

LAB REPORT

Dennis Loska, Tony Dorfmeister, Thi Ai Nhi Dong

Task 1: Creating Test Data

To create active records objects for the test data we add:

```
p1 = Pumpkin.create(species: 'Kuerbis', carved: false, size: 10, weight: 50, price: 450)
```

The seed data is added to a new file. This active record objects are then prepared for the pull by creating an own branch for them. To ensure the seeds are correctly in place, we run this on the terminal:

```
rake db:seed
```

Task 2: Show the Customer with its Name on Order pages

We've set up the 1:n association between Customer and Order, and the associated Customer is shown on the order#index and the order#show page, but currently with the object id. On these two pages, show the associated customer with its name.

Following the MVC principle, we checked the attributes of the object customer in model before going to view. Here we added a reference to name in the order#index and order#show files.

```
<td><%= order.customer.name %></td>
<td><%= link_to 'Show', order %></td>
<td><%= link_to 'Edit', edit_order_path(order) %></td>
<td><%= link_to 'Destroy', order, method: :delete, data: { confirm: 'Are you sure?' } %></td>
```

Task 3: Show all orders for a customer

Add a list of all orders placed by the customer on the page that shows a single customer. Hint: there is already a template showing a list of orders in orders#index. You can copy and use/modify that. As the order doesn't contain any data itself, you could just show the created_at date.

To show all orders of a single customer, we added this line to the view:

```
<% @customer.orders.each do |order| %>
```

Task 4: Compute the total of an order and the sum of all orders.

Compute the total of an order and show it on the order page. Where should the computations be done, what goes in the controller, what changed need to be made to the view? Compute the total of all orders for a customer and show it on the customer page.

A method has to be defined in model of customer, which adds the totalSum of each order to sumOfAllOrders. In views, we defined how the orders are shown to the user as shown previously.

The method in the model of orders calculates the total sum which is defined of the price of each item. With this line, we then show the total sum of the order.

```
+<p>
+  <strong>Total Price:</strong>
+  <%= @order.totalSum %>
+</p>
+
```

Various Code Snippets which are mentioned in the Report

```
def sumOfAllOrders
  sum = 0
  orders.each do |order|
    sum += order.totalSum
  end
  return sum
end
```

Image 1 Added to the Customer Model

```
def totalSum
  total = 0
  line_items.each do |item|
    total = total + item.price
  end
  return total
end
```

Image 2 Part of the Order Model

```

+<table>
+  <thead>
+    <tr>
+      <th>Customer</th>
+      <th colspan="3"></th>
+    </tr>
+  </thead>
+
+  <tbody>
+    <% @customer.orders.each do |order| %>
+      <tr>
+        <td><%= order %></td>
+        <td><%= link_to 'Show', order %></td>
+        <td><%= link_to 'Edit', edit_order_path(order) %></td>
+        <td><%= link_to 'Destroy', order, method: :delete, data: { confirm: 'Are you sure?' } %></td>
+      </tr>
+    <% end %>
+  </tbody>
+</table>
+
+<p>
+  <strong>Total Sum:</strong>
+  <%= @customer.sumOfAllOrders %>
+</p>
+

```

Image 3 Added to View