Exercise 2 – Globalization

Objectives

The main purpose of this assignment is to learn how to develop a web application with support for multiple languages and countries (locations).

This involves i18n training in Java and JSTL as well as training in how i18n can be implemented in a MVC architecture. In addition, you will get trained on the different mechanisms for managing user preferences, such as HTTP headers and cookies. Finally, there are elements in the exercise that is intended to be a training on developing customized tag libraries.

About this exercise

This assignment is divided into three parts and should be done in sequence.

1. The first part (Part 2.1) is about globalization in Java.
2. The second part (Part 2.2) is about globalization in a web application using Java and Java Servlets.
3. The third and last part (Part 2.3) is about JSTL and customized tags.

The whole assignment shall be submitted on Canvas in groups of 1-5 people.

The submission shall contain a zip file with the following content.

- A pdf with the names of all team members
- Screenshots of the running application
- Source Code for both non web application and we application

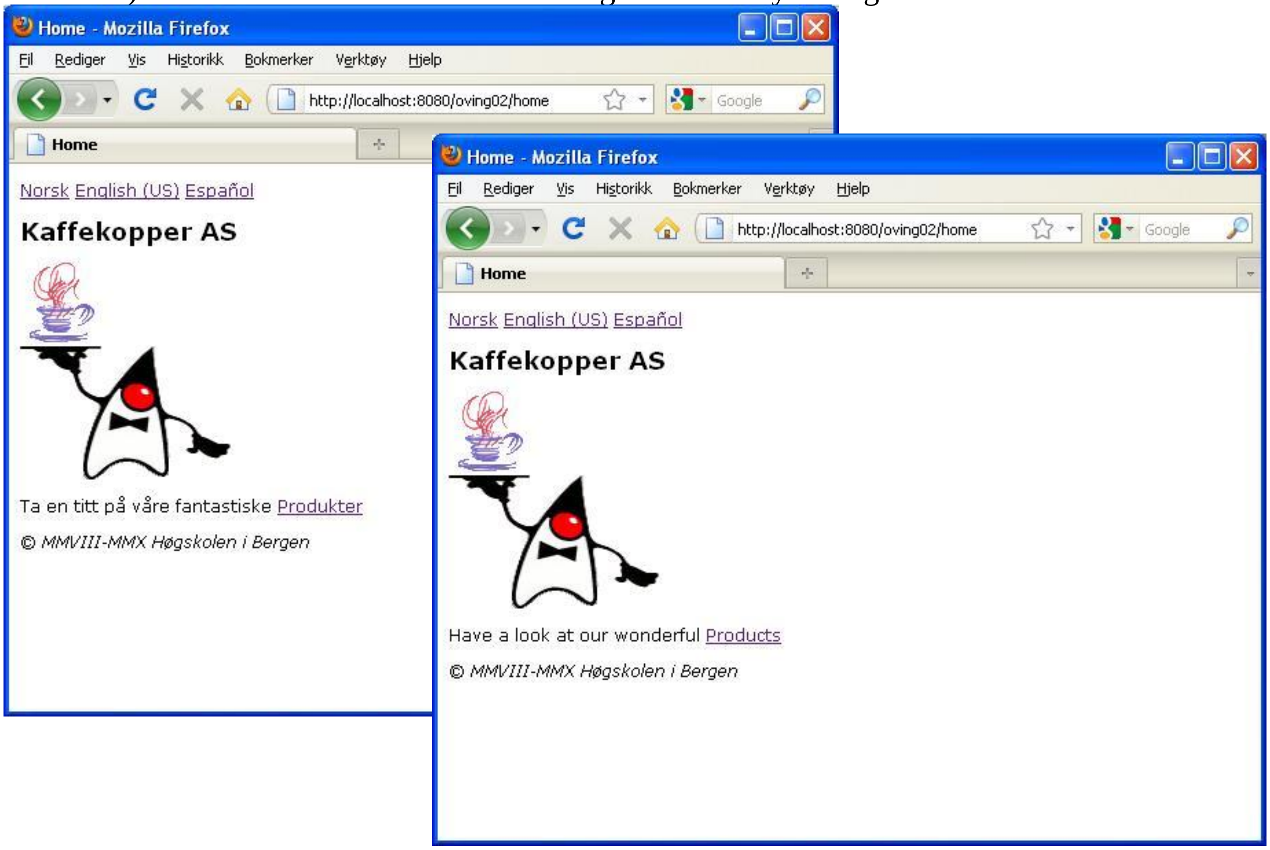
Deadline: Monday 3. October 2022.

Problem description for all three parts

You are going to create a small web application that contains some elements similar to a web store.

The application has three pages, see the examples below for views in different languages:

Home (/home)
(See examples of Norwegian and American - examples from 2010, therefore MMX and not MMXXII) and back in 2010 the name was Bergen University College.


Home - Mozilla Firefox
Fil Rediger Vis Historikk Bokmerker Verktøy Hjelp
http://localhost:8080/oving02/home
Google
Home

Norsk English (US) Español

Kaffekopper AS

Ta en titt på våre fantastiske Produkter

© MMVIII-MMX Høgskolen i Bergen

Home - Mozilla Firefox
Fil Rediger Vis Historikk Bokmerker Verktøy Hjelp
http://localhost:8080/oving02/home
Google
Home

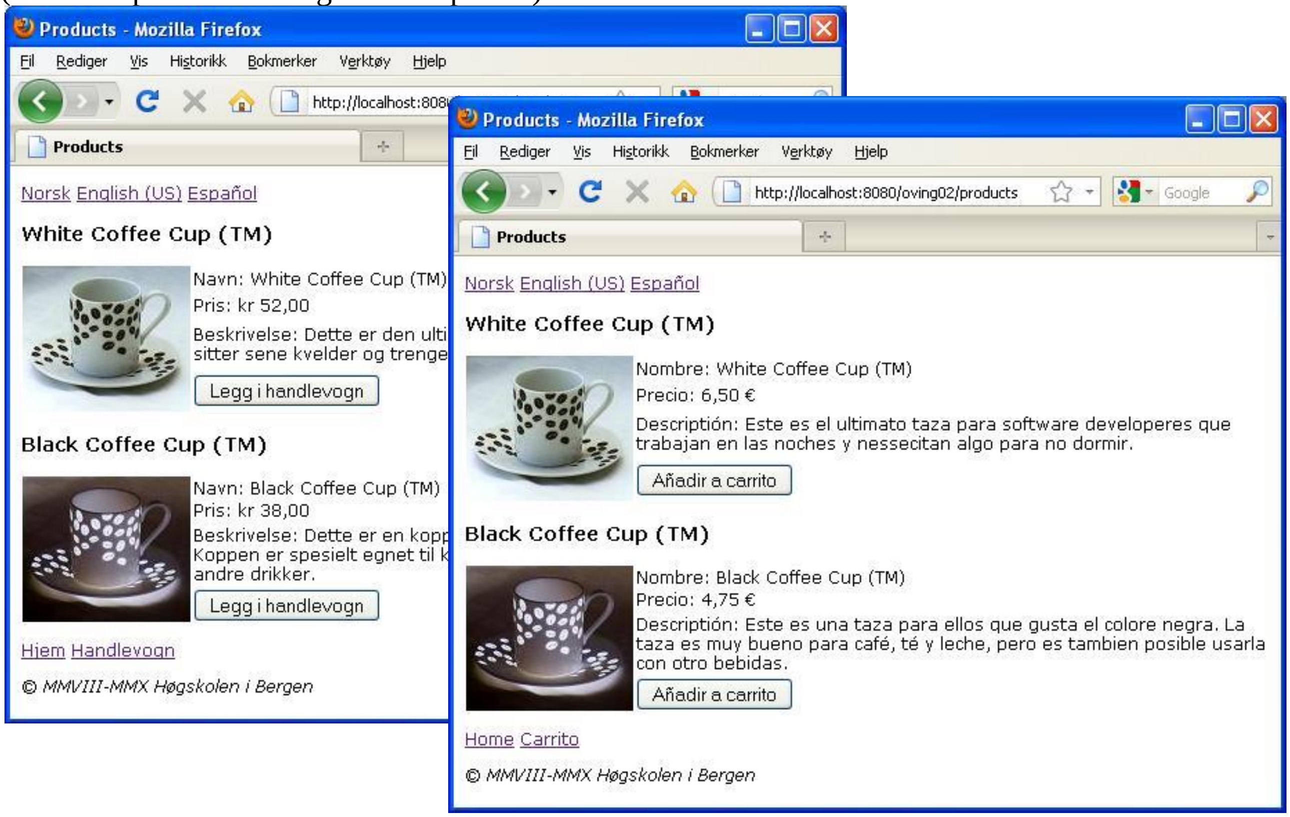
Norsk English (US) Español

Kaffekopper AS

Have a look at our wonderful Products

© MMVIII-MMX Høgskolen i Bergen

Product Page (/products)
(See examples of Norwegian and Spanish)


Products - Mozilla Firefox
Fil Rediger Vis Historikk Bokmerker Verktøy Hjelp
http://localhost:808
Products
http://localhost:8080/oving02/products
Google

Norsk English (US) Español

White Coffee Cup (TM)

Navn: White Coffee Cup (TM)
Pris: kr 52,00
Beskrivelse: Dette er den ulti
sitter sene kvelder og trenge

Legg i handlevogn

Black Coffee Cup (TM)

Navn: Black Coffee Cup (TM)
Pris: kr 38,00
Beskrivelse: Dette er en kopp
Koppen er spesielt egnet til k
andre drikker.

Legg i handlevogn

Hjem Handlevogn

© MMVIII-MMX Høgskolen i Bergen

Norsk English (US) Español

White Coffee Cup (TM)

Nombre: White Coffee Cup (TM)
Precio: 6,50 €
Descripción: Este es el ultimato taza para software developeres que trabajan en las noches y nessecitan algo para no dormir.

Añadir a carrito

Black Coffee Cup (TM)

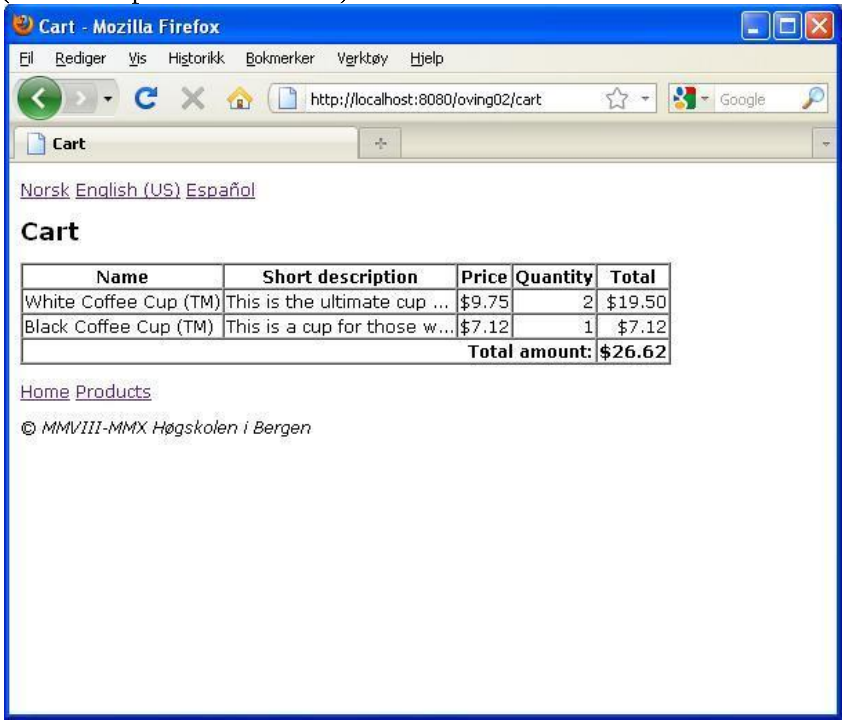
Nombre: Black Coffee Cup (TM)
Precio: 4,75 €
Descripción: Este es una taza para ellos que gusta el colore negra. La taza es muy bueno para café, té y leche, pero es tambien posible usarla con otro bebidas.

Añadir a carrito

Home Carrito

© MMVIII-MMX Høgskolen i Bergen

Cart (/cart)
(See example of American)


Cart - Mozilla Firefox
Fil Rediger Vis Historikk Bokmerker Verktøy Hjelp
http://localhost:8080/oving02/cart
Google
Cart

Norsk English (US) Español

Cart

Name	Short description	Price	Quantity	Total
White Coffee Cup (TM)	This is the ultimate cup ...	$9.75	2	$19.50
Black Coffee Cup (TM)	This is a cup for those w...	$7.12	1	$7.12
			Total amount:	$26.62

Home Products

© MMVIII-MMX Høgskolen i Bergen

Data on products conceivably stored in a "database". Relational model for the database:

Product (pno, pName, priceInEuro, imageFile)
Description (pno, langCode, text), FK pno -> product.pno

Product images should be saved in a subfolder of the project so that they can be referenced relatively.

Application Requirements:

- The application must follow the MVC pattern. Try to distribute responsibilities in a correct manner between Model (Product (s) and Cart), Views (JSP), Controllers (Servlets) and classes.
- The application must support three different locations, each consisting of language + country.
- The application must be internationalized so that there is only one JSP per web page.
- The application must be internationalized so that a new place can be added without having to change the code. (Not easy to get this to 100%. Try as best you can.)
- Application texts should be stored in a resource bundle (properties files).
- The amounts must be converted from Euros (reference value) to the correct value for selected country.
- JSTL/fmt should be used to display general application texts and for formatting data.
- Initial location should be obtained from a cookie, or. HTTP header if no cookie is set.
- When you change location during a session a cookie should be sent to the client.
- It is not necessary to use a database system (RDBMS). It's okay to make a mock DAO containing hard-coded data about products and product descriptions.
- The application must know which locations that are supported, ie it must be an object that gives this information.
- Copyright-line shall be generated by a custom tag.
  Example: <T:copyright since="2008"> HVL </ T:copyright> shall give © MMVIII-MMXXII HVL (ie the "since year" and the current year shall be written in Roman numerals)
- Product descriptions in the database can be arbitrarily long. In the carts summary, we want to limit the length of these descriptions. You must create a custom tag for this. Example:
- <T:shorttext maxchars = "10"> asødlk aøsdlkaøslølskdøaksdøaksd </ T:shorttext>
- shall give asødlk aøs ... (ie 10 characters of text plus "spacde dot dot dot"). If the text is shorter than "maxchars", the entire text is returned as it is.
- Otherwise, there should be minimal cut and paste in the code.

Part 2.1

Write a Java application (not web application) for the store. The store should contain the products and cart and should be internationalized localized to three locations. No GUI (graphical user interface) is needed, use the console for user interaction. Do not create the custom tags, but you should write code for the functionality for copyright and shorttext.

Part 2.2 (lab 20. September)

Change the code from 2.1 to become a web application for the store. Do not create the custom tags.

Part 2.3 (lab 20. September)

Add the custom tags.