HYP 2012-13 Project	
Teacher name (**)	Prof. Franca Garzotto
Delivery date (appello date)	
Project title	*Furniture Manufacturing Company*
Website url: Control Panel page:	http://localhost:3000/ http://localhost:3000/controlpanel
Group member count	3
Group leader: First name, Second name, telephone number[written exam not passed]|[written exam passed at date...with mark…]	Hossein Sedighizadeh 3272480273 Written Exam passed in Midterms With mark: 24
Member 2/teacher: First name, Second name, [written exam not passed]|[written exam passed at date...with mark…]	Saeed Tajfar 3273693455 Written Exam passed in Midterms With mark: 29.5
Member 3/Teacher First name, Second name, [written exam not passed]|[written exam passed at date...with mark…]	Navid Heidari 3473280824 Written Exam passed in Midterms With mark: 17
Notes: Design marke for the group: 28.04	
Evaluation (blank space for the teacher)	

POLITECNICO
DI MILANO

Hypermedia Project

Domain : Furniture Manufacturing Company

Group Name : HSN

Sedighizadeh + 796908 (Prof.GARZOTTO)

Tajfar + 803732 (Prof.GARZOTTO)

Heidari + 798726 (Prof.GARZOTTO)

Academic year 2012-2013

Furniture Company Technical Documentation:

1 Fragment of implementation

To show the implemented fragment of the project, we mark the implemented part with red color to stress them out in C-IDM, L-IDM and P-IDM.

1.1 The topics chosen are:

Single topic: General Info About the Company(About Us), Contacts
Multiple topics: Showroom and Brand shop, Product, Product Designer
Group of topics: All Products, All Usages, All Showrooms, All Designers
Multiple Group of topics: Showroom by Location(Merged into All Showrooms), Product by Usage

1.2 Relations:

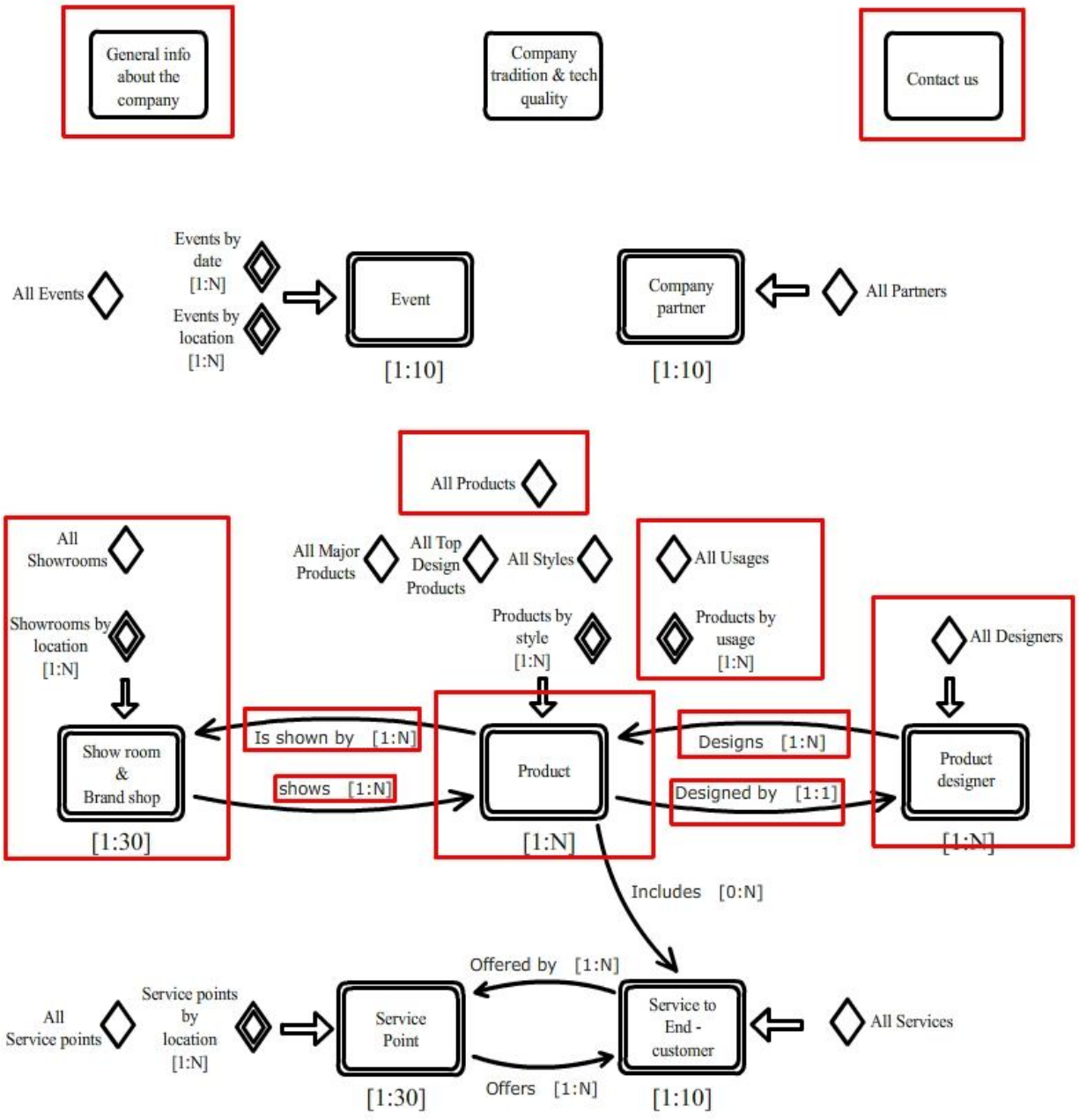
- One to one relations: Designed by (one product has exactly one designer)
- One to Many relations: Designes (one designer designes several product)
- Many to Many relations: Shows/Is shown by (one product is shown in many showrooms and one showroom shows several products. Also, an intermediate database table is implemented for many to many relationship)

1.3 Extra Pages Implemented:

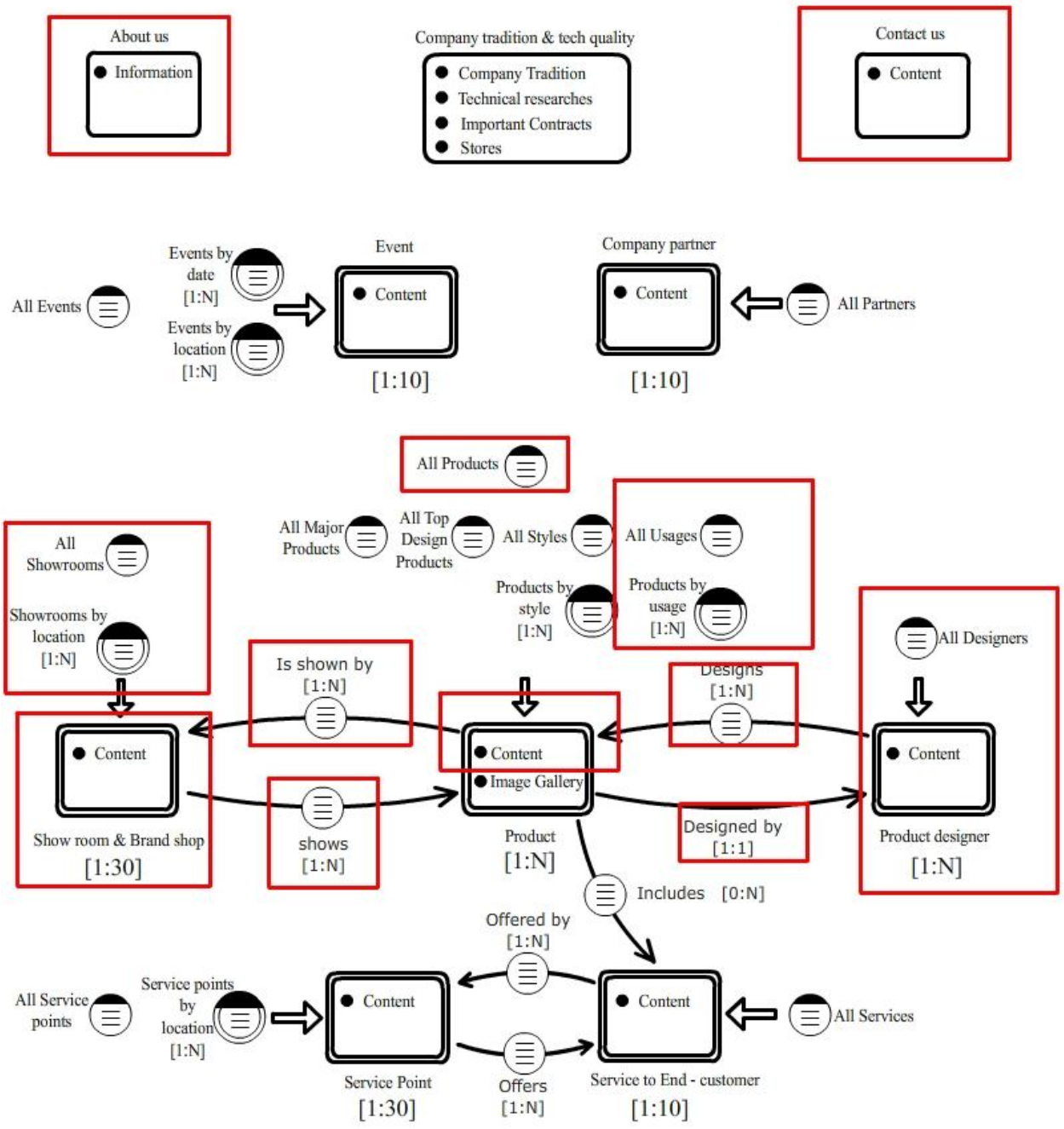
Also, some static and dynamic pages in addition to the requirements of the scenario has been implemented:

- All showroom
- Showroom by location
- All Designers
- Contact us
- Edit pages for admin:
    - Showroom
    - Product
    - Designer
    - Product Usages

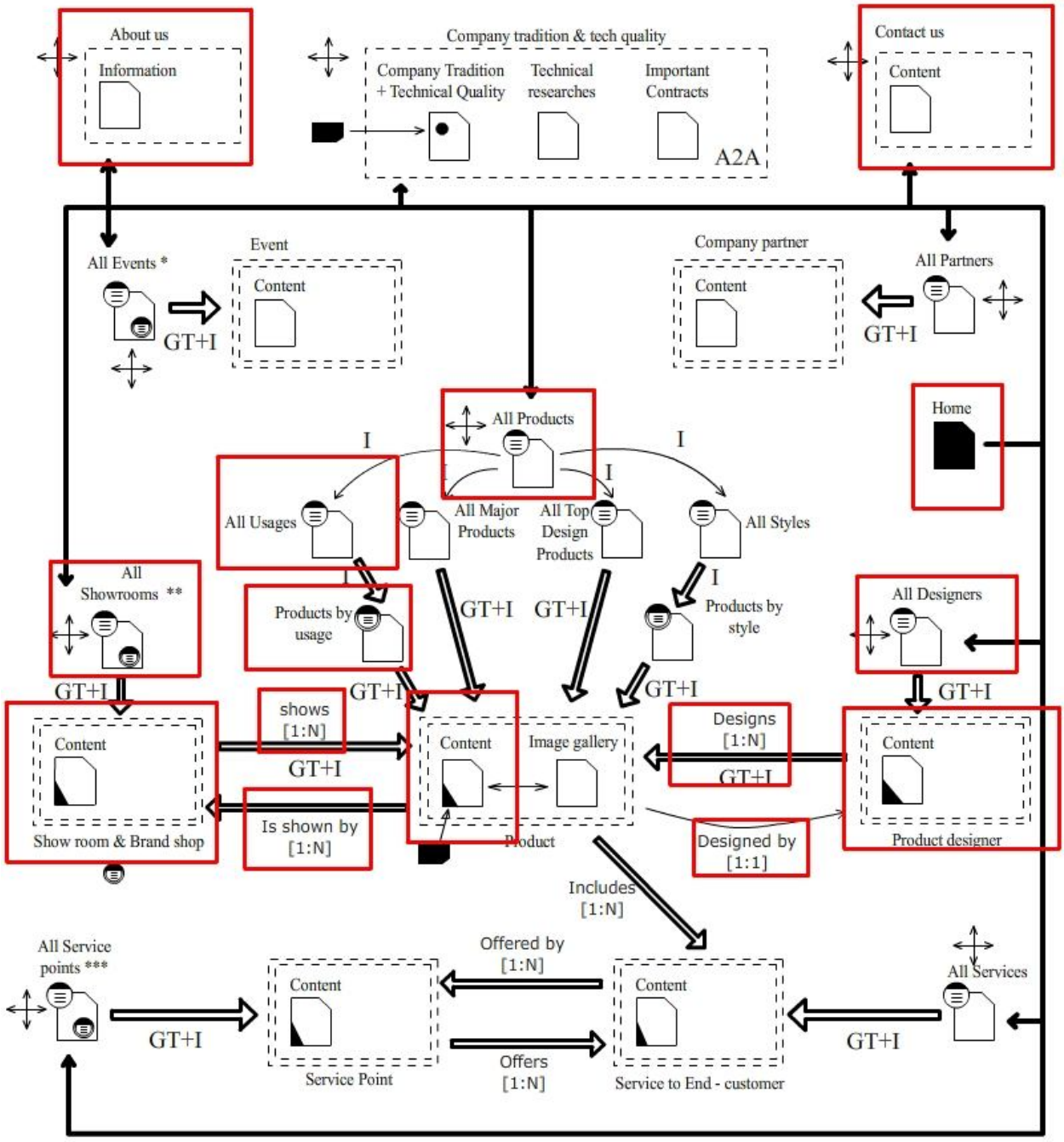
1.4 C-IDM schema:



1.5 L-IDM Schema:

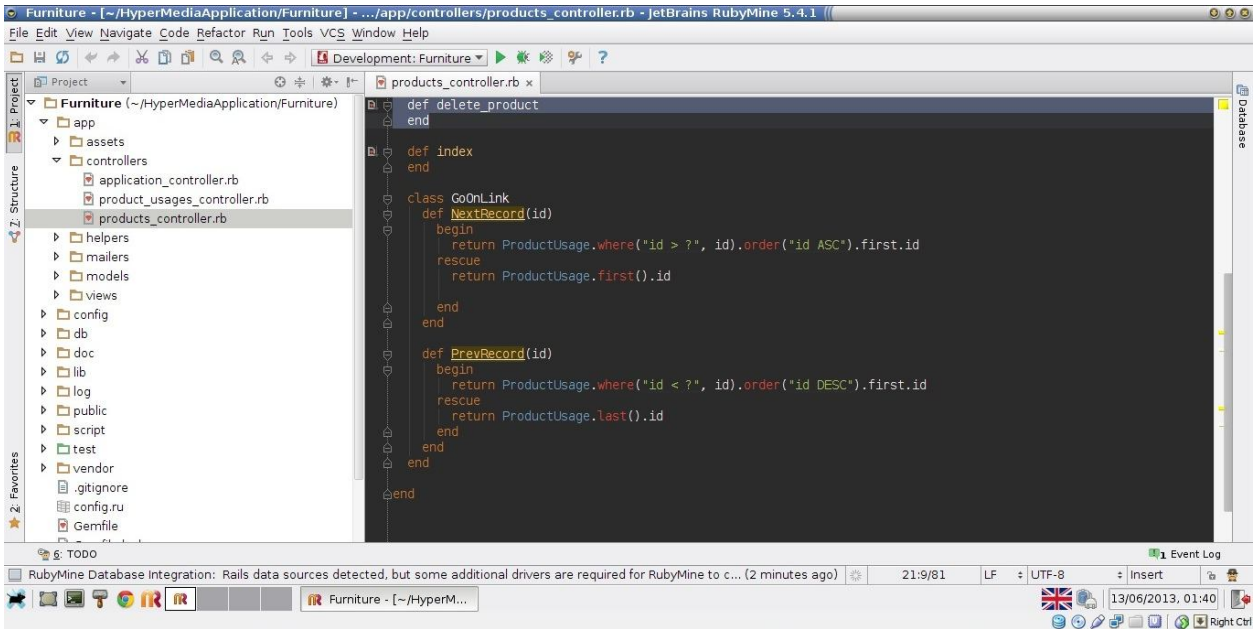


1.6 P-IDM Schema:



2 Implementing tools

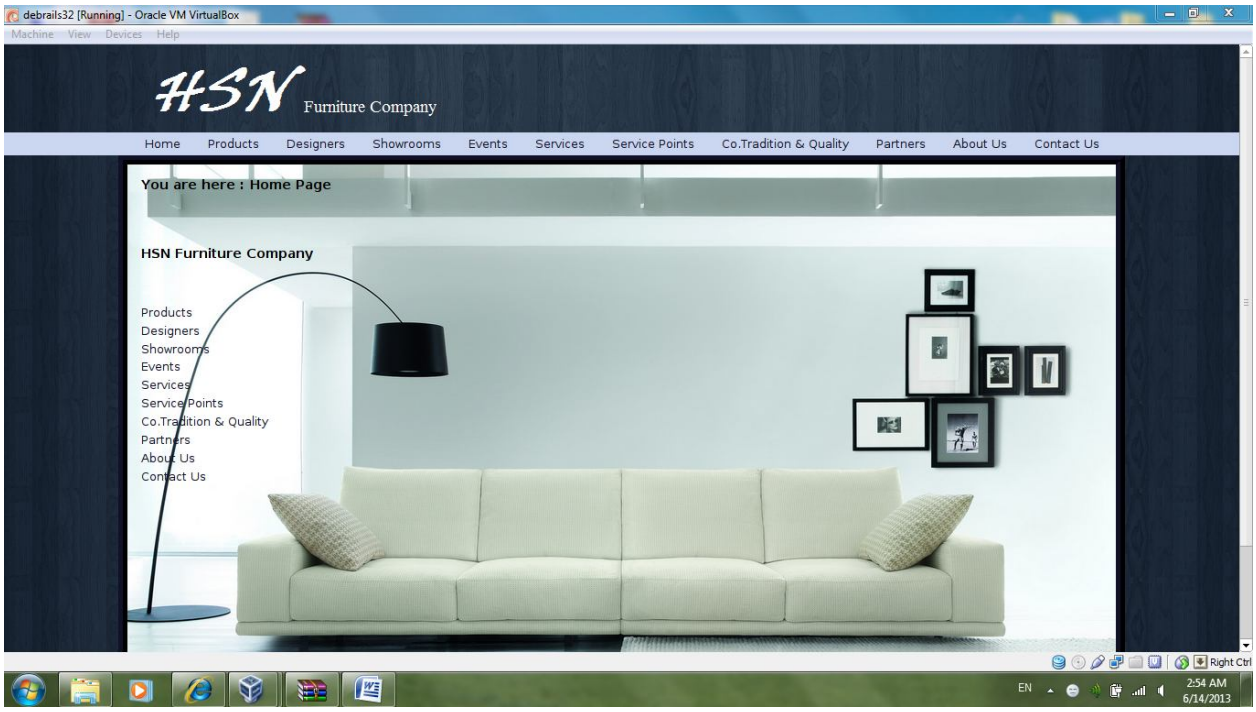
- We use Oracle VM VirtualBox as the developing platform.
- Ruby on Rails has been used as the technology for implementation of the project.
    - For any action a “controller” is defined which contian the classes and functions of a particular model
    - “Model”s are used to keep the Databse tables consistent with the classes of Ruby on Rails
    - We also used css, to achieve a better appearance and behavior.
- SQLite database is used as the repository of contents



3 Coherence with mock up

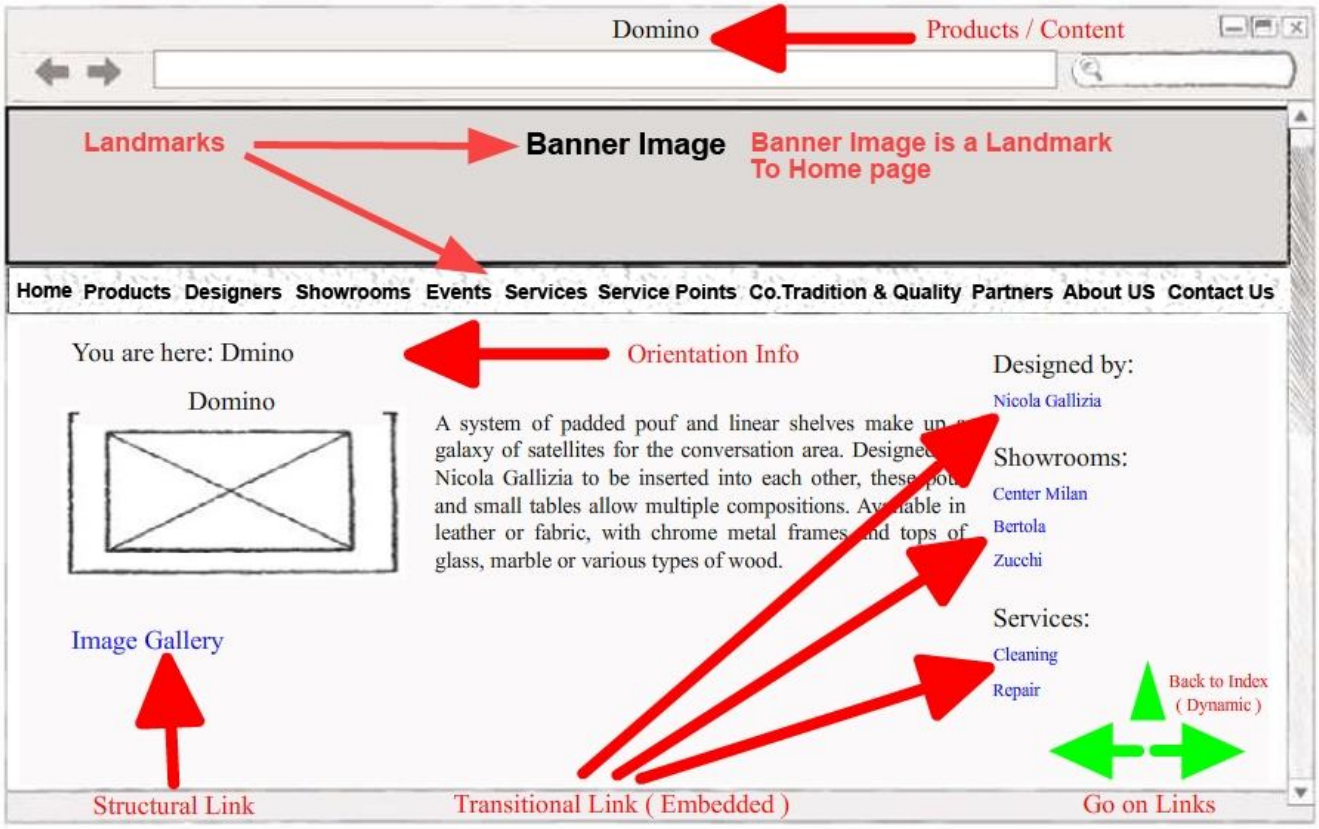
3.1 coherence with mockup

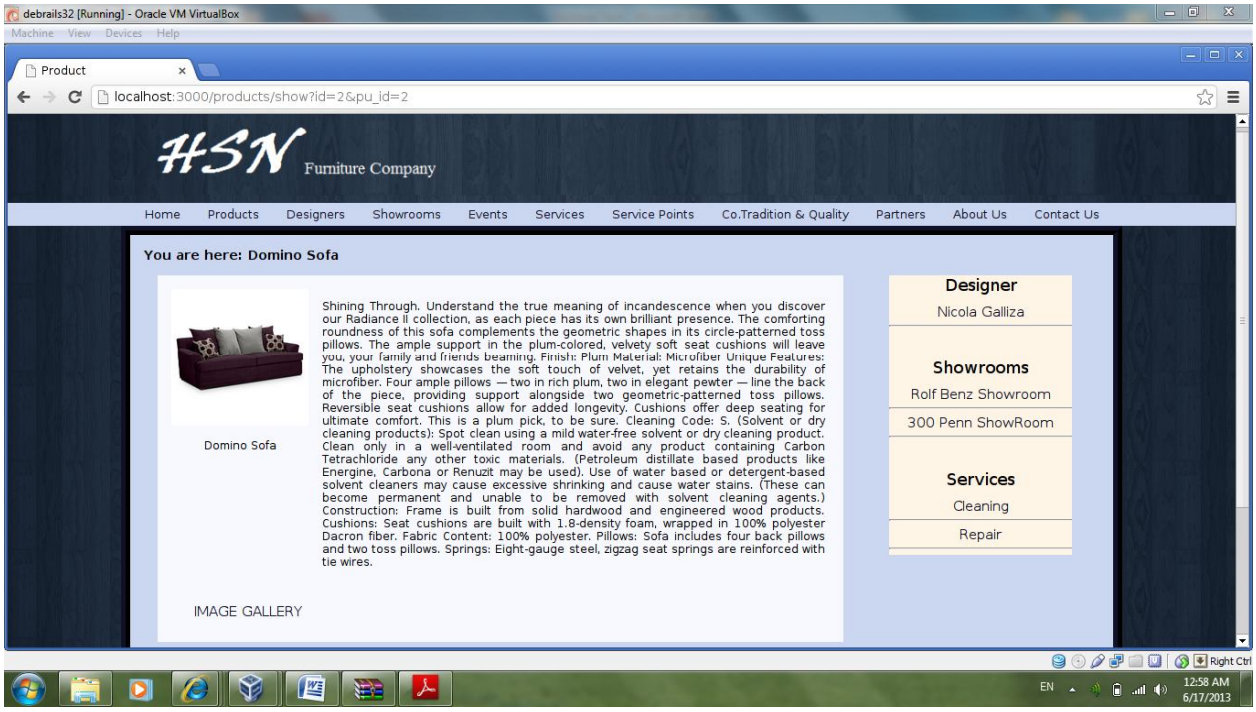
This section is to show the coherence of implementation of pages with the mockup.
The general structure is kept, but in real implementation we did more. To avoid the redundancy of document, we compare just one page here for showing the coherence. But all the pages are coherence according to the mockup in design document.
The general framework is like this:



4 Comparison with mockup:

The page implementation is strictly following mockup structure with optimized decorations.

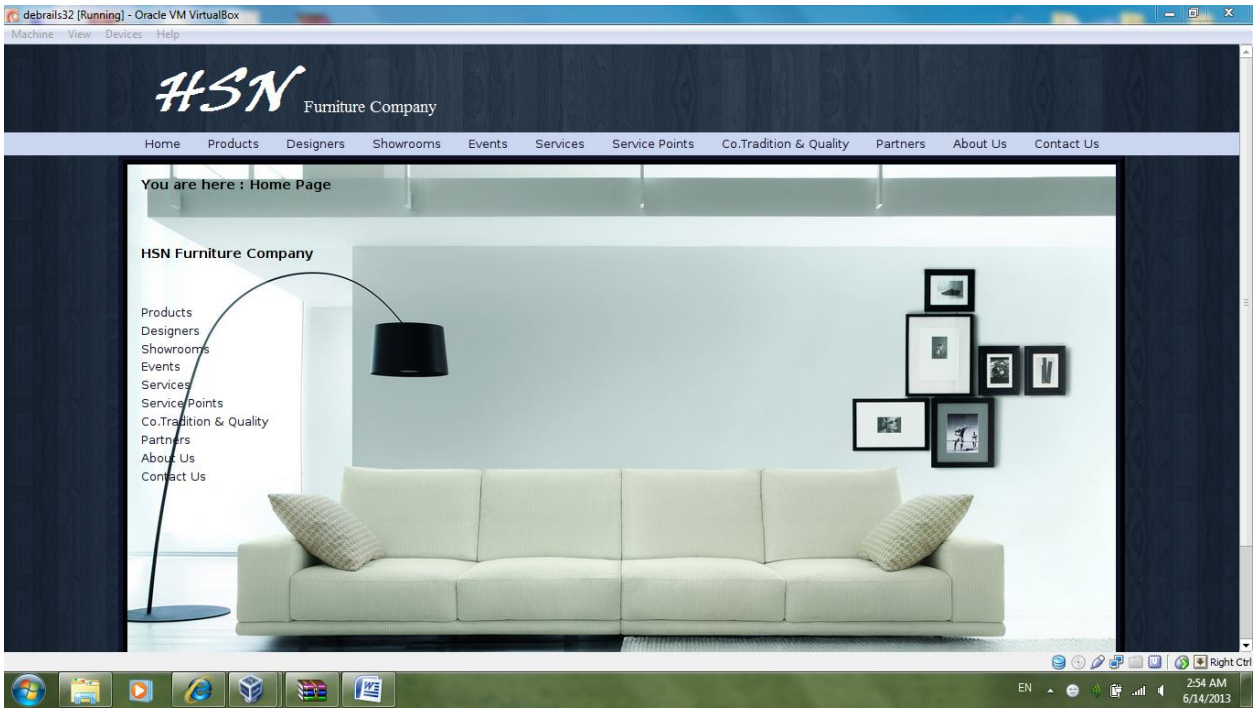




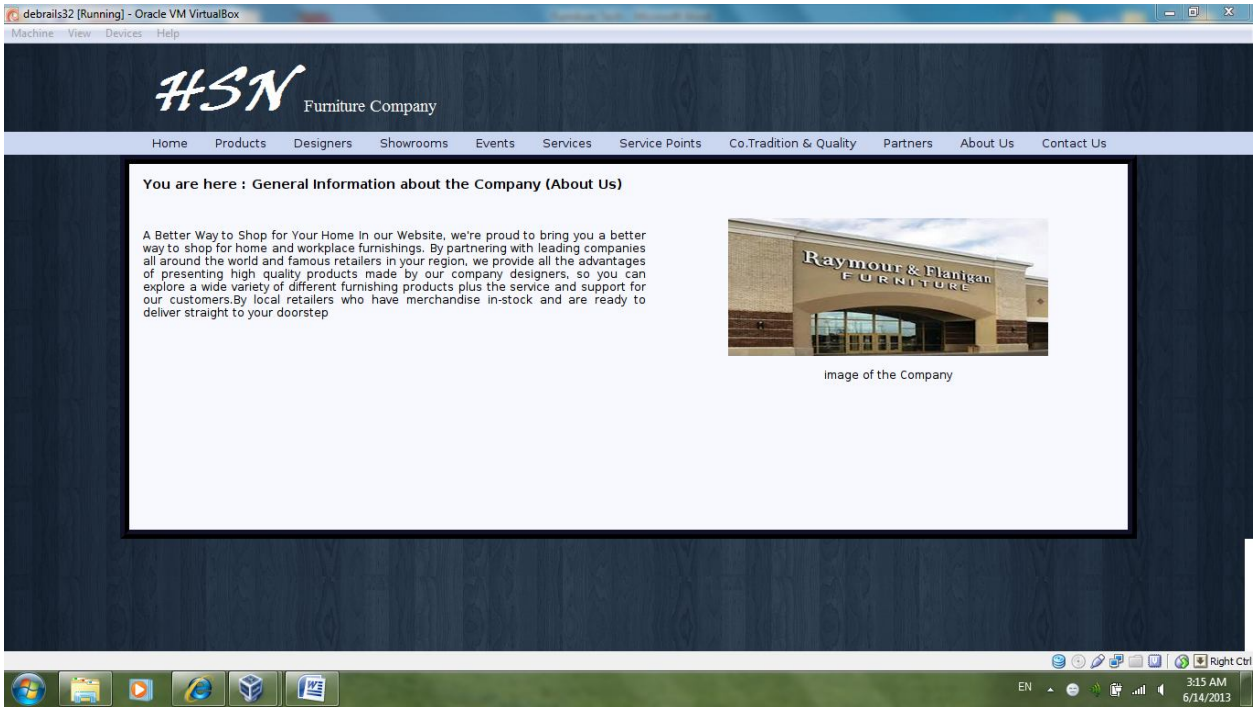
5 Instruction of using

Landmarks:

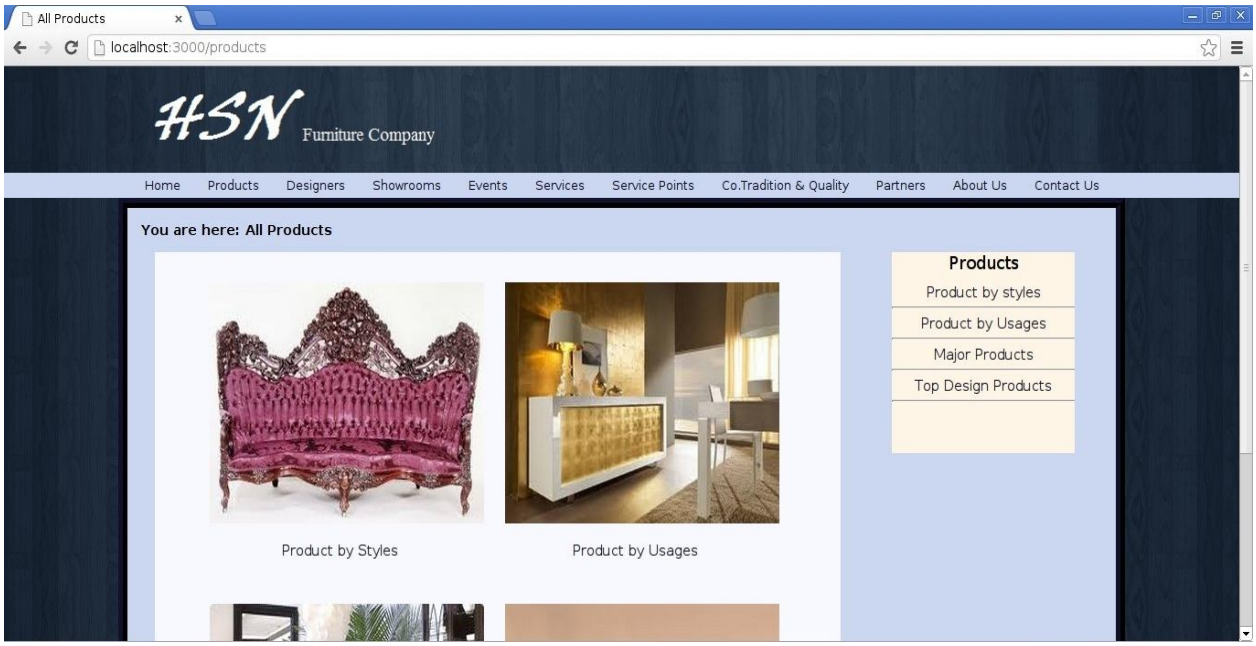
Homepage:



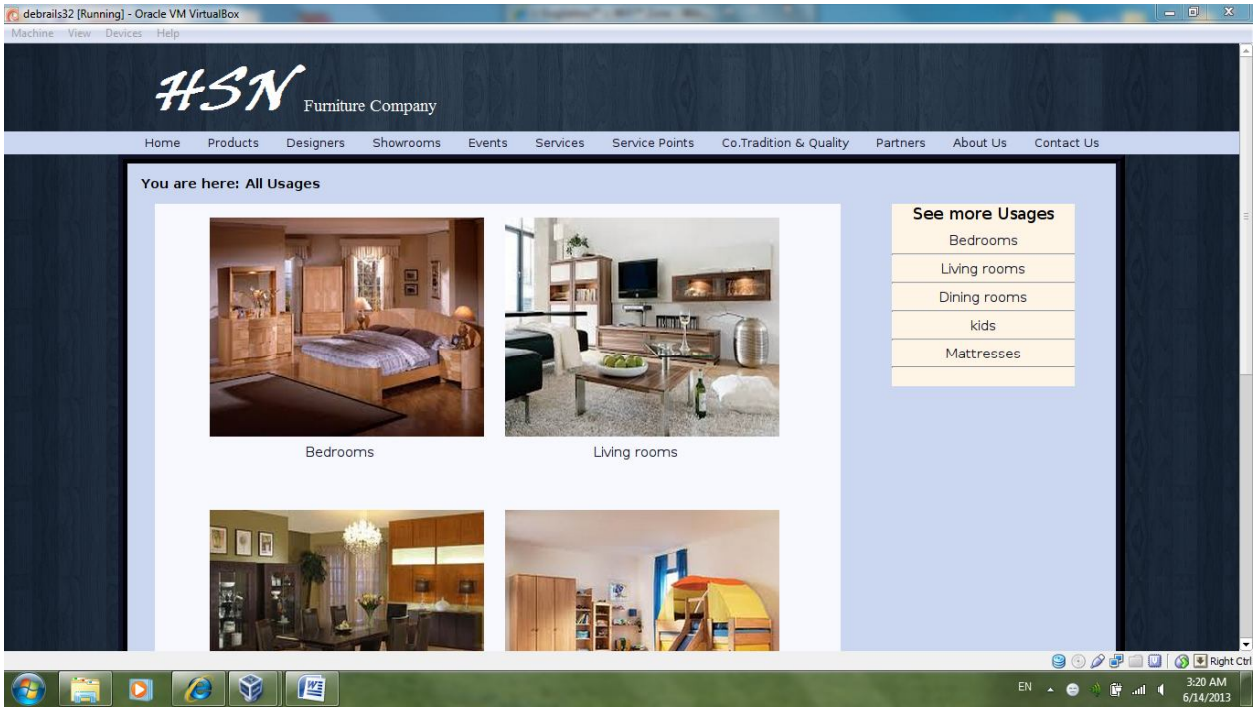
About Us:



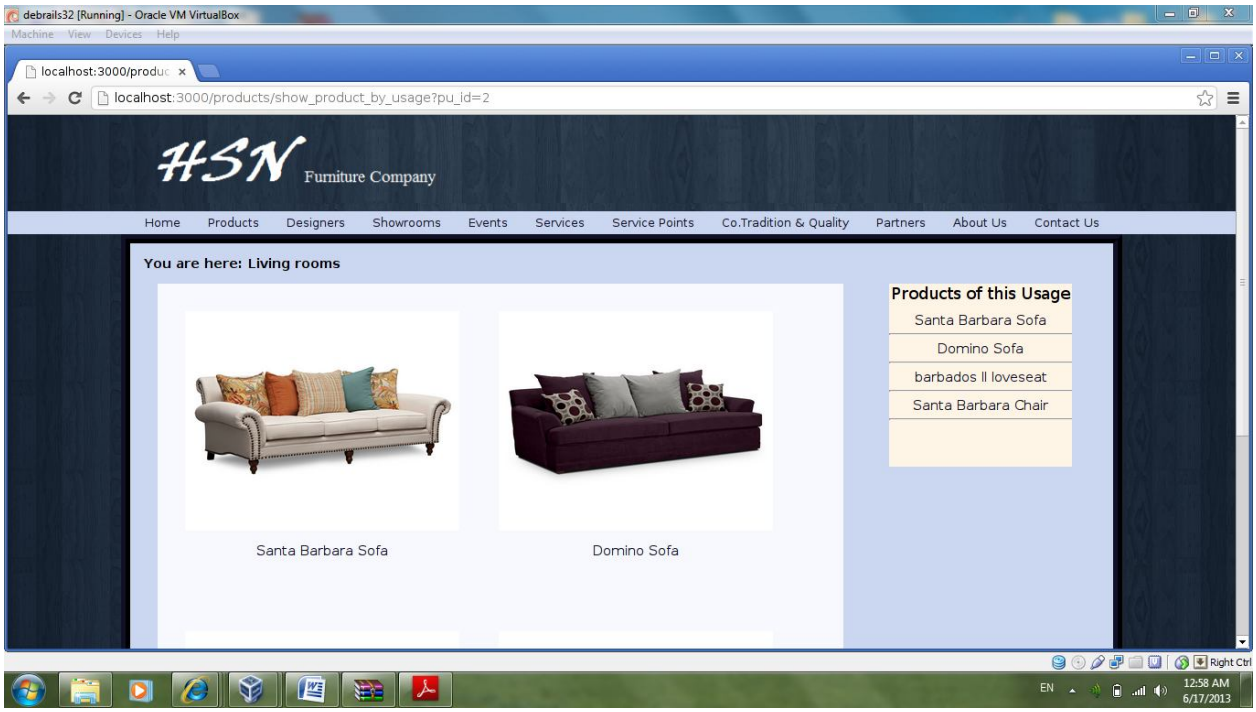
All Products:



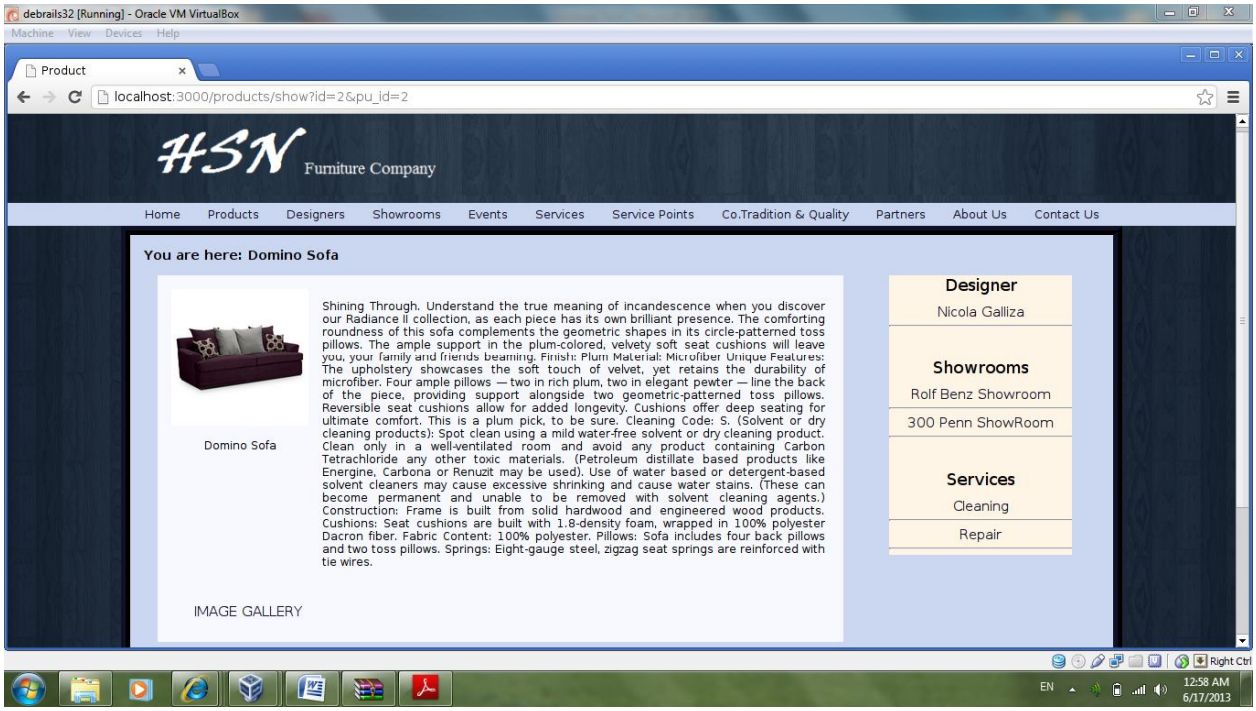
All usages:



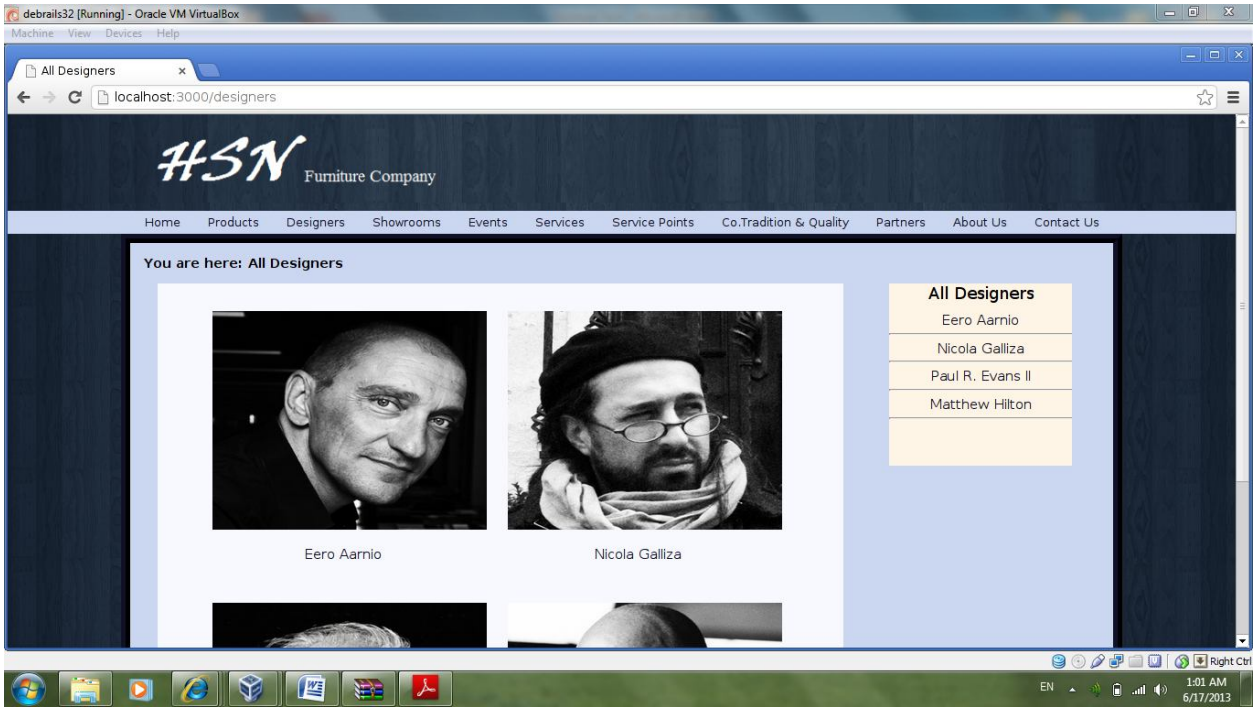
Products by usage:



Product:

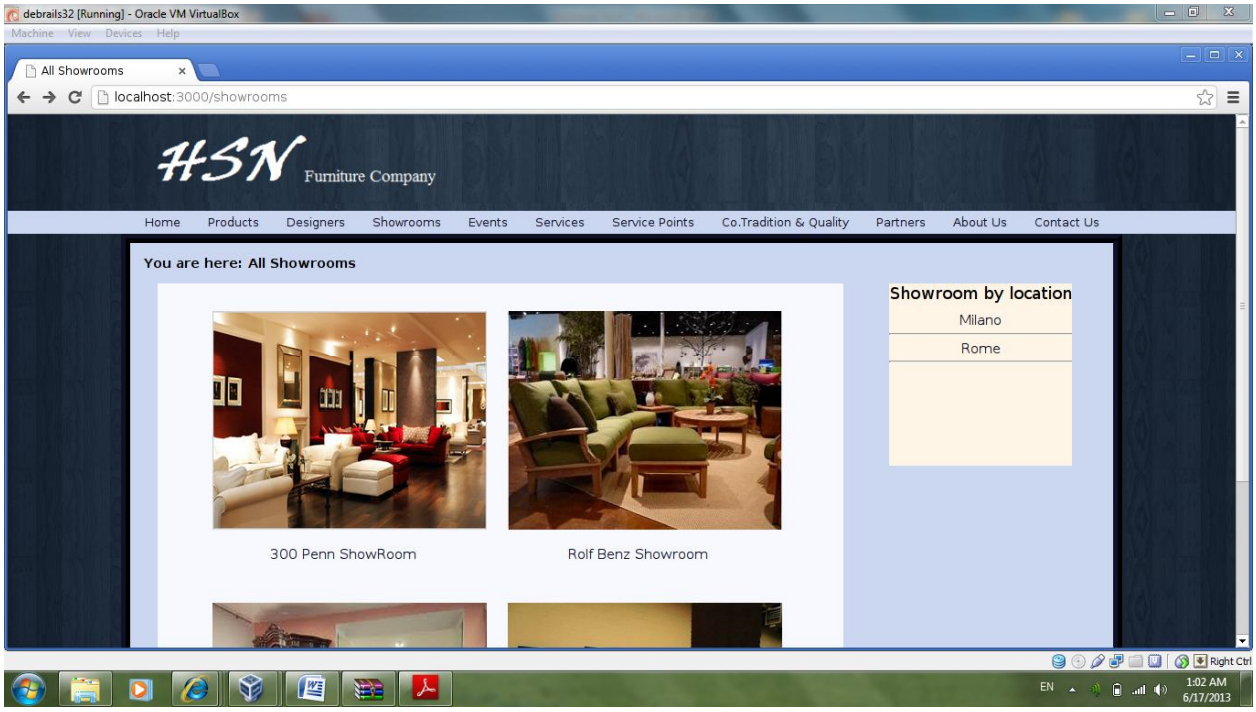


All Designers:

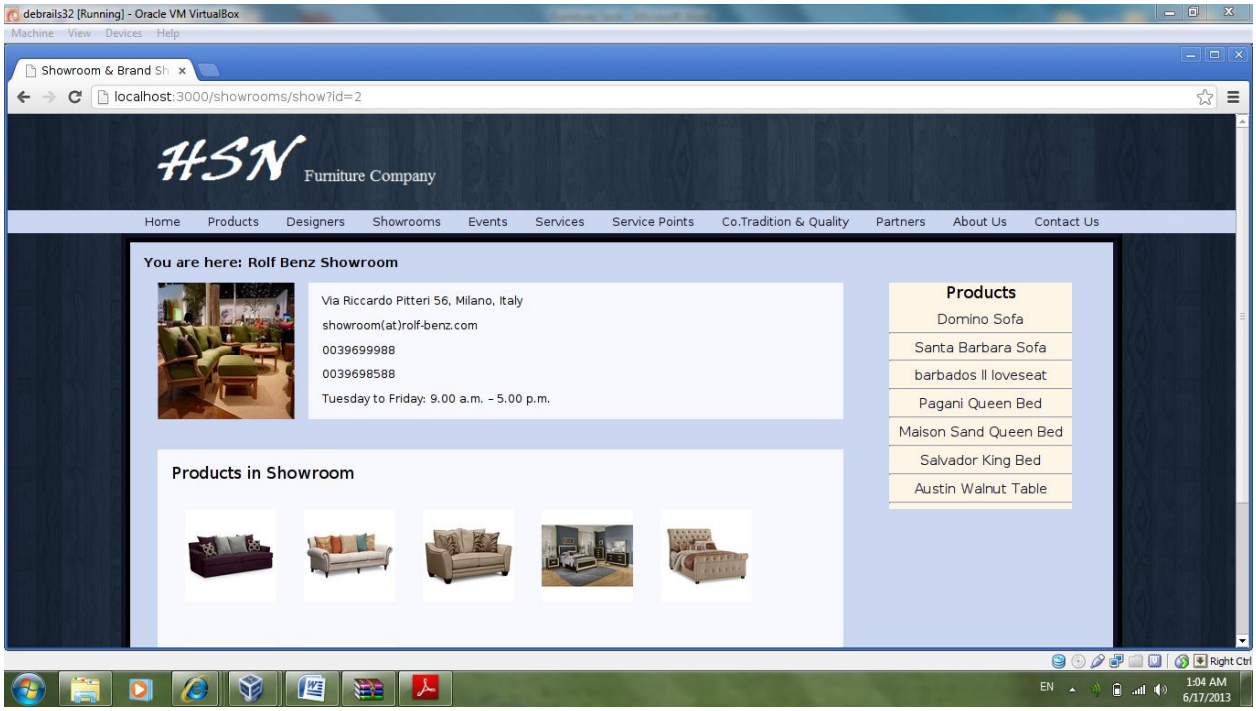


Designer:

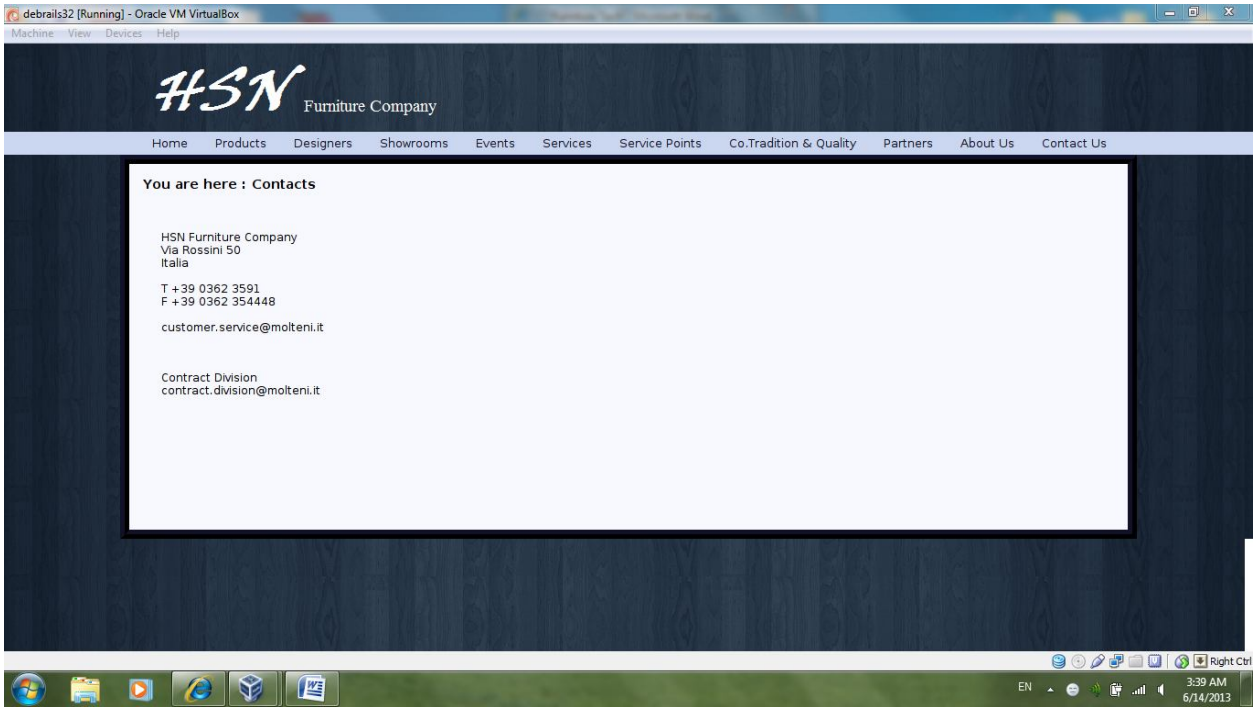
All Showrooms:



Showroom & Brand shop:



Contact us:



6 Technical details

6.1 Data Base explanation.

Here we present the tables of data base.

6.1.1 Designers

ID: The unique ID for a single designer

D_name: designer name

D_desc: description of designer

D_pic: picture of designer

6.1.2 Product_usages

ID: The unique ID for a single product_usage

Pu_title: title of the usage

Pu_pic: picture of the usage

6.1.3 Products

ID: The unique ID for a single product

Pr_name: name of the product

Pr_desc: description of the product

Pr_pic1: first picture of the product

Pr_pic2: second picture of the product

Pr_pic3: third picture of the product

Pr_pic4: forth picture of the product

Pr_pic5: fifth picture of the product

Pr_pic6: sixth picture of the product

Pr_pic7: seventh picture of the product

Pr_pic8: eighth picture of the product

Pr_major: defines if the product is one of major products or not

Pr_top: defines if the product is one of top-design products or not

Designer_id: the ID of designer table as a foreign key

Pr_style: the ID of style table as a foreign key

Pr_usage: the ID of product_usages table as a foreign key

6.1.4 Products_showrooms

This table is used as the interface table for the n to n relation between products and showrooms

Product_id: the id of the product table as a foreign key

Showroom_id: the id of the showrooms table as a foreign key

6.1.5 Showrooms

ID: The unique ID for a single showroom

s_title: title of the showroom

s_city: city where the showroom located in

s_address: the address of the showroom

s_email: email of the showroom

s_tel: tel number of the showroom

s_fax: fax number of the showroom

s_opening: opening time of the showroom

s_pic: map picture of the showroom

6.2 Controllers

In each controller we call the needed service functions. For each controller the Ruby on Rails class automatically is made by the platform.

6.2.1 General controllers

Designers_Controller
Adds and edits the Designer table, also shows the all Designers and manages go-on links for the Designers tour.

Product_usages_Controller
Adds and edits the Product_usages table, also shows the all usages.

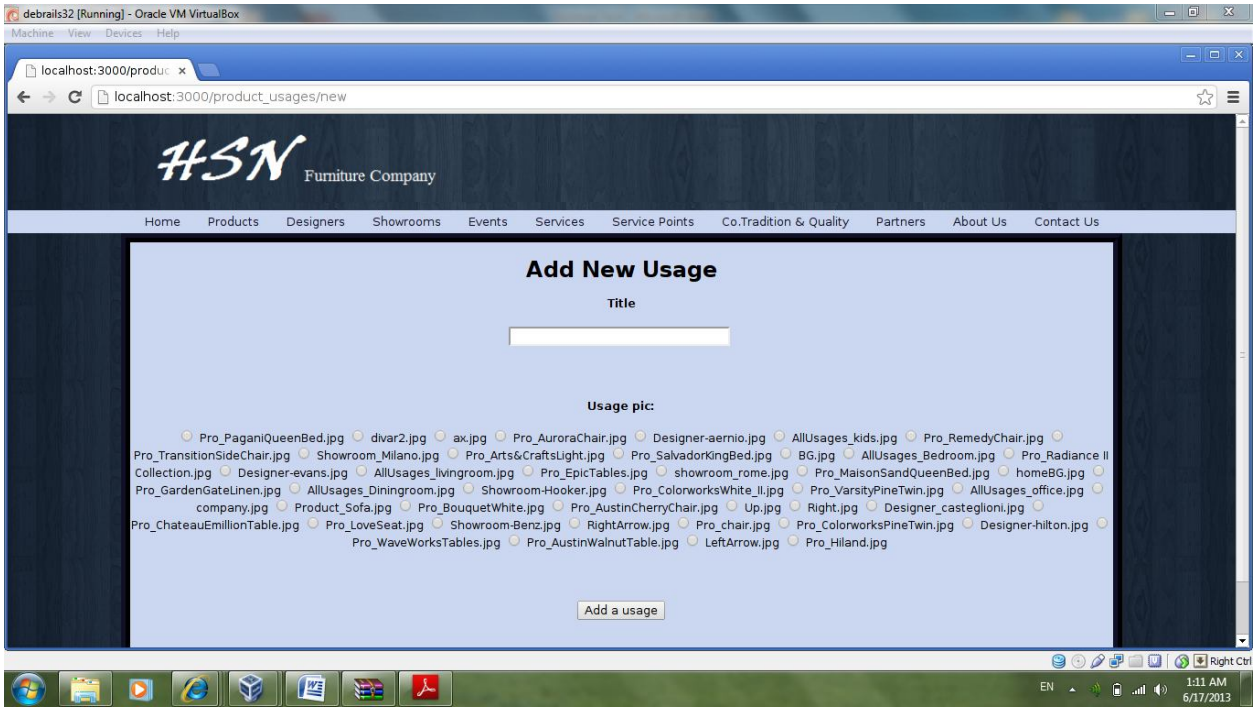
Prooducts_Controller
Adds and edits the Product table, also shows the Product by usage and manages go-on links for the Products tour. Moreover finds the showrooms and designer of the product.

Showrooms_Controller
Adds and edits the Showrooms table, also shows the all Showrooms and manages go-on links for the Showrooms tour.

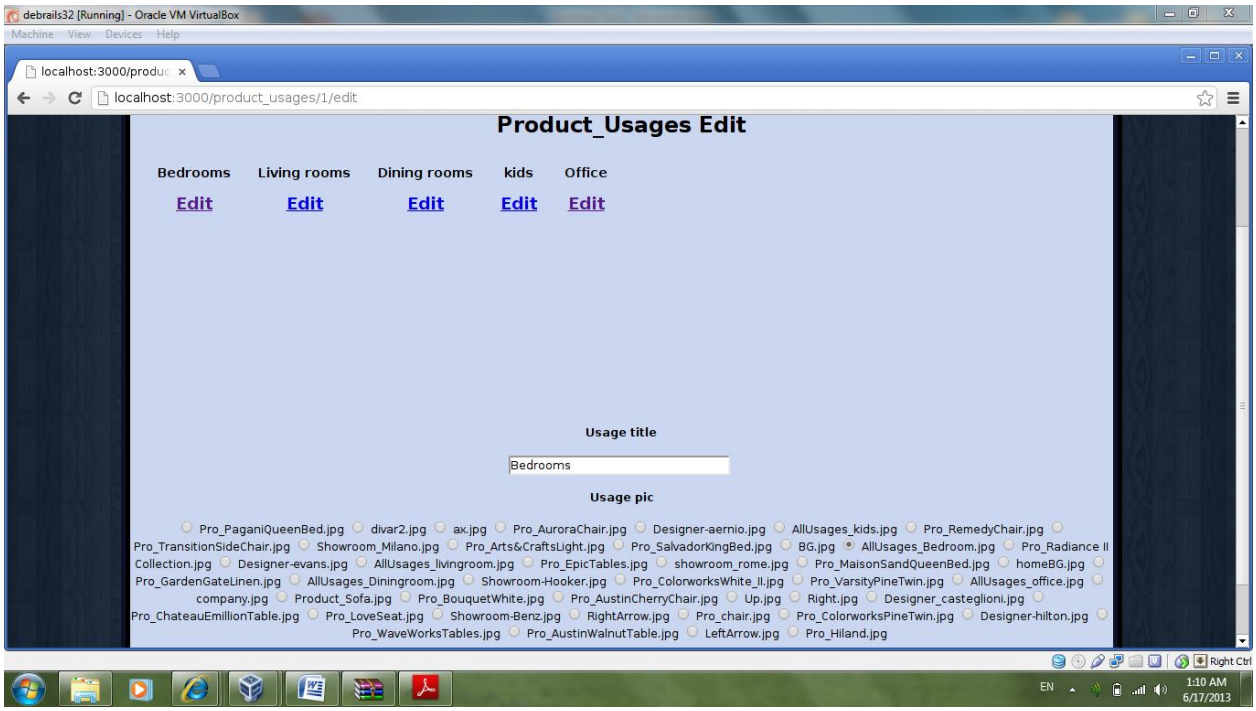
7 Testing admin pages

We implemented admin page for adding new Product, new Showroom, new Designer, new Product_usage and updating existing Products, Showrooms, Designers and Product_usages.
For accessing the admin pages, we implemented a Control Panel page containing all links to admin pages.
Note:
1. When adding a product, the usage and designer have to be in the data base. If we want to add a product, we have to add the designer and usage first.

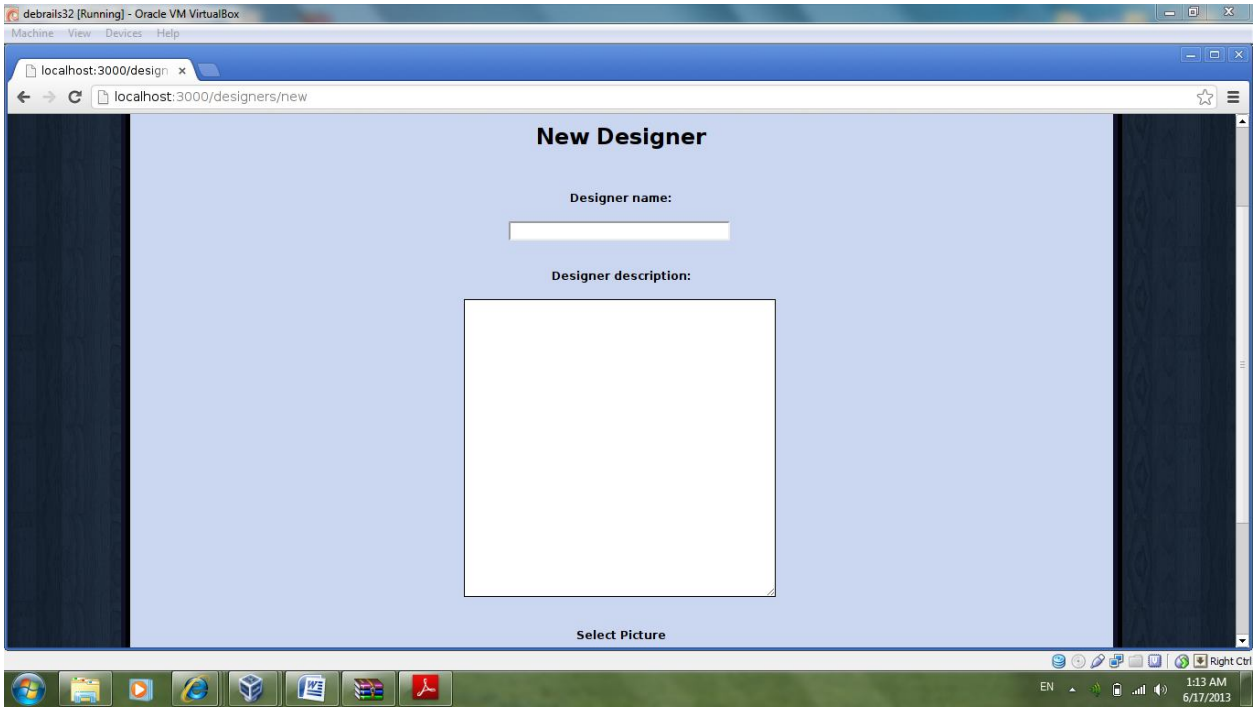
New Usage:



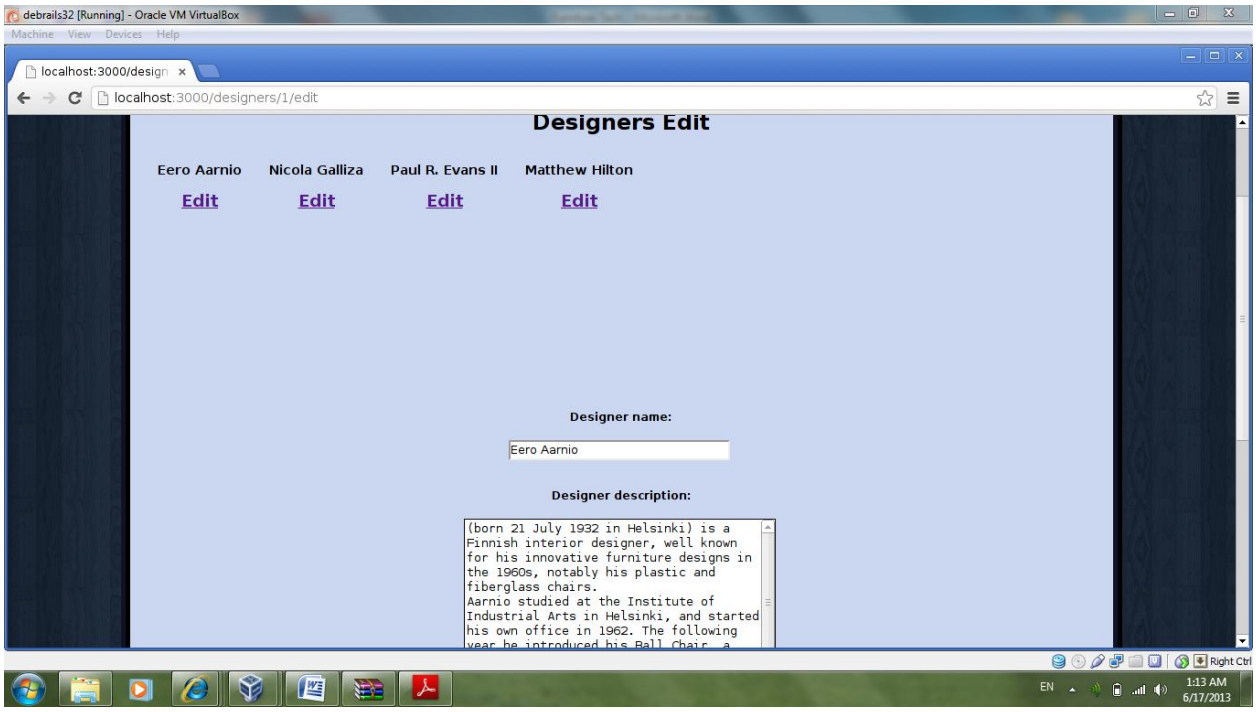
Edit Usage :



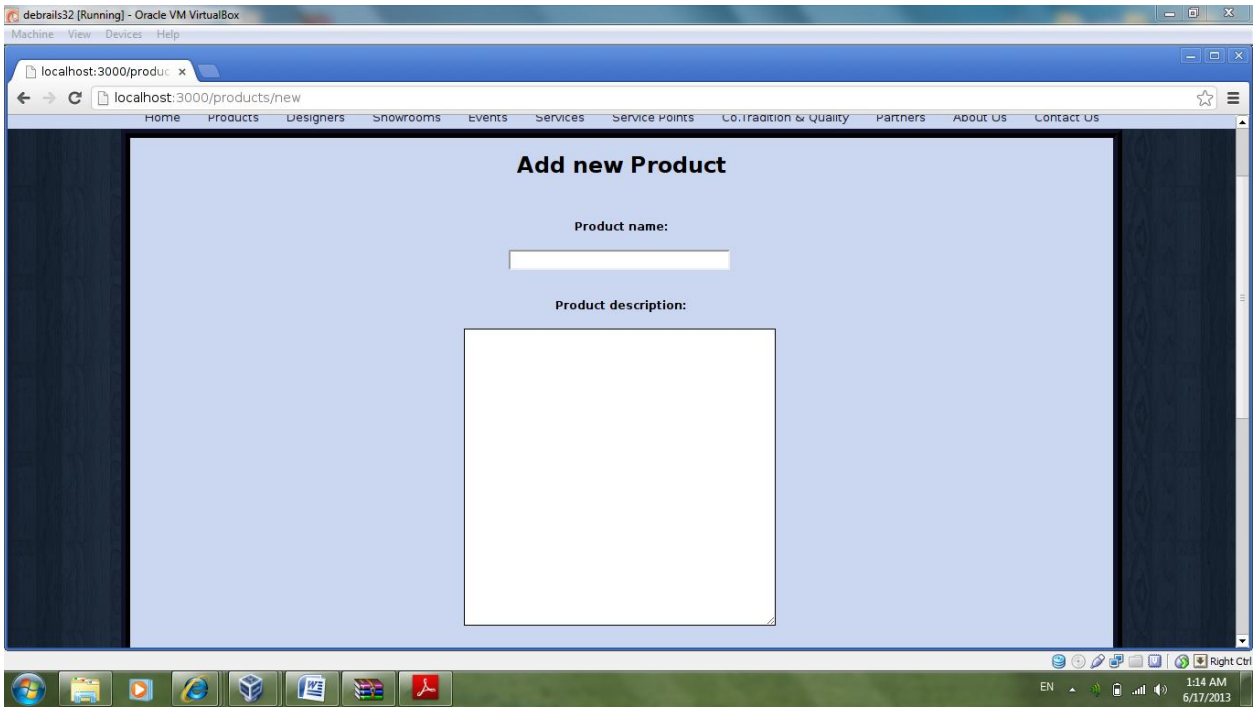
New Designer:



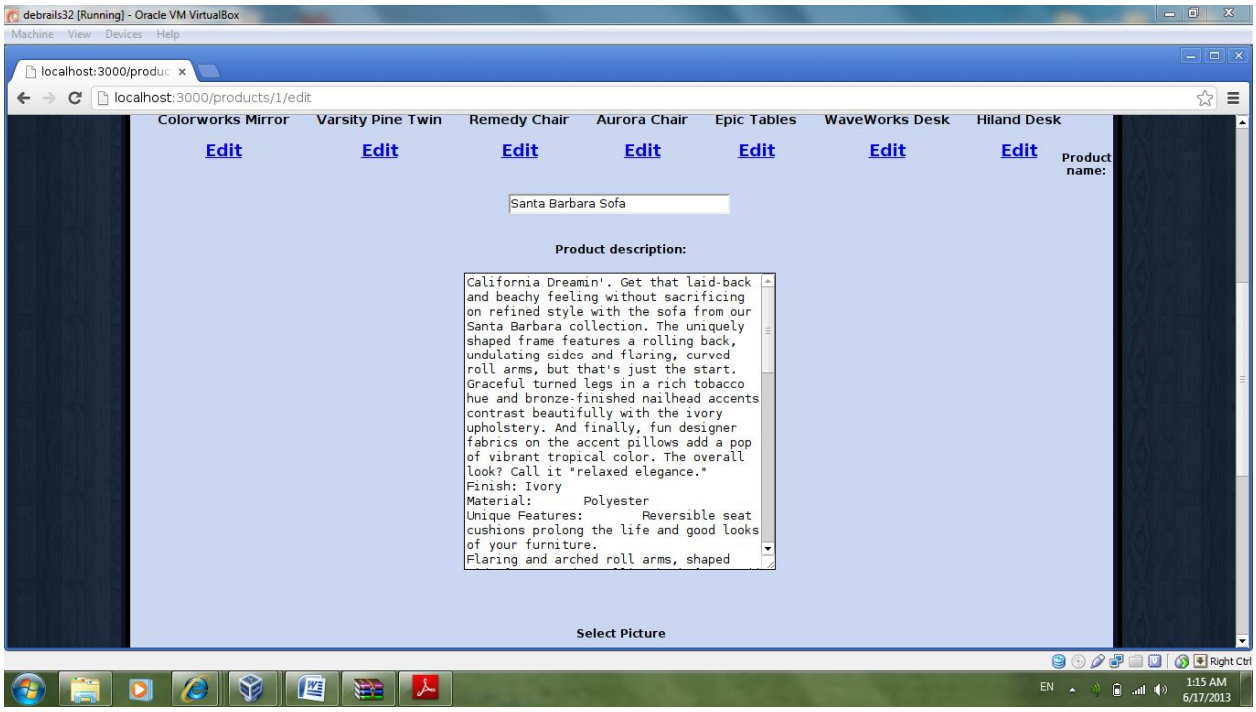
Edit Designer:



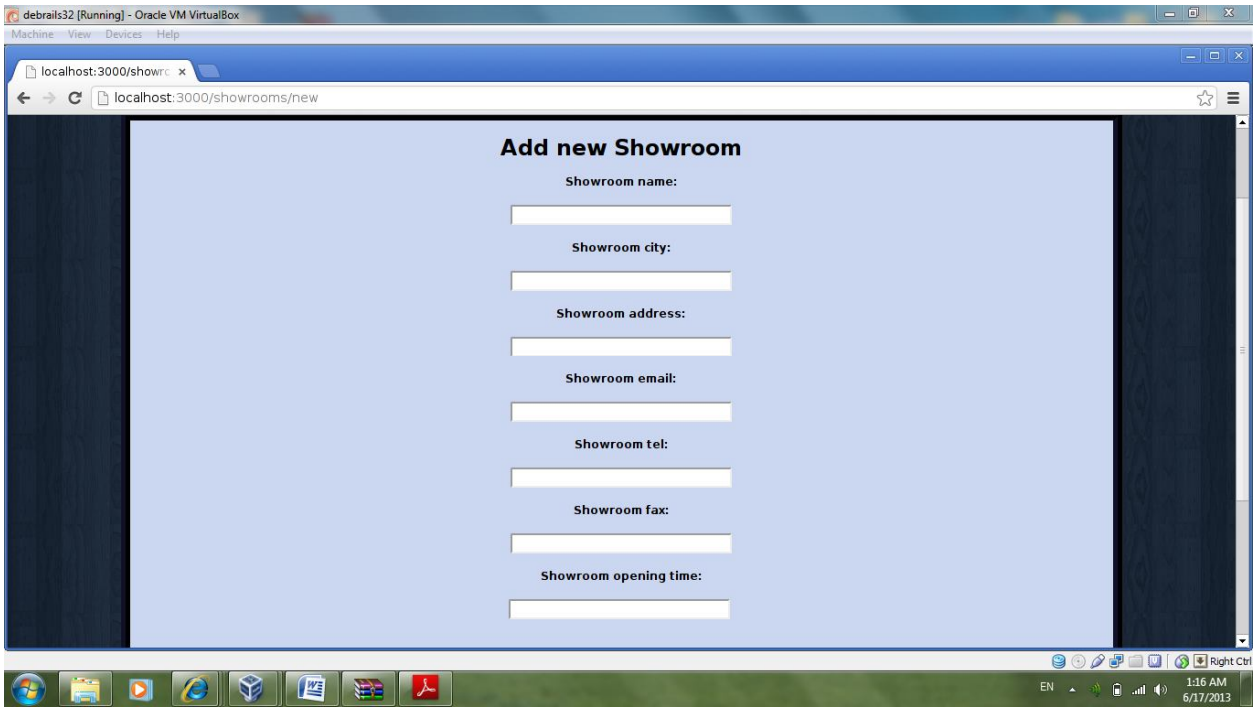
New Product:



Edit Product:



New Showroom:



Edit Showroom: