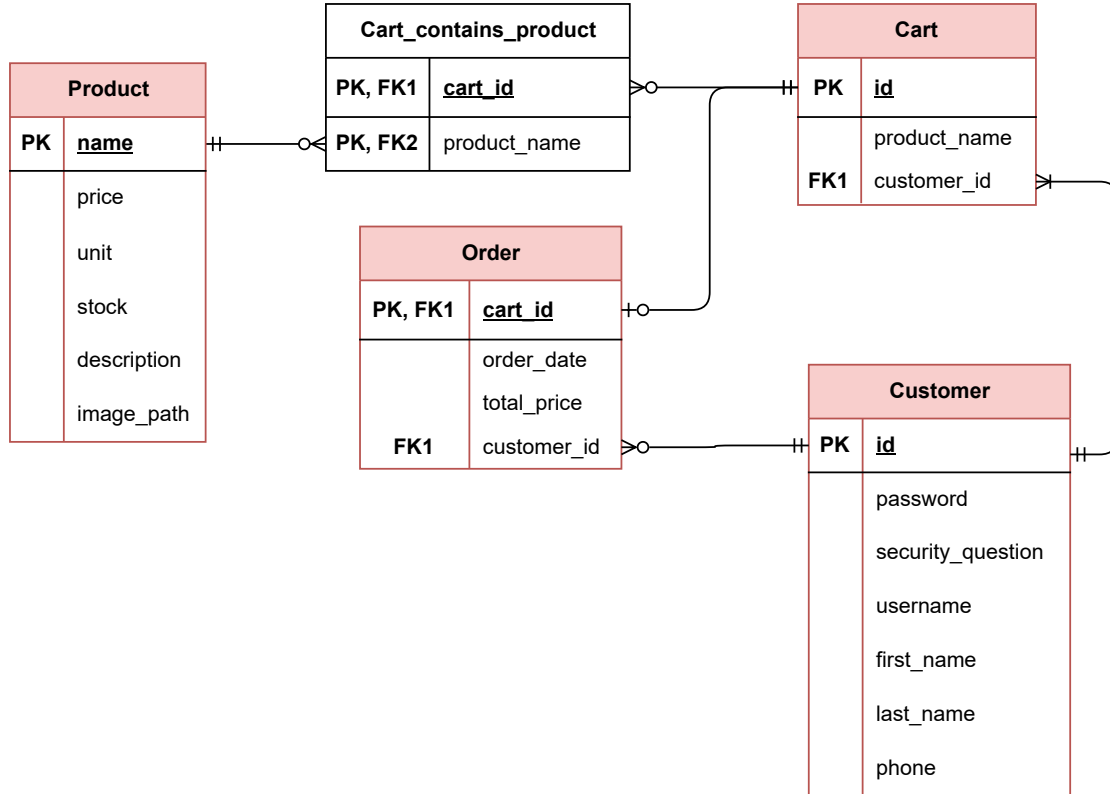
VESO (Online Store)



Reading example: A cart contains 0 to N products; A product can be contained in 0 to M carts.

## Logical Model

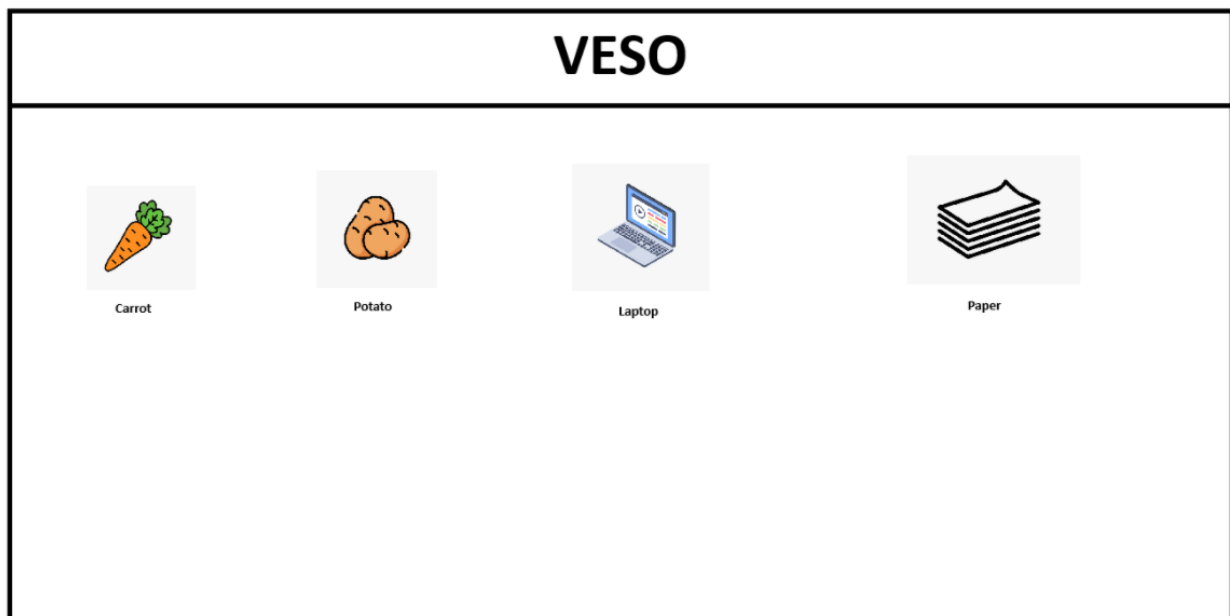
VESO (Online Store)



## Story: See Products

As a user,  
I want to see shop products,  
so that I can later choose and buy them.

## Wireframe



## Acceptance Criteria

- Shop products are displayed.
- Each product has an image, name and price.
- If a product is out of stock, that is indicated.
- Images are optimized and resized accordingly.

## Story: Add to Cart

As a user,

I want to put selected items in a cart,

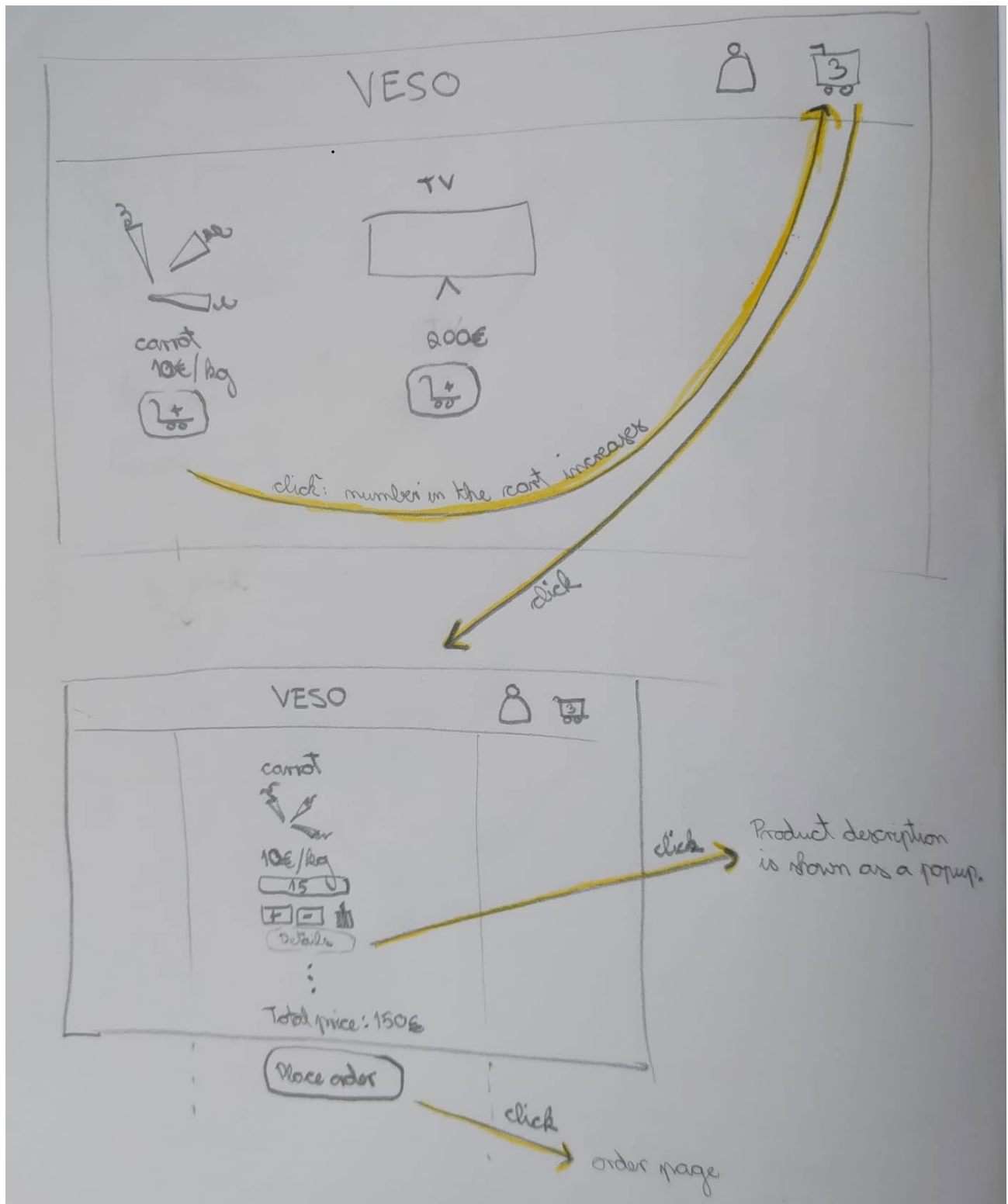
So that I can order/pay for them.

## Acceptance criteria

- A shopping cart is visible
- User can add items to the cart
- User can see items in the shopping cart
- User can remove items from the cart
- Number of items in the cart is updated accordingly (increased or decreased).

(Wireframe on the next page)

# Wireframe



## Story: Place Order (pay at the shop)

As a user,  
I want to place orders,  
So that I can pay and pick up the products.

### Wireframe

The wireframe shows a rectangular container with a header bar at the top. The header bar contains the text "VESO" in a bold, sans-serif font. Below the header bar is a large, rounded rectangular area containing a form. The form consists of several input fields and a button. The first three fields are stacked vertically: "First Name", "Last Name", and "Phone". Each field is a rectangular box with a label on the left and an empty space on the right. Below these three fields is a "Total Price" field, which is a rectangular box with the label "Total Price" on the left and the value "54 \$" on the right. Below the "Total Price" field is a red rectangular button with the text "Place order" in a bold, sans-serif font.

### Acceptance Criteria

- If a user is logged in, the credentials are filled automatically.
- Phone field must have a placeholder value that shows the format of the number.
- Besides the phone field, there must be an optional email field. One of these fields can be empty.
- The email or phone number format must be verified before placing an order.
- First name and last name are required.
- The order is saved in the database, after it has been placed.
- After the order has been placed, a message "Thank you for choosing VESO!" is displayed instead of the form.





## Story: Login

As a user,  
I want to login,  
So that credentials are automatically filled when ordering.

## Wireframe

VESO					
<div><table border="1"><tr><td>Username</td><td></td></tr><tr><td>Password</td><td></td></tr></table><div>Login</div></div>		Username		Password	
Username					
Password					

## Acceptance Criteria


- Both fields are required.
- Password must be at least 8 characters, containing at least one special character.
- Password cannot be changed (yet).
- After logging in, the customer is redirected to the home page with products.
- The password field must have hidden characters by default. There should be an eye icon to show the password if needed.

## Story: Search by Name


As a user,  
I want to search products by name,  
So that I can find them faster.

## Wireframe


# VESO




Carrot



Potato



Laptop



Paper

## Acceptance Criteria

- Search field is case insensitive and containing (searching for 'r' would return carrot and paper).
- Searching without inputting anything in the field does nothing.
- The search string cannot be longer than the longest possible product name in the database.


## Story: Profile


As a user,  
I want to see my profile,  
So that I know that the store has correct information about me.

## Wireframe

The wireframe shows a mobile app interface for a user profile. At the top is a header bar with the text "VESO". Below the header is a large rectangular area containing the user's profile information. The information is organized into two sections. The first section contains the "Username: Matej123" and "Password: M-@-t-e-j-1-2-3", each followed by a small edit icon (a square with a pencil). The second section contains the "First name: Matej", "Last name: Vesel", and "Phone: 00 32 234 235 234", with the phone number followed by a small edit icon.


**VESO**

Username: Matej123 

Password: M-@-t-e-j-1-2-3 

First name: Matej

Last name: Vesel

Phone: 00 32 234 235 234 

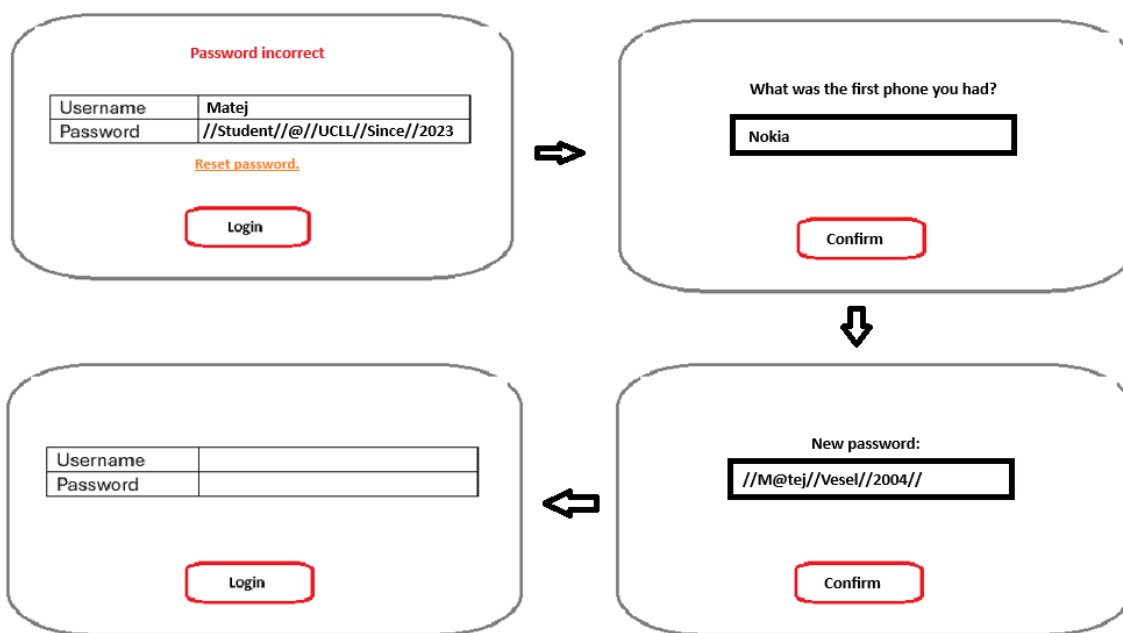
## Acceptance Criteria

- All fields except first and last name can be changed:
  - If username already exists, a message “Username already exists” is returned.
  - If password is not strong enough, a message “Weak password” is returned.
- The password field is **hidden** by default. To see or change it, one must reenter the password.

## Story: Change Password

As a user,  
I want to have an option to change the password,  
so that I do not have to worry about forgetting it.

### Wireframe



### Acceptance Criteria

- If user tries to log in and the password is incorrect, the 'Reset password' option is show.
- If the username is incorrect, this is shown by display the message: "User does not exist."
- The password is reset using the **security question** of the profile with his name.
- The answer to the security question is checked case insensitively.
- The new password must conform to all the rules.