

Secure, scalable and component-based webshop using struts and hibernate

A Master thesis proposal of GenX HB for Kungliga Tekniska Högskola.

Draft version 0.2: 26th February 2006

Diploma student: Lucas Díez
Examiner: Vladimir Vlassov
Industry supervisor: Viktor Lewin Duran
Timeframe: 27 February, 2006 - 15 August, 2006

1 Requirement description.

The goal of this thesis is to build a scalable, fail tolerance, internationalized, secure and component based web shop (S.F.I.S.C. Web shop).

1.1 Scalability.

Through the architecture the application will reach high scalability, the pattern is called layer, and divides the shop into several layers. In our case will be four: presentation (for the customer and the administrator), the business (which solves the tasks demands from the presentation layer), integration (which takes care of the communication between the database and the business) and the resource (the database itself).

1.2 Fail tolerance.

The program will implement several mechanism such a validation, transactions, rollbacks and more to accomplish the task solicited by the user or administrator.

1.3 Internationalized.

We live in a more global world everyday, and it's important to speak several languages so it's also important that the software can manage several languages for different customers.

1.4 Security.

The application will implement several mechanism for ensure that authentication, authorization and validation works and avoid the users to do mistakes or to malicious users to hack the shop.

1.5 Components.

The shop will implement interfaces in many part of the program with the intention of make the different components replaceable. The aim of this is to make easier for the developers to upgrade the shop without changing the design or the architecture. The components that become old or instable can be easily replace by new ones.

2 Details of the application.

The application will use a framework in the presentation. The framework will solve many of the commons problems linked to the presentation such as internationalization, validation, navigation, etc..

The files in the presentation such a jsp or html won't have any java code to make possible that design won't be affect by the java code. The file will have tags instead that invoke java code fom the tags file. This gives freehand to the designer of the pages to do anything. This is basically the MVC ¹ approach for the web pages.

After looking several books and comparisons between different frameworks we choose struts because it's more widely use, it has more documentation than other and it's well tested.

In the business layer we are going to use java classes with hibernate capabilities to talk to the database. The decision was made after comparing the java enterprise beans and the java classes with hibernate capabilities. The enterprise are harder to learn, implied much more code to write and it doesn't provided full object mapping to the database. Hibernate provides full object oriented communication between the application and the database in an easy way at the same time.

And most important of all, Hibernate provides caching, that minimize the traffic between the business and the resource layers.

2.1 Resource.

It will be MySQL 5.0 database. It has support for transactions, procedures and more features that are necessary for the application.

¹Model-View-Controller design pattern

2.2 Application server.

The application will be deployed into Jboss or Apache Tomcat, the final decision will be taken during the inception phase of the project.