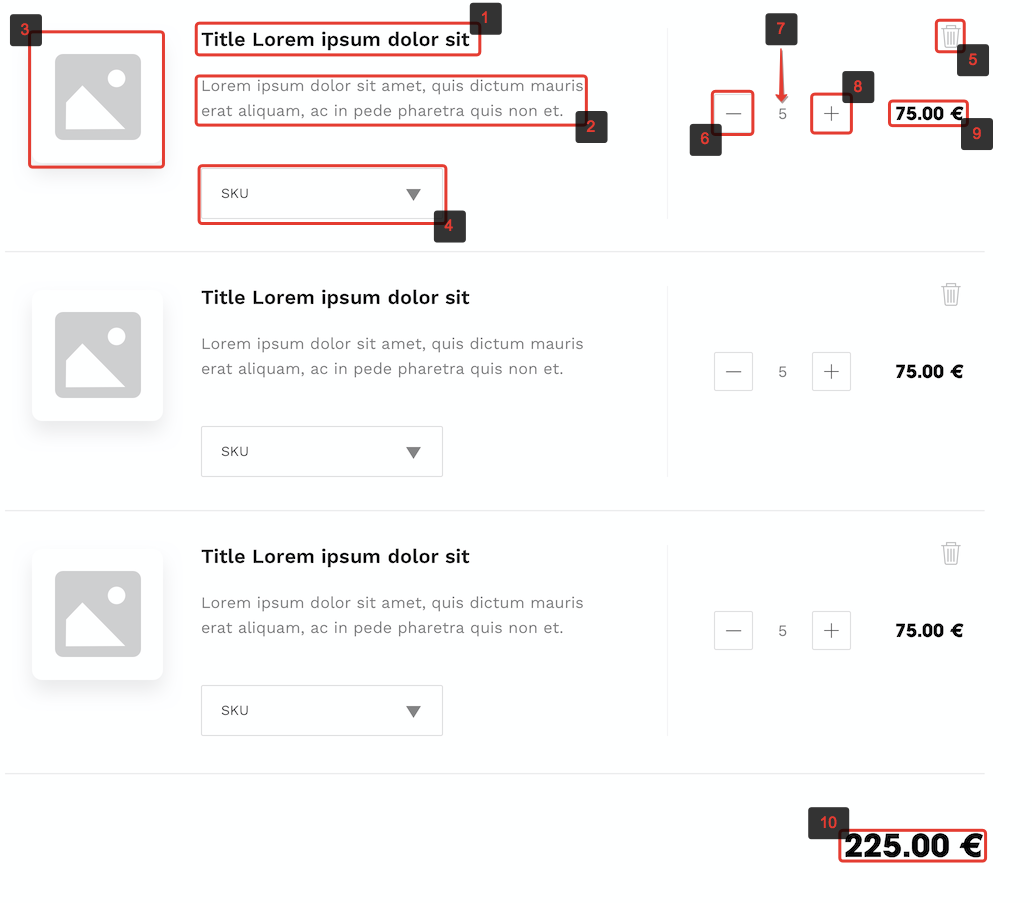
**Specification and design:**



|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Items** | **Description** | **Actions** |
| 1 | Title | Static text | - |
| 2 | Subtitle | Static text | - |
| 3 | Image | Image of active sku | **onHover**: changes opacity to 0.8 |
| 4 | Sku change select | Custom select with list of available product skus | **onChange**: updates item active sku |
| 5 | Delete button | Static button | **onHover**: changes opacity to 0.8  **onClick**: deletes product from cart |
| 6 | Decrease quantity button | Button with active and disabled states | **onHover**: changes opacity to 0.8  **onClick**: decreases quantity; if new quantity equals to sku min quantity, button becomes disabled |
| 7 | Quantity | Dynamic number displaying selected quantity of products | - |
| 8 | Increase quantity button | Button with active and disabled states | **onHover**: changes opacity to 0.8  **onClick**: increases quantity; if new quantity equals to sku max quantity, button becomes disabled |
| 9 | Item price | Dynamic text displaying total price of cart item. Equals active sku price \* quantity. | - |
| 10 | Total price | Dynamic text displaying total cart price. | - |

Each sku has it’s own specific options that should update application state on sku change:

1. Label (used in select)
2. Image
3. Price
4. Minimum quantity
5. Maximum quantity

**Browser compatibility:**

Latest versions of Chrome, Safari, FF, IE.

**Task:**

Single page cart application should be powered by FE framework and fetch, sync data with mock API of your choice (for example <http://www.mockapi.io>). Mock API structure should be created according to specification.

Development files and working production build should be included.  
  
**Login:** **feadx10@adyax.com**

**Pass:** **test[adyax]**

Design: Adyax Forms Task

App: https://zeplin.io/support.html

The same design is available without signing in to the app via the web interface - https://app.zeplin.io/ (using any OS and browser)